

İnce bir L^AT_EX 2_ε Elkitabı

veya, 116 dakikada L^AT_EX 2_ε

Yazarlar: Tobias Oetiker

Hubert Partl, Irene Hyna and Elisabeth Schlegl

İngilizceden deęişikliklerle Trkeleřtiren: **Bekir Karaoęlu**

Srm 4.20, 31 Mayıs 2006

Copyright ©1995-2005 Tobias Oetiker ve arkadaşları. Tüm hakları saklıdır.

Bu dokümanın telif hakları ücretsizdir, onu çoğaltıp dağıtabilir ve/veya Özgür Yazılım Vakfı'nın GNU Kamusal Lisansı şartlarına uygun olarak değiştirebilirsiniz.

Bu doküman yararlı olacağı düşüncesiyle, fakat kullanıcıya HİÇBİR GARANTİ vermeksizin dağıtılmaktadır. Daha fazla ayrıntı için GNU Kamusal Lisans şartnamesine bakınız.

Bu belgeyle birlikte GNU Kamusal Lisans şartnamesinin bir kopyasını da almış olmalısınız. Eğer almadıysanız (Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA) adresine yazarak temin edebilirsiniz.

Bu dokümanın Türkçe telif hakları Bekir Karaoğlu tarafından kamuya armağan edilmiştir, serbestçe çoğaltıp dağıtabilirsiniz.

Copyright ©1995-2005 Tobias Oetiker and Contributors. All rights reserved.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Teşekkürler!

Bu kitaptaki bilgilerin büyük çoğunluğu Almanca yazılmış L^AT_EX 2.09'a Giriş kitabından kaynaklanmaktadır. En başta, o kitabı yazan Avusturyalı meslekdaşlarımıza teşekkür ediyoruz:

Hubert Partl <partl@mail.boku.ac.at>

Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna <Irene.Hyna@bmwf.ac.at>

Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <noemail>

in Graz

Almanca kitapla ilgilenenler, Jörg Knappen'in L^AT_EX için yazdığı yeni sürümünü

CTAN:/tex-archive/info/lshort/german

adresinden temin edebilirler.

Aşağıda isimleri bulunun insanlar bu elkitabının daha iyi olabilmesi için düzeltmeler, öneriler ve yeni bilgilerle yardımcı oldular. Bu kitabın elinizdeki haline gelebilmesinde büyük katkıları oldu. Hepsine en içten teşekkürlerimi sunmak isterim. Kitapta bulacağınız diğer tüm hatalar benimdir.

Rosemary Bailey, Marc Bevand, Friedemann Brauer, Jan Busa,
Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos,
Neil Carter, Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke,
Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes,
Matthias Dreier, David Dureisseix, Elliot, Hans Ehrbar, Daniel Flipo, David Frey,
Hans Fugal, Robin Fairbairns, Jörg Fischer, Erik Frisk, Mic Milic Frederickx,
Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Andy Goth,
Cyril Goutte, Greg Gamble, Frank Fischli, Morten Høgholm, Neil Hammond,
Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen,
Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones,
Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec,
Sander de Kievit, Alain Kessi, Christian Kern, Tobias Klauser, Jörg Knappen,
Kjetil Kjernsmo, Maik Lehardt, Rémi Letot, Flori Lambrechts, Axel Liljencrantz,
Johan Lundberg, Alexander Mai, Hendrik Maryns, Martin Maechler,
Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren,
Richard Nagy, Philipp Nagele, Lenimar Nunes de Andrade, Manuel Oetiker,
Urs Oswald, Martin Pfister, Demerson Andre Polli, Nikos Pothitos,
Maksym Polyakov Hubert Partl, John Reffing, Mike Ressler, Brian Ripley,
Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Risto Saarelma,
Hanspeter Schmid, Craig Schlenker, Gilles Schintgen, Baron Schwartz,
Christopher Sawtell, Miles Spielberg, Geoffrey Swindale, Laszlo Szathmary,
Boris Tobotras, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli,
Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone,
and Mikhail Zotov.

Önsöz

\LaTeX [1] yüksek baskı kalitesinde bilimsel ve matematiksel dokümanlar hazırlamaya çok uygun bir dizgi programıdır. Ayrıca, basit bir mektuptan komple bir kitaba kadar, diğer her türlü belge yazımında da kolaylık sağlar. \LaTeX dokümana biçim verirken daha kapsamlı olan \TeX programını kullanır.

Bu kısa elkitabı \LaTeX programına bir giriş olarak pekçok \LaTeX uygulamaları için yeterlidir. \LaTeX sistemini daha kapsamlı olarak öğrenmek istiyorsanız, [1, 3] gibi kaynaklara başvurabilirsiniz.

Bu elkitabı 6 bölümden oluşur:

Bölüm 1 \LaTeX ile hazırlanan bir dokümanın temel yapısını anlatır. Biraz da \LaTeX 'in tarihçesinden söz eder. Bu bölümü okuduktan sonra, \LaTeX 'in nasıl çalıştığı hakkında kabaca bir fikir sahibi olursunuz.

Bölüm 2 doküman dizmenin ayrıntılarına girer. Temel \LaTeX komutlarını ve ortamlarını açıklar. Bu bölümü okuduktan sonra ilk belgenizi yazabilecek konumda olursunuz.

Bölüm 3 \LaTeX 'te formüllerin nasıl girildiğini anlatır. \LaTeX 'in en güçlü olduğu bu konuyu bol örneklerle gösterir. Bölüm sonundaki tablolarda \LaTeX 'te kullanabileceğiniz tüm matematik sembolleri bulunmaktadır.

Bölüm 4 belgeniz için nasıl izin ve kaynakça hazırlayacağınızı, EPS formatında nasıl grafik ekleyebileceğinizi anlatır. $\text{pdf}\LaTeX$ programı ile PDF formatında doküman çıkarmayı gösterir ve çok kullanışlı bazı program paketlerini tanıtır.

Bölüm 5 Grafik çiziminde \LaTeX 'in nasıl kullanıldığını anlatır. Başka grafik programlarında çizilmiş ve dosya olarak saklanmış grafikleri \LaTeX 'e eklemek yerine, grafiği tanımlayıp \LaTeX 'e çizdirmenin yolunu gösterir.

Bölüm 6 \LaTeX 'in kullandığı standard doküman şablonlarını değiştirebileceğiniz bazı tehlikeli bilgileri içerir. \LaTeX 'in ürettiği güzelim çıktıları, yeteneğinize bağlı olarak, nasıl daha çirkin veya daha göz kamaştırıcı yapabileceğinizi anlatır.

Bu bölümleri sırayı bozmadan öğrenmelisiniz— zaten, elkitabı fazla uzun değil. Örnekleri dikkatli inceleyin, zira kitaptaki pekçok önemli bilgi örnekler içinde yer almaktadır.

L^AT_EX programı küçük PC veya Mackintosh'tan, büyük UNIX veya VMS işletim sistemlerine kadar, pekçok bilgisayar türü için mevcuttur. Bazı üniversite ağlarında herkesin kullanımına hazır olarak bulunabilmektedir. Kurumunuzdaki yerel L^AT_EX programına nasıl girip kullanabileceğiniz *Local Guide* [5] (Yerel Rehber) dosyasında bulunmalıdır. Programı başlatmakta zorluk çekerseniz size bu elkitabını veren kişiye danışın. Bu elkitabının amacı L^AT_EX programını kurup çalıştırmayı öğretmek *değildir*, L^AT_EX ile işlenebilecek dokümanları nasıl yazabileceğinizi öğretir.

L^AT_EX konusunda herhangi bir belge veya programa ihtiyacınız olursa, Comprehensive T_EX Archive Network (CTAN) sitelerine bir bakın. Ana site <http://www.ctan.org> adresinde bulunmaktadır. Tüm program paketleri <ftp://www.ctan.org> ftp arşivinden ve diğer tüm ülkelerdeki ayna sitelerden indirilebilir.

Kitapta değişik yerlerde CTAN arşivinden indirebileceğiniz yazılım ve belgelere atıfta bulunuyorum. Bunların tam URL adresini yazmak yerine, sadece CTAN: yazıp alt klasörlerin yerini ekliyorum.

Siz de kendi bilgisayarınızda L^AT_EX ile çalışmak istiyorsanız, ne tür seçenekler olduğunu <CTAN:/tex-archive/systems> klasörüne bakarak görebilirsiniz.

Bu elkitabına eklenebilecek, çıkarılabilecek veya değiştirilebilecek bir öneriniz varsa, bana bildirin. Özellikle acemi L^AT_EX kullanıcılarının hangi konuları anlamakta zorluk çektiklerini, hangi konularda daha fazla bilgi istediklerini duymak isterim.

Tobias Oetiker <oetiker@ee.ethz.ch>

Department of Information Technology and
Electrical Engineering,
Swiss Federal Institute of Technology

Bu dokümanın en güncel sürümünü
<CTAN:/tex-archive/info/lshort> adresinde bulabilirsiniz.

İÇİNDEKİLER

Teşekkürler!	iii
Önsöz	v
1 Bilmeniz Gerekenler	1
1.1 Meselenin Özü	1
1.1.1 T _E X	1
1.1.2 L ^A T _E X	2
1.2 Temel Kavramlar	2
1.2.1 Yazar, Tasarımcı ve Dizgici	2
1.2.2 Şablon Tasarımı	2
1.2.3 Ne Getirir, Ne Götürür?	3
1.3 L ^A T _E X Girdi Dosyaları	4
1.3.1 Boşluklar	4
1.3.2 Özel Karakterler	5
1.3.3 L ^A T _E X Komutları	5
1.3.4 Yorum Satırları	6
1.4 Girdi Dosya Yapısı	6
1.5 Tipik Bir Doküman Yazımı	7
1.6 Bir Dokümanın Yerleşim Planı	9
1.6.1 Doküman Sınıfları	9
1.6.2 Paketler	9
1.6.3 Sayfa Biçimleri	10
1.7 Karşılaşabileceğiniz Dosya Türleri	13
1.8 Büyük Projeler	14
2 Dizgi	17
2.1 Metin ve Dil Yapısı	17
2.2 Satır ve Sayfa Kesme	19
2.2.1 Çift Yaslanmış Paragraflar	19
2.2.2 Heceleme	20
2.3 Hazır Özel Kelimeler	21
2.4 Özel Karakterler ve Semboller	21

2.4.1	Tırnaklar	21
2.4.2	Çizgiler ve Tireler	22
2.4.3	Tilda (\sim)	22
2.4.4	Derece İşareti (\circ)	22
2.4.5	Avro Para Sembolü (€)	22
2.4.6	Üç Nokta (\dots)	23
2.4.7	Kalıplama	24
2.4.8	Aksanlar ve Özel İşaretler	24
2.5	Uluslararası Dil Desteği	25
2.5.1	Türkçe Desteği	27
2.5.2	Almanca Desteği	27
2.6	Kelime Arası Boşluklar	28
2.7	Başlıklar, Bölümler, ve Kısımlar	29
2.8	İç Atıflar	31
2.9	Dipnotlar	31
2.10	Vurgulanmış Kelimeler	32
2.11	Ortamlar	32
2.11.1	Sıralandırma, Numaralandırma, ve Maddeleme	33
2.11.2	Sola Yaslama, Sağa Yaslama, ve Ortalama	33
2.11.3	Alıntı Yapmak ve Şiir Dizmek	34
2.11.4	Özet (Abstrak)	35
2.11.5	Yazıldığı Gibi (Verbatim) Basmak	35
2.11.6	Tablolar	36
2.12	Yüzer-Gezer Nesnelere	38
2.13	Kırılgan Komutları Korumak	40
3	Matematik Formülleri	41
3.1	Genel	41
3.2	Matematik Kipinde Gruplandırma	43
3.3	Bir Matematik Formülünün Yapıtaşları	43
3.4	Matematikte Boşluklar	47
3.5	Düşey Hizalanmış İfadeler	48
3.6	Hayaletler	50
3.7	Matematik Yazıtipi Puntosu	50
3.8	Teoremler, Yasalar, ...	51
3.9	Kalın Semboller	53
3.10	Matematik Semboller Listesi	54
4	Özel Konular	63
4.1	POSTSCRIPT Formatında Grafik Ekleme	63
4.2	Kaynakça	64
4.3	Dizin	66
4.4	Tepelik ve Diplikler	67
4.5	Verbatim Paketi	69

4.6	Diğer Paketleri Kurmak	69
4.7	PDF Formatında Doküman Çıkarmak	70
5	Grafik Çizimleri	73
5.1	Genel Bakış	73
5.2	picture (resim) Ortamı	74
5.2.1	Temel Komutlar	74
5.2.2	Doğru Parçaları	75
5.2.3	Oklar	76
5.2.4	Çemberler	77
5.2.5	Metin ve Formüller	78
5.2.6	\multiput ve \linethickness Komutları	78
5.2.7	Ovaller	79
5.2.8	Önceden Tanımlanmış Resim Kutuları Kullanma	80
5.2.9	Kuadratik Bézier Eğrileri	81
5.2.10	Zincir Eğrisi	82
5.2.11	Özel Görelilik Teorisinde Hızlılık	83
5.3	Xy-pic Paketi	83
6	L^AT_EX'i Özelleştirmek	87
6.1	Yeni Komutlar, Ortamlar ve Paketler	87
6.1.1	Yeni Bir Komut Tanımlamak	88
6.1.2	Yeni Ortam Oluşturmak	89
6.1.3	Fazladan Boşluklar	89
6.1.4	Kendi Paketinizi Oluşturmak	90
6.2	Yazıtipleri ve Puntolar	91
6.2.1	Yazıtipi Değiştiren Komutlar	91
6.2.2	Dikkat, Başımıza Taş Düşebilir	93
6.2.3	Tavsiye	93
6.3	Aralıklar	94
6.3.1	Satır Aralığı	94
6.3.2	Paragraf Formatlama	94
6.3.3	Yatay Aralıklar	95
6.3.4	Düsey Aralıklar	96
6.4	Sayfa Düzeni	97
6.5	Aralıklarla Diğer İşlemler	99
6.6	Kutular	100
6.7	Çizgiler ve Payandalar	102
	Kaynakça	103
	Index	105

Şekil Listesi

1.1	Minimum bir L ^A T _E X dosyası.	7
1.2	Gerçekçi bir dergi makalesi örneği.	8
4.1	fancyhdr Paketiyle Tepelik Örneği.	68
6.1	Bir Paket Örneği.	90
6.2	Sayfa Düzeni Parametreleri.	98

Tablo Listesi

1.1	Doküman Sınıfları.	10
1.2	Document Sınıfı Opsiyonları.	11
1.3	L ^A T _E X ile Gelen Bazı Paketler.	12
1.4	Önceden Tanımlanmış L ^A T _E X Sayfa Biçimleri.	13
2.1	Bir dolu Avro sembolü	23
2.2	Aksanlar ve Özel İşaretler.	24
2.3	Türkçe Doküman Yazmak için Gerekli Ön Komutlar.	28
2.4	Özel Almanca Harfler.	28
2.5	Yüzer-gezer Yerleştirme Seçenekleri.	38
3.1	Matematik Kipi Aksanları.	54
3.2	Yunan Harfleri.	54
3.3	İkili Bağıntılar.	55
3.4	İkili İşlemciler.	55
3.5	BÜYÜK İşlemciler.	56
3.6	Oklar.	56
3.7	Gruplandırıcılar.	56
3.8	Büyük Gruplandırıcılar.	57
3.9	Değişik Semboller.	57
3.10	Matematiksels Olmayan Semboller.	57
3.11	AMS Gruplandırıcıları.	57
3.12	AMS Yunanca ve İbranice.	57
3.13	AMS İkili Bağıntılar.	58
3.14	AMS Oklar.	59
3.15	AMS Olumsuz İkili Bağıntılar ve Oklar.	60
3.16	AMS İkili İşlemciler.	60
3.17	AMS Değişik Semboller.	61
3.18	Matematik Yazıtipleri.	61
4.1	graphics Paketinin Önemli Komutları.	65
4.2	Dizin Maddesi Girme Örnekleri.	67
6.1	Yazıtipleri.	91

6.2	Yazıtipi Puntoları.	92
6.3	Standart Doküman Sınıflarının Mutlak Puntoları.	92
6.4	Matematik Yazıtipleri.	92
6.5	T _E X Uzunluk Birimleri.	96

Bölüm 1

Bilmeniz Gerekenler

Bu bölümün ilk kısmında $\LaTeX 2_{\epsilon}$ 'nin felsefesi ve tarihçesinin kısa bir özeti yer alır. İkinci kısım bir \LaTeX dokümanının temel yapısını anlatır. Bu bölümü okuduktan sonra, \LaTeX 'in nasıl çalıştığı konusunda kabaca bir fikir sahibi olursunuz ve kitabın geri kalanını anlamanız daha kolay olur.

1.1 Meselenin Özü

1.1.1 \TeX

\TeX , Donald E. Knuth [2] tarafından yazılmış bir bilgisayar programıdır. Amacı metinleri ve matematik formülleri dizmektir. Knuth 70li yıllarda, kitap basımında kullanılmaya başlayan otomasyon teknolojilerinin özellikle kendi kitaplarının ve makalelerinin baskı kalitesini nasıl düşürmekte olduğunu görünce, bu teknolojinin imkanlarını araştırmak üzere 1977 yılında \TeX dizgi programını yazmaya başladı. Bugün kullandığımız şekliyle \TeX 1982 yılında ortaya çıktı, daha sonra 8-bitlik karakter işleyebilen ve diğer dilleri de destekleyen sürümü 1989 yılında yapıldı. \TeX her türlü bilgisayarda çalışabilen ve hatası yok denecek kadar az olan, sağlam ve güvenilir bir programdır. \TeX 'in sürüm sayısı π sayısına her defasında bir ondalık eklenerek ifade edilir, bugün 3.141592 sayılı sürümdedir.

\TeX telaffuz edilirken Almancadaki “Ach,”¹ sesine, veya Arapçadaki “Khalid” isminde olduğu gibi, genizden söylenen “kh” sesine benzeterek söylenir. Bu ses Yunan alfabesindeki X veya “ki” harfinden esinlenmiş olup, Yunanca texnologia (teknoloji) sözcüğünün ilk hecesidir. Bilgisayardaki ASCII ortamında bunun adı **TeX** olur.

¹Aslında almancada “ch” için iki farklı telaffuz vardır ve “Pech” sözcüğünde olduğu gibi yumuşak telaffuz edileceği düşünülebilir. Bu konuda bilgisine başvuru Knuth, Alman Wikipedia ansiklopedisine şunları yazdı: *İnsanların \TeX 'i beğendikleri gibi telaffuz etmelerine itirazım yok. . . Rusçada da buna benzer bir ‘tyekh’ sesi var. Fakat, en doğru telaffuzun, ‘ach’ veya ‘Loch’ sesindeki sertliği veren Yunancada olduğunu düşünüyorum.*

1.1.2 L^AT_EX

L^AT_EX önceden belirlenmiş profesyonel bir şablon çerçevesinde, yazarlara dokümanlarını en yüksek dizgi kalitesinde hazırlama ve yazıcılarda basma olanağı veren bir makro pakettir. Leslie Lamport [1] tarafından yazılmış olan L^AT_EX, formatlayıcı dizgi aracı olarak T_EX programını kullanır. Günümüzde Frank Mittelbach tarafından güncelleştirilmektedir.

L^AT_EX “Ley-tekh” veya “La-tekh” olarak telaffuz edilir. Bir ASCII ortamında L^AT_EX’e atıfta bulunmak isterseniz, klavyeden L^AT_EX yazarak girersiniz. L^AT_EX 2_ε ise “Ley-tekh iki e” olarak telaffuz edilir ve klavyeden L^AT_EX2_ε yazarak girilir.

1.2 Temel Kavramlar

1.2.1 Yazar, Tasarımcı ve Dizgici

Bir eseri yayımlamak isteyen yazar, onu daktilo edip bir kopyasını yayıncı şirkete verir. Şirketin kitap tasarımcısı bunun için bir plan yapar (sayfa genişliği, marjlar, kullanılan yazı tipi, başlıklardan önce ve sonraki boşluklar, ...). Tasarımcı bu bilgileri de ekleyip bir matbaaya gönderir, orada dizgici bu plana uyarak kitabı dizer.

Canlı kanlı bir dizgici, yazarın bu kitabı kaleme alırken ne düşündüğünü tahmin etmeye çalışır ve bölüm başlıkları, atıflar, örnekler, formüller, vs. dizerken, elindeki taslağın içeriğine ve kendi mesleki tecrübesine dayanarak karar verir.

L^AT_EX ortamında ise, tasarımcının rolünü L^AT_EX, dizgicinin rolünü ise T_EX üstlenir. Ama, L^AT_EX sadece bir bilgisayar programıdır ve bu yüzden yol gösterilmeye muhtaçtır. Yazar kitabının mantıksal kurgusu hakkında biraz daha fazla bilgi vermelidir. Bu bilgiler metin içinde “L^AT_EX komutları” olarak yer alırlar.

Bu yaklaşım *MS Word* veya *Corel WordPerfect* gibi modern kelime işlemcilerinin WYSIWYG² yaklaşımından farklıdır. O programlarda yazar metni girerken bilgisayarla karşılıklı etkileştiği için, yazıcıda çıkacak olanın aynısını ekranda görür.

L^AT_EX’de metni yazarken kağıt üzerinde çıkacak olan halini göremezsiniz, ancak dosyanızı L^AT_EX ile işlemden geçirdikten sonra ekranda gözleyebilirsiniz. Bu gözlemde bulacağınız hataları düzelttikten sonra, en son halini yazıcıya gönderebilirsiniz.

1.2.2 Şablon Tasarımı

Kitap tasarımı bir zenaattır. Acemi yazarların kitaplarını formatlarken yaptıkları en büyük hata bunu sadece bir estetik sorunu olarak görmeleri, yani

²What you see is what you get (Ne görüyorsan o).

“Bir doküman göze hoş görünüyorsa iyi tasarlanmış sayılır,” diye düşünceleridir. Oysa, bir doküman okunmak için vardır, bir tablo gibi duvara asılmak için değil. Okunabilir ve anlaşılabilir olmak görünüm güzelliğinden daha önemli olmalıdır. Örneğin:

- Kitaptaki bölüm ve kısım başlıklarının yazıtipi, punto büyüklüğü ve numaralanmış olması, okuyucunun bölüm ve kısımları açık olarak görmesini sağlar.
- Satır uzunluğu okuyucunun gözünü yormayacak kadar kısa, ama sayfayı güzelce dolduracak kadar uzun olmalıdır.

WYSIWYG sistemlerinde okuyucu göze hoş gelen dokümanları kolayca, ama bazan tutarsızca oluşturabilir. \LaTeX bu tür formatlama hatalarına izin vermemek için, yazarı dokümanın *mantıksal* yapısını baştan deklare etmeye zorlar. Bunun üzerine \LaTeX en uygun şablonu seçer.

1.2.3 Ne Getirir, Ne Götürür?

Ne zaman WYSIWYG kullanan insanlar \LaTeX kullanıcıları ile bir araya gelseler, hangisinin daha üstün olduğu konusunda bir tartışma başlar. Böyle durumlarda yapabileceğiniz en iyi şey fazla üstelememektir. Fakat, bazan da tartışmayı kolaylaştırmaktan kaçamayabilirsiniz . . . Size, böyle durumlar için biraz cephane vereyim. \LaTeX 'in diğer kelime işlemcisi programlara göre üstün olduğu noktalar şunlardır:

- Profesyonelce tasarlanmış şablonlar kullanıldığı için, dokümanlar “matbaadan çıkmış gibi” olurlar.
- Matematik formüllerini dizmek kolay anlaşılır bir yöntemle sağlanır.
- Kullanıcının, dokümanın mantıksal yapısını belirten ve anlaşılması kolay komutlardan sadece birkaç tanesini öğrenmesi yeterlidir. Ondan sonra, dokümanın alacağı şekli dert etmesine pek gerek kalmaz.
- Dipnotlar, referanslar, içindekiler sayfası, kaynakça gibi karmaşık yapılar çok daha kolay üretilirler.
- Temel \LaTeX programının doğrudan desteklemediği diğer dizgi araçları, ücretsiz ilave paketler halinde kullanıma hazırdır. Örneğin, POSTSCRIPT formatında grafikleri metin içine yerleştirmek, veya belli bir standarda göre kaynakça hazırlamak için özel paketler vardır. Bu ilave paketler *The \LaTeX Companion* [3]'da tanıtılmaktadır.
- \LaTeX yazarları kesin belirtilmiş yapıda dokümanlar yazmaya zorlar, çünkü \LaTeX 'in kendisi böyle, yapılar belirleyerek çalışır.

- \LaTeX 2 ϵ programı taşınabilir ve ücretsizdir. Her işletim sisteminde ve her donanımda çalışır.

\LaTeX 'in dezavantajlı yönleri de vardır. Bunları söylemek bana zor geliyor, ama değişik ağızlardan duyabileceğiniz eleştiriler şunlar olabilir:

- \LaTeX ile çalışmak ruhunu satmış olanlar için zordur.
- Önceden tasarlanmış bir doküman şablonu içinde bazı parametreleri değiştirmek kolay olabilir, ama yepyeni bir şablon tasarlamak zor ve zaman alıcıdır.³
- Evinizdeki kedi, siz ne kadar yardımcı olsanız da, Kuantum teorisini öğrenemez.

1.3 \LaTeX Girdi Dosyaları

\LaTeX en basit olan ASCII yapısındaki metin dosyaları üzerinde çalışır. Uzantısı `.tex` olan bu dosyaları herhangi bir metin yazıcı program ile yazabilirsiniz. Bu dosyaların içinde, sizin yazdığınız metnin yanısıra, \LaTeX 'in bu metni hangi işlemlere tabi tutacağını bildiren komutlar bulunur.

1.3.1 Boşluklar

İster boşluk çubuğuyla girdiğiniz “akboşluk” karakteri, ister sekme (tab) tuşuyla girdiğiniz aralık, \LaTeX tarafından “boşluk” olarak kabul edilir. *Ardarda çok sayıda* boşluk da olsa, yine bir “boşluk” gibi işlem görür. Bir satırın başındaki ilk boşluk dikkate alınmaz. Keza, bir kez atılan satırbaşı (şaryo) tuşu da “akboşluk” sayılır.

İki metin satırı arasındaki boş satır paragraf sonu demektir. *Ardarda çok sayıda* boş satır yine bir boş satır sayılır. Aşağıdaki örnekte bunu görebilirsiniz. Sol tarafta girdi dosyasındaki metin, sağ tarafta onun formatlanmış hali gösterilmiştir.

Bir kelimededen sonra bir veya
birçok boşluk olması
aynı kapıya çıkar.

Boş bir satır yeni bir
paragraf başlatır.

Bir kelimededen sonra bir veya birçok boşluk
olması aynı kapıya çıkar.

Boş bir satır yeni bir paragraf başlatır.

³Bu sorunun yeni hazırlanmakta olan \LaTeX 3 sürümünde giderileceği söyleniyor.

komutları kullanan iki örnek verelim. Bunları anlayamazsanız, şimdilik dert etmeyin, daha sonra hepsini anlatacağız.

```
Yaslan güzelim,  
kalbime \textsl{yaslan}.
```

Yaslan güzelim, kalbime *yaslan*.

```
Burada yeni  
bir satır başlatın.  
Burada!\newline Teşekkürler!
```

Burada yeni bir satır başlatın. Burada!
Teşekkürler!

1.3.4 Yorum Satırları

\LaTeX bir girdi dosyasını işleme tabi tutarken, metin içinde bir % sembolü gördüğü anda, o satırın geri kalanındaki herşeyi işlemeden bırakır ve bir sonraki satırın başına geçip devam eder.

Bu özellik, girdi dosyası içine, baskıda gözükmeyen, sadece yazarın kendine hatırlatma amacıyla yaptığı yorumları eklemekte kullanılır.

```
İşte bir % saçmalık.  
% Buraya bir şekil koyalım.  
örnek: Slo%  
vakyalılaştıramadık%  
larımız
```

İşte bir örnek: Slovakyalılaştıramadıklarımız

Bu % sembolü uzun girdi satırlarını, kağıt üzerindeki sürekliliğini bozmadan, görünüşte bölmekte kullanılır.

Daha uzun yorumlar katmak istiyorsanız, *verbatim* paketinde bulunan *comment* komutunu kullanabilirsiniz. Bunun için, dokümanınızın başlangıç kısmına `\usepackage{verbatim}` satırını eklemeniz gerekir.

```
Dokümanınıza yorum katmak  
için, bu da  
\begin{comment}  
saçma bir örnektir,  
ama işe yarar.  
\end{comment}  
faydalı bir örnektir.
```

Dokümanınıza yorum katmak için, bu da faydalı bir örnektir.

Fakat, bu yorumlar karmaşık ortamlarda, örneğin matematik ortamında geçerli değildirler.

1.4 Girdi Dosya Yapısı

\LaTeX bir girdi dosyasını işlemeye başladığında, onun belli bir yapı izlediğini görmek ister. Bu yüzden, her girdi dosyası şu komutla başlamalıdır:

```
\documentclass{...}
```

Bu komut ne tür bir doküman yazmak niyetinde olduğunuzu belirtir. Ondan sonra, dokümanı bir bütün olarak etkileyecek diğer komutları ekleyebilir, veya \LaTeX sistemine yeni olanaklar katan paketleri yükleyebilirsiniz. Bir paketi yüklemek için şu komutu kullanırsınız:

```
\usepackage{...}
```

Tüm bu eklemeler ve değiştirmeler bittikten sonra,⁵ kendi eserini yazmaya başlamak üzere, şu komutu girersiniz:

```
\begin{document}
```

Buradan itibaren kendi metnini, bazı yararlı \LaTeX komutları da ekleyerek, girersiniz. Yazmanız bittiğinde, dokümanın sona erdiğini şu komutla bildirirsiniz:

```
\end{document}
```

Bu komut \LaTeX 'e paydos etmesini söyler. Daha sonra gireceğiniz satırları \LaTeX dikkate almaz.

Şekil 1.1 minimum bir $\LaTeX 2_{\epsilon}$ dosyasının içeriğini gösteriyor. Biraz daha karmaşık girdi dosyası örneği Şekil 1.2 de gösterilmiştir..

1.5 Tipik Bir Doküman Yazımı

Herhalde artık, sayfa 7 deki minimum \LaTeX girdi dosyasını deneyip sonucu görmek için sabırsızlanıyorsunuzdur. O zaman, şu bilgilerin size yardımı olur:

\LaTeX 'in kullanıcıya sunduğu bir grafik arabirimi veya gösterişli menüleri yoktur. O sadece sizin girdi dosyanızı işleyip durur. Bazı \LaTeX kurulumlarında bir düğmeye basıp girdi dosyanızı işlemesini sağlayabilirsiniz. Diğer sistemlerde \LaTeX 'e ne yapması gerektiğini klavyeden girmek zorundasınızdır. Windows sisteminde komut isteme penceresi açarsınız. Burada vereceğimiz bilgiler böyle klavyeden girerek çalışılan sistemler içindir. Fakat, her halukarda bilgisayarınızda kurulu ve çalışan bir \LaTeX programı mevcut olduğunu varsayıyorum.

⁵ \documentclass komutu ile $\begin{document}$ komutu arasında kalan bölgeye *sahanelik* (preamble) adı verilir.

```
\documentclass{article}
\begin{document}
Azı karar çoğu zarar.
\end{document}
```

Şekil 1.1: Minimum bir \LaTeX dosyası.

1. Önce \LaTeX girdi dosyanızı yazıp oluşturun. Bu dosya sadece ASCII karakterlerden oluşan bir metin dosyası olmalıdır. Unix işletim sisteminde tüm metin yazıcılar (editörler) bunu yapabilirler. Windows sisteminde, dosyanızı ASCII veya *sade metin* formatında saklayabilen bir metin yazıcısı kullanın. Dosyanıza bir ad vermek gerektiğinde, uzantısının mutlaka `.tex` olmasına dikkat edin.
2. Girdi dosyanıza \LaTeX ile işlem yapın (Buna ‘derlemek’ veya ‘latekslemek’ de denir):

```
latex makale.tex
```

Başarılı olursanız, sabit diskte uzantısı `.dvi` olan bir dosya oluşacaktır. İçindekiler sayfasını ve atıfları oluşturabilmek için aynı dosyayı birden fazla latekslemeniz gerekebilir. Girdi dosyanızda bir hata varsa, \LaTeX bu durumu size bildirir ve yarıda bırakıp çıkar. `kontrol-D` tuşuna basarak komut isteme satırına geri dönün.

3. Şimdi, DVI dosyasından önizleme yapabilir, yani dokümanınızın kağıt üzerinde alacağı şekli görebilirsiniz. Bu birkaç türlü yapılabilir. Dosyayı ekranda görmek için şu satırı girin:

```
xdvi makale.dvi &
```

Bu komut sadece Unix sisteminde X11 arabirimi için geçerlidir. Win-

```

\documentclass[a4paper,11pt]{article}
% Makale başlığını tanımla
\author{M.~Yılmaz}
\title{Fillerin Anatomisi}
\begin{document}
% başlığı oluştur
\maketitle
% İçindekiler sayfasını buraya ekle
\tableofcontents
\section{Afrika Filleri}
Bu çalışmamızda değişik bölge fillerinin \ldots{}
\section{Hindistan Filleri}
\ldots{} burada bitiriyoruz.
\end{document}

```

Şekil 1.2: Gerçekçi bir dergi makalesi örneği. Bu örnekte gördüğümüz tüm komutlar daha sonra açıklanacaktır.

dows sisteminde yap önizleyicisini kullanabilirsiniz.

4. DVI dosyanızı POSTSCRIPT formatına çevirip Ghostscript programı ile de ön izleme yapabilirsiniz. Bunun için şu satırı girin:

```
dvips -Pcmz makale.dvi -o makale.ps
```

Eğer şansınız varsa, L^AT_EX kurulumunda, .dvi dosyanızı pdf formatına çeviren bir program da olabilir. Bunu şu komutla yaparsınız:

```
dvipdf makale.dvi
```

1.6 Bir Dokümanın Yerleşim Planı

1.6.1 Doküman Sınıfları

L^AT_EX herhangi bir girdi dosyasını derlerken bilmesi gereken ilk şey, yazarın ne tür bir doküman yazmak istediğidir. Bu da `\documentclass` komutuyla belirtilir:

```
\documentclass[opsiyonlar]{sınıf}
```

Burada *sınıf* dokümanın cinsini belirtir. Bu kitapta anlatılacak olan doküman türleri Tablo 1.1 de gösterilmiştir. L^AT_EX sürümü içinde mektup, asetat gibi diğer doküman türleri de mevcuttur. *opsiyonlar* kısmına, o doküman sınıfında bir takım değişiklikler yapmak istiyorsanız, bazı parametreler ekleyebilirsiniz. Her opsiyon virgülle ayrılmalıdır. Standard bir doküman sınıfında bulunabilecek opsiyonlar Tablo 1.2 de gösterilmiştir.

Örnek: Bir L^AT_EX girdi dosyası şu satırla başlayabilir

```
\documentclass[11pt,a4paper,twoside]{article}
```

Bu komut L^AT_EX'e, *article* (makale) sınıfından bir dokümanı *onbir puntoluk* harflerle, *A4 boyutlu kağıda* ve *iki taraflı* basacak şekilde dizmesini söyler.

1.6.2 Paketler

Bir dokümanı yazarken temel L^AT_EX programının halledemeyeceği bazı istekleriniz olabilir. Örneğin, içine grafik eklemek, bazı yerlerde renkli harfler kullanmak veya bir bilgisayar programını dizmek isteyebilirsiniz. Bu durumda

L^AT_EX'in yeteneklerini güçlendirmeniz gerekir. Bu tür ek işler paketlerle yapılır. Bir paket programı şu komutla etkin hale getirilir:

```
\usepackage[opsiyonlar]{paket}
```

Burada *paket* kullanılmak istenen paketin adı, *opsiyonlar* ise paketin marifetlerini harekete geçiren anahtar kelimelerdir. Bazı paketler temel L^AT_EX kurulumu ile doğrudan gelir (Bakınız Tablo 1.3), diğerlerini ayrıca elde etmek gerekir. Çalıştığınız yerdeki kurulu paketlerin listesini *Local Guide* [5] (Yerel Rehber) de bulabilirsiniz. L^AT_EX paketleri konusunda en temel kaynak *The L^AT_EX Companion* [3] kitabıdır. Orada yüzlerce paketin açıklaması yanısıra, L^AT_EX 2_ε için kendi paketlerinizi nasıl yazabileceğinizi de öğrenirsiniz.

Modern T_EX sürümleri kurulurken çok sayıda paketi de otomatik olarak yükler. Unix sisteminde çalışıyorsanız, `texdoc` komutunu yazarak paket bilgilerine erişebilirsiniz.

1.6.3 Sayfa Biçimleri

L^AT_EX önceden tanımlanmış üç türlü sayfa biçimi kullanır. Bunların tepe-lik/diplik bölgeleri farklı olur. Hangisinin kullanılacağı şu komutla belirtilir:

```
\pagestyle{biçim}
```

Buradaki *biçim* parametresi için önceden tanımlanmış seçenekler Tablo 1.4

Tablo 1.1: Doküman Sınıfları.

article bilimsel dergi makaleleri, sunumlar, kısa raporlar, program dokümanları, davet bildirimleri, ...

proc konferans bildirimleri için, article sınıfından esinlenmiştir.

minimal olabilecek en küçük sınıf. Yaptığı iş sadece sayfa büyüklüğünü ve yazıtıpini belirtmektir. Hata ayıklamakta kullanılır.

report çok sayıda bölümden oluşan uzun raporlar, küçük kitapçıklar, doktora tezleri, ... için.

book gerçek bir kitap için.

slides asetatlar için. Bu sınıfta büyük, serifsiz harfler kullanılır. Bundan daha iyisi için FoilT_EX^a paketine bakabilirsiniz.

^amacros/latex/contrib/supported/foiltex

Tablo 1.2: Document Sınıfı Opsiyonları.

<code>10pt, 11pt, 12pt</code>	Dokümanın ana yazıtipi puntosunu belirler. Bir değer belirtilmemişse, <code>10pt</code> punto varsayılır.
<code>a4paper, letterpaper, ...</code>	Kağıt boyutunu belirler. Varsayılan boyut Amerikan standardı olan <code>letterpaper</code> dir. Bunlara ek olarak <code>a5paper, b5paper, executivepaper</code> , ve <code>legalpaper</code> seçilebilir.
<code>fleqn</code>	Formülleri ortada yazmak yerine, sola bitişik dizer.
<code>leqno</code>	Formül numaralarını sağ yerine sol tarafa koyar.
<code>titlepage, notitlepage</code>	Doküman başlığını attıktan sonra yeni bir sayfa açıp açmayacağını belirler. Varsayılanına göre, <code>article</code> sınıfı yeni sayfa açmaz, <code>report</code> ve <code>book</code> sınıfları açar.
<code>onecolumn, twocolumn</code>	Dokümanın tek sütun veya çift sütun dizileceğini belirtir.
<code>twoside, oneside</code>	Dokümanın kağıdın hep tek tarafına mı yoksa iki tarafına mı basılacağını belirtir. Varsayılan durum <code>article</code> ve <code>report</code> sınıfları için <code>single sided</code> (tek taraflı), <code>book</code> sınıfı için <code>double sided</code> (çift taraflı) olur. Dikkat edin, <code>twoside</code> opsiyonu yazıcıyı kağıdı çift taraflı basmaya zorlayamaz, çıkan kağıdı tekrar beslemek sizin işinizdir.
<code>landscape</code>	Dokümanı enine tutulmuş kağıda basılmak üzere hazırlar.
<code>openright, openany</code>	Dokümanda bölümleri hep sağ sayfalardan veya ilk gelen boş sayfadan başlatır. Bu opsiyon <code>article</code> (makale) sınıfında bölümler olmadığı için geçersizdir. <code>report</code> sınıfında ilk boş sayfadan başlanacağı, <code>book</code> sınıfında da hep sağ sayfadan başlanacağı varsayılır.

Tablo 1.3: L^AT_EX ile Gelen Bazı Paketler.

<code>doc</code>	L ^A T _E X programlarının dokümantasyonunu çıkarır. <code>doc.dtx</code> dosyasında açıklanmıştır. ^a
<code>exscale</code>	Matematik formüllerinde kullanılan ilave yazıtiplerinin küçültülmüş sürümlerini kullanmanızı sağlar. <code>ltxscale.dtx</code> dosyasında açıklanmıştır.
<code>fontenc</code>	L ^A T _E X'in hangi yazıtipi kodlama grubunu kullanacağını belirtir. <code>loutenc.dtx</code> dosyasında açıklanmıştır.
<code>ifthen</code>	Provides commands of the form 'if... then do... otherwise do...' Described in <code>ifthen.dtx</code> and <i>The L^AT_EX Companion</i> [3].
<code>ifthen</code>	'eğer... ise ... yi değilse ... yi yap.' türü komutları kullanmanızı sağlar. <code>ifthen.dtx</code> ve <i>The L^AT_EX Companion</i> [3]'da açıklanmıştır.
<code>latexsym</code>	L ^A T _E X'in sembol fontlarını kullanabilmeyi sağlar. <code>latexsym.dtx</code> ve <i>The L^AT_EX Companion</i> [3]'da açıklanmıştır.
<code>makeidx</code>	Dizin oluşturmak için gerekli komutlara erişim sağlar. Bu kitabın <code>refsec:indexing.</code> kısmında ve <i>The L^AT_EX Companion</i> [3]'da açıklanmıştır.
<code>syntonly</code>	Bir dokümanı dizmeden sadece işler.
<code>inputenc</code>	Değişik dillerde harf girmek için ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM kodlama sistemlerinden birini, Apple Macintosh, Next, ANSI-Windows veya kullanıcının belirlediği kodsayıları seçer. <code>inputenc.dtx</code> de açıklanmıştır.

^aBu dosya sisteminizde kurulmuş olmalıdır ve `latex doc.dtx` komutu girdiğinizde seçtiğiniz bir klasörde `dvi` dosyasını üretir. Aynı şey bu tablodaki diğer dosyalar için de geçerlidir.

de gösterilmiştir.

Tablo 1.4: Önceden Tanımlanmış L^AT_EX Sayfa Biçimleri.

plain (sade) sayfanın diplik bölgesinin orta yerine sayfa numarası koyar. Varsayılan sayfa biçimi budur.

headings (tepelik) her sayfanın tepelik bölgesine o bölümün adını ve dokümanın sayfa numarasını koyar, dipliği boş bırakır. (Bu kitapta kullandığımız biçim budur.)

empty (boş) tepelik ve dipliği boş bırakır.

Bazan dokümandaki bir sayfayı diğerlerinden farklı dizmek gerekebilir. Bunun için şu komut kullanılır:

```
\thispagestyle{biçim}
```

Kendi tepelik ve diplik tasarımlarınızı da oluşturabilirsiniz. Bunun için Sayfa 67'deki Bölüm 4.4 veya *The L^AT_EX Companion* [3]'aa bakınız.

1.7 Karşılaşabileceğiniz Dosya Türleri

L^AT_EX ile çalışırken, kısa süre içinde sabit diskinizde değişik uzantılı pek çok dosya türü oluşur ve ne işe yaradıklarını bilemezsiniz. Aşağıdaki listede T_EX programıyla çalışırken karşılabileceğiniz dosya türleri açıklanmıştır. Ama, bu listenin eksiksiz olduğunu söyleyemem, burada göremediğiniz bir uzantı türüne raslarsanız bana haber ulaştırın.

- .tex** L^AT_EX veya T_EX girdi dosyası. `latex` komutu bu uzantılı dosyaları işler.
- .sty** L^AT_EX Makro biçim paketi. Bu tür dosyaları L^AT_EX dokümanınız içine `\usepackage` komutuyla yüklersiniz.
- .dtx** Dokümantasyonlu L^AT_EX paketleri. L^AT_EX biçim dosyalarının en yaygın dağıtım formatı. Bir `.dtx` dosyasını derlediğinizde, içindeki L^AT_EX paketinin dokümantasyonlu bir makro programını elde edersiniz.
- .ins** Aynı isimli `.dtx` dosyalarını kuran dosya. İnternette bir L^AT_EX paketi indirdiğinizde, normal olarak bir `.dtx` ve bir de `.ins` dosyası gelir. Bu `.ins` dosyasını L^AT_EX'den geçirirseniz, `.dtx` dosyasını çözümler.

- .cls** Sınıf dosyaları dokümanınızın sınıfını tanımlar. Hangisini seçeceğinizi `\documentclass` komutuyla bildirirsiniz.
 - .fd** Yazıtipi tanımlama dosyaları. \LaTeX 'e yeni yazıtiplerini tanıtır.
- Aşağıdaki uzantılı dosyalar da girdi dosyanızı latekslediğinizde oluşurlar:
- .dvi** Aygıttan bağımsız dosya (device independent). Girdi dosyasını \LaTeX ile derlemenin başlıca sonucu bu dosyadır. İçeriğini bir DVI önizleyici programla görebilir veya, `dvips` komutuyla yazıcıya veya benzer aygıtlara gönderebilirsiniz.
 - .log** En son lateksleme işlemi sırasında olan bitenin kayıt defteri.
 - .toc** Tüm bölüm ve kısım başlıklarını biriktirir. Bir sonraki derleme sırasında, İçindekiler tablosunu oluşturmakta kullanılır.
 - .lof** Bu da `.toc` gibi, ama Şekiller tablosu içindir.
 - .lot** Tablolar listesi içindir.
 - .aux** Bir derlemeden diğerine bilgilerin aktarıldığı dosya. Yaptığı en önemli iş, doküman içindeki denklem veya sayfa atıflarını saklamaktır.
 - .idx** Belgenizde dizin oluşturacaksınız, dizine girmek üzere işaretlediğiniz kelimeleri saklar. Bu dosyayı `makeindex` programından geçirirsiniz. Dizinleme hakkında daha fazla bilgi için Sayfa 66 deki Kısım 4.3'ya bakınız.
 - .ind** Dizin dosyasının `makeindex` programından geçmiş hali. Bir dahaki derleme sırasında dokümanınızda dizin oluşturur.
 - .ilg** `makeindex` programı sırasında olup bitenin kayıt defter.

1.8 Büyük Projeler

Büyük dokümanlar üzerinde çalışırken, girdi dosyanızı çok sayıda parçalar halinde ayırıp hazırlamak isteyebilirsiniz. Bunun için \LaTeX 'in size sağladığı iki komut vardır:

```
\include{ekdosya}
```

Bu komutu doküman metni içine koyduğunuz yerde `ekdosya.tex` adlı dosyanın içeriği eklenmiş olur. Fakat dikkat, \LaTeX bu eklenen dosyayı işlemeyen önce yeni bir sayfa açar.

İkinci komut sadece dokümanın sahanlığında yeralabilir:

```
\includeonly{ekdosya,ekdosya,...}
```

Bu komut \LaTeX 'e sadece liste halinde belirttiği *ekdosyaların* eklenebileceğini bildirir. Bu komut sahanlıkta bulduktan sonra, metin içinde bir yerdeki `\include{ekdosya}` komutundaki ekdosya adı ancak bu listede yer alıyorsa işleme tabi tutulur. Bu komuttaki dosya adları ile virgüller arasında hiçbir boşluk bulunmamasına dikkat edin.

`\include` komutuyla eklenen metin yeni bir sayfadan başlar. Bu bir bakıma yararlıdır, çünkü `\includeonly` komutuyla belirtilen dosyalardan bazılarını eklemesiniz bile, sayfa bitiş yerleri değişmeyecektir. Bazen da yeni bir sayfadan başlaması istenmeyebilir. Bu durumda şu komutu kullanırsınız:

```
\input{filename}
```

Bu komut sadece belirtilen ek dosyayı, yeni sayfa açmadan, bulunduğu yerden itibaren ekler.

\LaTeX 'in dokümanınızı şöyle çabucak bir yoklamasını isterseniz `syntonly` paketini kullanırsınız. Bu, DVI çıktı dosyası üretmeden, dokümanınızdaki komutları doğru girip girmediğinizi \LaTeX 'e kontrol ettirip çıkar. Bu size zaman kazandırabilir. Kullanımı çok basittir:

```
\usepackage{syntonly}
\syntonly
```

Sonra, sayfaları dizmesini istiyorsanız, ikinci satırdaki komutun önüne bir yüzde işareti koyarak devreden çıkarırsınız.

Bölüm 2

Dizgi

Bir önceki bölümü okuduktan sonra, bir \LaTeX dokümanının temel yapısını kavramış olmalısınız. Bu bölümde gerçek bir doküman yazımı için bilmeniz gereken diğer konuları anlatacağım.

2.1 Metin ve Dil Yapısı

Bir metin yazmanın temel amacı bir düşünceyi veya bir bilgiyi okuyucuya aktarmaktır. Eğer bu fikirler iyi organize edilmişse okuyucu metni daha iyi anlayacak, eğer içeriğin mantıksal ve dilbilgisel yapısına uygun bir metin dizgisi varsa, bu yapıyı daha iyi görecektir.

\LaTeX diğer dizgi sistemlerinden farklıdır, sizin ona metnin mantık ve dilbilgisi yapısını söylemeniz yeterlidir. Oradan itibaren metin dizgisini, doküman sınıfındaki ve diğer biçim dosyalarındaki “kurallar” uyarınca, kendisi yapar.

\LaTeX 'de (ve matbaa dizgisinde) en önemli metin birimi paragraftır. Buna “metin birimi” diyoruz, çünkü tutarlı bir düşünceyi veya bilgiyi yansıtan dizgi bütünü paragraftır. Aşağıdaki kısımlarda, örneğin `\` komutuyla nasıl satırı yarıda keseceğinizi, veya boş bir satır girerek nasıl paragrafı sonlandıracağınızı öğreneceksiniz. O halde, yeni bir düşünceye başlıyorsanız, yeni bir paragraf açmalısınız, aksi takdirde satır yazmaya devam edersiniz. Paragraf kesme konusunda kuşkunuz varsa, metninizin bir düşünce iletme aracı olduğunu hatırlayın. Yeni bir paragrafa başlamışsınız, ama bir önceki düşünce devam ediyorsa, paragraf kesmeyi kaldırın. Aynı paragraf içinde ikinci bir fikre geçmişseniz, o paragrafı ikiye bölün.

Bazı kişiler iyi kesilmiş paragrafların önemini hafife alıyorlar. Çoğu kişi de paragraf kesmenin anlamını bilmiyor ve, özellikle \LaTeX 'te bilmeden paragraf kesiyorlar. Metinde denklemler yer alıyorsa, bu hatayı yapmak daha da kolaydır. Aşağıdaki örneklerle bir bakın ve denklemlerden önce ve sonra, neden bazılarında boş satır (paragraf sonu) bırakılıp diğerlerinde bırakılmadığını anlamaya çalışın. (Bu örneklerdeki komutları henüz anlamıyorsanız,

o zaman geçin, bu ve sonraki bölümü okuyup bu örneklere geri dönün.)

```
% Örnek 1
\ldots Einstein'ın meşhur formülü
\begin{equation}
e = m \cdot c^2 \ ; \ ,
\end{equation}
pek çok kişinin bildiği ama pek az kişinin anladığı
bir fizik formülüdür.
```

```
% Örnek 2
\ldots buradan Kirchhoff'un akım kanunları elde edilir:
\begin{equation}
\sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
```

Kirchhoff'un voltaj kanunlarına gelince \ldots

```
% Örnek 3
\ldots nin pekçok uygulaması vardır.
```

```
\begin{equation}
I_D = I_F - I_R
\end{equation}
denklemini çok farklı bir transistör modelinin
temelini oluşturur. \ldots
```

Bir sonraki daha küçük metin birimi cümledir. İngilizce metinlerde cümleyi bitiren noktadan sonra bir boşluk bırakılır, baş harf kısaltmalarından sonra bırakılmaz. \LaTeX hangisini istediğinizi kestirmeye çalışır. Eğer \LaTeX yanlış tahmin ederse, ona ne istediğinizi söylemeniz gerekir. Bu nokta bölüm sonuna doğru açıklanacaktır.

Metin yapılandırması cümlelerin parçalarına kadar uzanır. Bazı dillerin noktalama kuralları çok karmaşıktır, fakat diğer dillerde (İngilizce ve Almanca da dahil), virgülün ne iş yaptığını biliyorsanız noktalamanın çoğunu doğru yaparsınız: virgül dil akışında kısa bir duraklamadır. Nereye virgül koyacağımızdan emin değilseniz, o cümleyi yüksek sesle okuyun ve her virgülden biraz duraklayın. Nerede tuhaf kaçırıyorsa, oradaki virgülü kaldırın; başka bir yerde nefes alma ihtiyacı duymuşsanız, oraya bir virgül koyun.

Son olarak, bir metnin paragrafları daha yüksek bir düzeydeki mantığa göre, bölümler, kısımlar, altkısımlar, vs. şeklinde yapılandırılmalıdır. Fakat, örneğin `\section{Metin ve Dil Yapısı}` yazmanın dizgideki etkisi

o kadar açıktır ki bu yüksek düzeyli yapıları nasıl oluşturacağımızı kolayca görebilirsiniz.

2.2 Satır ve Sayfa Kesme

2.2.1 Çift Yaslanmış Paragraflar

Kitaplarda satırlar hep aynı uzunlukta dizilirler. \LaTeX bir paragrafın içeriğindeki kelimeler arasındaki boşlukları ve satır kesmelerini öyle bir ayarlar ki paragraflar iki tarafa da yaslanmış ve eşit uzunlukta satırlardan oluşur. Bu amaçla, gerekirse bir satıra sığmayan kelimelerde heceler bölünebilir. Paragrafların nasıl dizildiği doküman sınıfına bağlıdır. Normalde bir paragrafın ilk satırı biraz içerden başlar ve iki paragraf arasındaki boşluğa ilave yapılmaz. Daha fazla bilgi için Kısım 6.3.2 ye bakın.

Bazı özel durumlarda \LaTeX 'in bir satırı yarıda kesmesi şu komutla sağlanır:

```
\ \ veya \newline
```

ve yeni bir paragraf açmadan satırbaşı yapılır. Keza,

```
\ \*
```

komutu, zorunlu kesilen satırdan hemen sonra sayfa kesilmesini önler.

Belli bir yerde sayfa kesilmesi istemi şu komutla sağlanır:

```
\newpage
```

ve yeni bir sayfa başlar. Benzer diğer komutlar

```
\linebreak[n], \nolinebreak[n], \pagebreak[n], \nopagebreak[n]
```

sırasıyla, satır kes, satır kesme, sayfa kes ve sayfa kesme komutları olup, yazarın bu kesimleri daha ince yapabilmesini sağlarlar. Buradaki opsiyonel n parametresi 0 ile 4 arasında bir sayıdır. n sayısı 4 den küçük seçilirse, sonuç kötü olduğunda \LaTeX 'in sizin isteğinizi gözardı etmesine izin verir. Buradaki “linebreak” (satır kes) komutuyla “newline” (yeni satır) komutu arasında ince bir fark vardır. “linebreak” komutu verdiğinizde \LaTeX yarım kalan satırı hala sağa yaslamaya çalışır, “newline” komutunda ise, hemen alt satıra geçer. Benzer bir fark “pagebreak” ile “newpage” komutları arasında vardır.

\LaTeX daima mümkün olan en iyi yerde satırı kesmeye çalışır. Kendi yüksek standardına ulaşan bir kesim yeri bulamazsa, o satırın sağda dışarı fırlamasına yol açar. Bu durumda \LaTeX (“overfull hbox”) mesajıyla şikayet

edip derlemeye devam eder. Bu sorun genellikle bir kelimenin hece bölmesini beceremeyince oluşur.¹

\LaTeX 'in kendi standartlarını düşürmesini istiyorsanız, `\sloppy` (gevşek) komutunu verebilirsiniz. Böylece kelimeler arasındaki boşlukları daha da artırarak satırın sağa fırlamasını önler. Bu durumda kullanıcıyı (“underfull hbox”) mesajıyla uyarır. Böyle durumlarda görünüm pek iyi olmaz. `\fussy` (titiz) komutu verirseniz \LaTeX varsayılan davranışına geri döner.

2.2.2 Heceleme

\LaTeX gerekli gördüğü yerde kelimeleri hecelere böler. Hecelemede kullandığı kuralların doğru sonuç vermediği kelimelerde ona yardımcı olmak için, şu komutu kullanırız:

```
\hyphenation{kelime listesi}
```

Buradaki *kelime listesi*, hece yerleri tire (“-”) işaretiyle hecelenmiş kelimelerden oluşur ve \LaTeX bu listedeki kelimeleri o heceleme yerlerinden böler. Bu liste sadece normal bir alfabedeki harflerden ve işaretlerden oluşmalıdır. \LaTeX bunları o anda etkin olan dil için saklar. Yani, `hyphenation` komutunu sahanlık kısmında verirseniz, bunu İngilizce kelimeleri hecelemede kullanır. Fakat, `\begin{document}` komutundan sonra kullanırsanız ve `babel` paketinden başka bir dil seçip yazmaya başlamışsanız, o zaman bu hecelenmiş kelimeler o dil içinde kullanılır.

Örnek:

```
\hyphenation{FORTRAN İnceleme}
```

Bu örnekte hecelere bölme işlemi “inceleme” kelimesini gösterilen yerlerden bölmeye izin verir, fakat “FORTRAN”, “Fortran” ve “fortran” kelimelerini bölmeden dizmeye zorlar. Listede özel harfler ve sembollere izin yoktur.

Metin içinde bir kelime arasındaki (`\-`) komutu o kelimenin sadece bölünebileceği yerleri gösterir. Bu komut özellikle aksanlı harflerde kullanışlıdır, çünkü \LaTeX aksanlı kelimeleri nasıl heceleyeceğini bilmez.

```
Siz de mi slo\vak\ya\l1\-%  
lař\t1\ra\ma\d1k\la\-%  
r1\m1z\dan\s1\n1z?
```

```
Siz de mi slovakyalılařtıramadıklarımız-  
dansımız?
```

¹ \LaTeX bu (overfull box) mesajını verdiği satırı numarasıyla işaret eder, ama hangisi olduğunu bulmak bazan zor olabilir. Eğer `\documentclass` sınıfını `draft` opsiyonuyla seçerseniz, sağ taraftan fırlamış olan satırlar kalın siyah bir çizgiyle işaretlenmiş olurlar.

Birçok kelimeyi birlikte aynı satırda tutmak, şu komutla mümkündür:

```
\mbox{metin}
```

Buradaki *metin* içindeki kelimeler her durumda birleşik kalırlar.

Telefon numaram yakında değişecek.
Yeni numaram: \mbox{0116 291 2319}.

Bu komutu kullanırken
\mbox{\emph{dosya adı, dosya
uzantısı, uzunluk}} bilgilerini
yazar girmelidir.

Telefon numaram yakında değişecek. Yeni
numaram: 0116 291 2319.

Bu komutu kullanırken
dosya adı, dosya uzantısı, uzunluk
bilgilerini yazar girmelidir.

\fbox komutu da \mbox ile aynı görevi yapar, fakat metin çevresine bir kutu çizer.

2.3 Hazır Özel Kelimeler

Önceki sayfalarda L^AT_EX'in bazı kelime gruplarını dizmek için basit komutlar kullandığı gözünüzden kaçmamıştır:

Komut	Örnek	Tanımı
\today	18 Eylül 2006	O günün tarihi
\TeX	T _E X	en sevdiğiniz dizgici
\LaTeX	L ^A T _E X	Meselenin özü
\LaTeXe	L ^A T _E X 2 _ε	Mevcut sürümü

2.4 Özel Karakterler ve Semboller

2.4.1 Tırnaklar

L^AT_EX'de tırnak işaretleri daktiloda yazar gibi " şeklinde *girilmez*. Kitap diziminde özel açış ve kapanış tırnakları vardır. L^AT_EX'de açış tırnağı için iki kez ` (ters aksan) ve kapanış tırnağı için iki kez ' (dikey tırnak-apostrof) girilir. Tek tırnak için her iki cinsten bir tane kullanırsınız.

“Lütfen ‘x’ tuşuna basınız.”

“Lütfen ‘x’ tuşuna basınız.”

Biliyorum, tırnak açıp kapamak için farklı tuşlar kullanmak hoş değil, ama sonuçta istenilen görünümü veriyorlar.

2.4.2 Çizgiler ve Tireler

\LaTeX dört farklı türden çizgi ve tire çekebilir. Bunların üçünü klavyenizdeki eksi işaretini farklı sayıda kullanarak elde edebilirsiniz. Dördüncüsü aslında çizgi veya tire değil, matematikteki eksi işaretidir.

```
çelik-çomak, mavi-yeşil\\
sayfa 13--67\\
hımm---bir düşüneyim.\\
$0$, $1$ ve $-1$
```

```
çelik-çomak, mavi-yeşil
sayfa 13-67
hımm—bir düşüneyim.
0, 1 ve -1
```

Bu çizgilerin adı, sırasıyla şöyledir: ‘-’ tire, ‘-’ orta-tire, ‘—’ uzun-tire ve ‘-’ eksi işareti.

2.4.3 Tilda (~)

Tilda işareti bazı internet site adreslerinde bulunur. \LaTeX 'te tildayı dizmek için \sim kullanabilirsiniz, ama sonuç \sim tam istediğiniz gibi olmaz. Bunun yerine ikinci satırdakini deneyin:

```
http://www.itu.edu.tr/~{bk} \\
http://www.metu.edu.tr/{sim}$abc
```

```
http://www.itu.edu.tr/~bk
http://www.metu.edu.tr/~abc
```

2.4.4 Derece İşareti (°)

Derece işaretini \LaTeX 'te basmanın yolunu bir örnekte gösterelim:

```
Hava $35\,^{\circ}\mathrm{C}$.
Denize gitsek iyi olur.
```

```
Hava 35 °C. Denize gitsek iyi olur.
```

`textcomp` paketinde derece işaretini `\textcelsius` yazarak da elde edebilirsiniz.

2.4.5 Avro Para Sembolü (€)

Bugünlerde para sözkonusu olduğunda Avro sembolüne ihtiyaç duyulabiliyor. Pekçok yazı tipinde bu sembol vardır. Dokümanımızın sahanlık kısmına

textcomp paketini yükleyin:

```
\usepackage{textcomp}
```

Sonra, şu komutu verdiğinizde Avro sembolü dizilmiş olur:

```
\texteuro
```

Kullandığımız yazı tipinin kendi Avro sembolü yoksa, veya onu beğenmiyorsanız iki seçeneğiniz var: Önce `eursym` paketini deneyip, resmi Avro sembolü için şunu girersiniz:

```
\usepackage[official]{eurosym}
```

Fakat, kendi yazı tipinize uyan bir Avro sembolü istiyorsanız, opsiyon bölümündeki `official` yerine `gen` yazarsınız.

Pekçok sembol ve işaretin bulunduğu `marvosym` paketinde de Avro sembolü `\EURtm` adıyla yer alır. Fakat, buradaki sembolün italik ve kalın seçenekleri yoktur.

Tablo 2.1: Bir dolu Avro sembolü

LM+textcomp	<code>\texteuro</code>	€	€	€
eurosym	<code>\euro</code>	€	€	€
[gen]eurosym	<code>\euro</code>	€	€	€
marvosym	<code>\EURtm</code>	□01	□01	□01

2.4.6 Üç Nokta (...)

Daktiloda bir virgül veya nokta diğer harfler kadar yer kaplar. Kitap basımında ise, bu işaretlere daha az yer ayrılır ve önlerinde gelen harfe iyice yanaştırılır. Bu nedenle, üç nokta elde etmek için üç tane noktayı yan yana yazamazsınız, aralıklar doğru olmaz. Bunun yerine, üç nokta için özel bir komut kullanılır:

```
\ldots
```

Böyle böyle ... değil. Şöyle:\\
İstanbul, Ankara, İzmir, \ldots

Böyle böyle ... değil. Şöyle:
İstanbul, Ankara, İzmir, ...

2.4.7 Kalıplama

Bazı harf gruplarını dizerken harfleri birbiri ardına girmek yerine, özel semboller kullanılır.

ff fi fl ffi ... değil, ff fi fl ffi...

kalıplama (ligature) denilen bu tür yanıştırılmaları istemiyorsanız, ilgili iki harf arasına bir `\mbox{}` koyarsınız. İki kelimedenden oluşan kelimeler için bu gerekli olabilir.

`\Large Saffet bizi affet`, veya `\Saf\mbox{ }fet bizi af\mbox{ }fet`.

Saffet bizi affet, veya
Saffet bizi affet.

2.4.8 Aksanlar ve Özel İşaretler

\LaTeX birçok dünya dilinde bulunan aksanlı harfleri ve özel işaretleri destekler. Tablo 2.2 de ‘o’ harfine eklenebilecek her türlü aksan gösterilmiştir. Bu aksanlar diğer harflere de eklenebilirler.

Fakat, i veya j harflerine aksan eklemek için önce üstlerindeki noktaları kaldırılmak gerekir. `\i` ve `\j` yazıldığında noktalar kalkar, peşinden aksan eklersiniz.

`K\^atip`, `\'ecole`, `na\"i` ve, `\K\"orfez`, `\"Usk\"udar`, `A\u grı`
`Br\o d`, `Se\~norita`, `Stra\ss e`

Kâtip, école, naïve,
Körfez, Üsküdar, Ağrı
Brød, Señorita, Straße

Tablo 2.2: Aksanlar ve Özel İşaretler.

ò	<code>\'o</code>	ó	<code>\'o</code>	ô	<code>\^o</code>	õ	<code>\~o</code>
ō	<code>\=o</code>	ó	<code>\.o</code>	ö	<code>\"o</code>	ç	<code>\c c</code>
ö	<code>\u o</code>	ö	<code>\v o</code>	ő	<code>\H o</code>	q	<code>\c o</code>
ø	<code>\d o</code>	o	<code>\b o</code>	oo	<code>\t oo</code>		
œ	<code>\oe</code>	Œ	<code>\OE</code>	æ	<code>\ae</code>	Æ	<code>\AE</code>
å	<code>\aa</code>	Å	<code>\AA</code>				
ø	<code>\o</code>	Ø	<code>\O</code>	ı	<code>\l</code>	Ł	<code>\L</code>
ı	<code>\i</code>	ı	<code>\j</code>	ı	<code>!'</code>	ı	<code>?'</code>

2.5 Uluslararası Dil Desteği

İngilizceden farklı bir dilde yazıyorsanız, \LaTeX 'in üç ayrı yerde ayarlarının değiştirilmesi gerekir:

1. \LaTeX 'in otomatik olarak ürettiği (İçindekiler, Şekiller Listesi, . . .) başlıkların da yeni dile uyarlanması gerekir. Pekçok dil için bu değişiklikler Johannes Braams'ın yazdığı `babel` paketi ile kendiliğinden oluşurlar.
2. \LaTeX 'in yeni dildeki heceleme kurallarını bilmesi gerekir. \LaTeX 'in heceleme kurallarını değiştirmek biraz karmaşık bir konudur, format dosyalarının farklı heceleme tabanına göre yeniden inşa edilmesi gerekir. Bu konuda *Local Guide* [5](Yerel Rehber) size daha fazla bilgi verebilir.
3. Varsa, o dile has dizgi kuralları. Örneğin, Fransızca'da (:) işaretinden önce ve sonra boşluk bırakılır.

Bilgisayarınızda \LaTeX doğru kurulmuşsa, `babel` paketini etkin hale getirmek için, `\documentclass` komutundan sonra şu komutu eklersiniz:

```
\usepackage[dil]{babel}
```

Sizin \LaTeX sisteminizde kurulmuş olan dillerin listesi her derleme sırasında ekrana yansıtılmaktadır. Bu dillerden birini kullanıyorsanız, Babel hemen o dili etkin hale getirir. Fakat, kullanacağınız dil bu listede yoksa, `babel` hala değiştirmek ister, ama sonuçlar hiçbir zaman istenilen düzeyde olmaz.

`babel` paketinde bazı diller için, özel harfleri girmeyi kolaylaştıran yeni komutlar da vardır. Örneğin, Almanca ve Türkçede pekçok umlaut (iki nokta) aksanlı harfler vardır. `babel` ile ö harfini yazmak için `\"o` yerine `"o` girebilirsiniz.

Babel paketini birkaç dilde birden yükleyebilirsiniz:

```
\usepackage[dilA,dilB]{babel}
```

Bu komutla listedeki dillerin en sonuncusu (burada, `dilB`) etkin hale gelir. Etkin dili değiştirmek için şu komutu kullanırsınız:

```
\selectlanguage{dilA}
```

Modern bilgisayar sistemlerinde farklı dillerin harfleri doğrudan klavyeden girilebilmektedir. Değişik dil gruplarındaki bu değişik alfabeleri, değişik

işletim sistemlerinde destekleyebilmek için \LaTeX `inputenc` paketini kullanır:

```
\usepackage[kodlama]{inputenc}
```

Bu paketi kullanırken, diğer kişilerin kendi bilgisayarlarında sizin girdi dosyalarınızı doğru göremeyeceklerini hesaba katmalısınız, çünkü onların kullandığı kodlama farklı olabilir. Örneğin, Almanca `ä` harfi OS/2 işletim sisteminde 132 kodlu, ISO-Latin 1 kullanan Unix sistemlerinde 228 olarak kodlanmış olup, Rusçanın `cp1251` kodlamasında ise hiç bulunmaz. O halde, bu seçeneği dikkatli kullanmalısınız. Aşağıdaki tabloda ençok kullanılan bazı kodlama sistemleri gösterilmiştir.²

İşletim sistemi	kodlama	
	batı Latin	Kiril
Mac	<code>applemac</code>	<code>macukr</code>
Unix	<code>latin1</code>	<code>koi8-ru</code>
Windows	<code>ansinew</code>	<code>cp1251</code>
DOS, OS/2	<code>cp850</code>	<code>cp866nav</code>

Bir dokümanda birkaç dili birden kullanıyorsanız ve girdi kodlamaları birbiriyle çelişiyorsa, `ucs` paketini yükleyip unikod kodlamasına geçebilirsiniz:

```
\usepackage{ucs}
\usepackage[utf8x]{inputenc}
```

Unikod herbir harfin bir ila dört bit arasında kodlanabildiği çok-bitli `utf8x` kodlama sisteminde \LaTeX girdi dosyalarınızı yazmaya olanak verir.

Yazıtipi kodlaması apayrı bir sorundur. Bu, herhangi bir \TeX yazıtipinde herbir harfin nerede saklandığını belirler. Çok sayıda girdi kodlaması bir tek yazıtipi kodlamasına aktarılarak gerekli yazıtipi sayısı azaltılmış olur. Yazıtipi kodlamaları `fontenc` paketiyle yönetilir:

```
\usepackage[kodlama]{fontenc}
```

Burada *kodlama* yazıtipi kodlaması olup birkaç tanesi birlikte yüklenebilir.

\LaTeX 'in varsayılan yazıtipi kodlaması `OT1` olup orijinal \TeX yazıtipi olan Computer Modern (CM) yazıtipinin kodudur. 7-bitlik ASCII sembol setinde sadece 128 sembol saklayabilir. Aksanlı harfler gerektiğinde \TeX normal bir harfi uygun bir aksanla birleştirip üretir. Bu yöntem görünüşte mükemmel olsa da, aksanlı harflerin hecelenmesine izin vermez. Ayrıca, bazı Latince

²Latince ve Rusça tabanlı sistemlerde girdi kodlaması konusunda daha fazla bilgi için, sırasıyla `inputenc.dtx` ve `cyinpenc.dtx` dokümanlarına bakabilirsiniz. Paket doküman-tasyonu nasıl yazabileceğinizi Kısım 4.6 de bulabilirsiniz.

harfler aksanla birleşerek üretilemeyebilir, Yunanca veya Kiril alfabelerinde sorun daha da büyük olur.

Tüm bu sorunları çözmek üzere, 8-bitlik CM tipi yazıtipi takımları üretilmiştir. Bunlardan T1 kodlamasına sahip *Genişletilmiş Cork* (EC) yazıtipi pekçok Avrupa dilindeki Latin harflerini tümüyle kapsar. LH yazıtipi takımı Rusça doküman hazırlamak için gerekli Kiril harflerini bulundurur. Kiril alfabesinde aşırı sayıda aksanlı harf bulunduğundan, dört ayrı kodlama—T2A, T2B, T2C, ve X2— bulunur.³ LGR kodlamasına sahip olan CB paketi ise Yunanca metin dizmekte kullanılır.,

Bu yazıtiplerini kullanmak İngilizce dışındaki dillerde hecelemeyle kolaylaştırır. Yeni CM-tipi yazıtipleri kullanmanın diğer bir avantajı, CM ailesinin yazıtiplerini her punto ve kalınlıkta verebilmesidir.

2.5.1 Türkçe Desteği

Önce, \LaTeX 'in otomatik olarak ürettiği (İçindekiler, Şekiller Listesi, ...) gibi başlıkların Türkçeleşmesi için, `\documentclass` komutundan sonra şu komutu eklersiniz:

```
\usepackage[turkish]{babel}
```

Sonra, Türkçedeki aksanlı ve kuyruklu harfleri dizabilmek için girdi kodlamasını şöyle seçersiniz:

```
\usepackage[latin5]{inputenc}
```

Son olarak, hecelemeyle doğru yapabilmek için yazıtipi kodlamasını seçersiniz:

```
\usepackage[T1]{fontenc}
```

Artık hem klavyenizden Türkçe harfleri girebilir, hem de Türkçe dokümanlar dizabilirsiniz. Tablo 2.3 de özetlenen bu üç komut Türkçe doküman yazmak için yeterlidir.

2.5.2 Almanca Desteği

\LaTeX ile Almanca doküman yazmanın birkaç püf noktasını belirtmekle yetinelim. Almanca dil desteğini şu komutla yüklersiniz:

```
\usepackage[german]{babel}
```

Böylece Almanca hecelemeyle doğru yapılır, otomatik üretilen başlıklar

³Bu kodlamaların herbirinin hangi dilleri desteklediğini [11] dokümanında bulabilirsiniz.

Tablo 2.3: Türkçe Doküman Yazmak için Gerekli Ön Komutlar.

```

\usepackage[turkish]{babel}
\usepackage[latin5]{inputenc}
\usepackage[T1]{fontenc}

```

Almanca dizilir. Örneğin, bölüm başlığında “Chapter” yerine “Kapitel” gelir. Yeni bir takım komutlar Almanca harfleri girmenizi kolaylaştırdığı için, ayrıca inputenc paketini yüklemenize gerek kalmaz. Tablo 2.4 de aksanlı harflerin nasıl girileceği gösterilmiştir. Elbette bu işi inputenc paketiyle daha kolay yapabilirsiniz, ama o zaman girdi dosyanız ayrı bir kodlamaya mahkum kalır.

Tablo 2.4: Özel Almanca Harfler.

"a	ä	"s	ß
"‘	”	"’	“
"< veya \flqq	«	"> veya \frqq	»
\flq	‹	\frq	›
\dq	"		

2.6 Kelime Arası Boşluklar

L^AT_EX sağ taraftaki marjı hep aynı genişlikte tutabilmek için, kelimeler arasına değişik uzunlukta boşluklar koyar. Okumayı kolaylaştırmak amacıyla cümle sonlarında biraz daha boşluk bırakır. L^AT_EX her cümlenin nokta, soru işareti veya ünlem işaretiyle bittiğini varsayar. Eğer bir büyük harften sonra nokta geliyorsa, bunu cümle sonu saymaz, çünkü sadece kısaltmalarda büyük harflerden sonra nokta gelir.

Bu varsayımların istisnaları varsa, yazar bunu belirtmelidir. Örneğin, bir boşluğun önüne geribölü işareti konmuşsa, genişlemeyen bir boşluk açılır. Bir tilda ‘~’ işareti hem genişlemeyen boşluk bırakır, hem de orada satır kesilmesini önler. Bir noktanın önüne \@ işareti konursa, büyük harften sonra gelse bile, cümle sonu olduğunu belirtir.

```
Mrs.~Smith pazara gidiyor\\
bkz.~Şek.~5\\
İşte FORTRAN\@. Ne dersiniz?
```

```
Mrs. Smith pazara gidiyor
bkz. Şek. 5
İşte FORTRAN. Ne dersiniz?
```

Noktadan sonraki ekstra boşluğu istemiyorsanız,

```
\frenchspacing
```

komutuyla \LaTeX 'in noktadan sonra fazladan boşluk *koymamasını* sağlarsınız. Bu usül İngilizce dışındaki bazı dillerde yaygındır. `\frenchspacing` kullandıysanız, artık noktadan önce `\@` koymanıza gerek yoktur.

2.7 Başlıklar, Bölümler, ve Kısımlar

Bir okuyucunun sizin dokümanınız içinde yolunu bulabilmesi için, onu bölümler, kısımlar ve altkısımlar olarak ayırmalısınız. \LaTeX bu işi özel komutlarla yapar, bu komutlarda örneğin bölüm adı argüman olarak verilir.

`article` (makale) sınıfı bir dokümanda kullanabileceğiniz başlık komutları şunlardır:

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

Kitap (`book`) veya rapor (`report`) sınıfı dokümanlarda daha üst düzeyde 'bölüm' (`chapter`) komutu vardır:

```
\chapter{...}
```

`article` (makale) sınıfında bölümler olmadığından, makaleler bir kitabın bölümleri olarak eklenebilirler. \LaTeX bölüm ve kısım başlıklarının numaralanmasını ve harf punto seçimini otomatik olarak kendisi yapar.

Doküman bölen komutlardan ikisinin ayrı birer özelliği vardır:

- Çok hacimli dokümanları bölmek için `\part` (kesim) komutu kullanılır:

```
\part{...}
```

Kesim komutu bölüm ve kısım numaralarını bozmaz.

- The `\appendix` (ekler) komutu içine argüman almaz. Sadece bölüm numaralarını harflere çevirir. (article sınıfında kısım numaralarını harfe çevirir.)

L^AT_EX dokümandaki başlık adları ve sayfa numaralarını kullanarak İçindekiler tablosunu şu komutla oluşturur:

```
\tableofcontents
```

Bu komutun verildiği yerde İçindekiler tablosu oluşur. Fakat, bu bilgiler dokümanın bir önceki derlemesine ait olduğundan, doğru olmayabilirler ve dokümanın iki (bazan üç) kez derlenmesi gerekir. Tekrar derlenmesini istediğinde L^AT_EX bunu size bildirir.

Yukardaki bölme komutlarının herbiri için “yıldızlı” bir kullanım şekli daha vardır. Komut adından sonra bir * işareti konulursa, bölüm ve kısım adları numaralanmaz ve İçindekiler tablosunda yer almazlar. Örneğin, `\section{Giriş}` yerine `\section*{Giriş}` yazılır.

Normalde İçindekiler tablosunda başlık adları yazıldığı şekilde yer alırlar. Bazan, uzun başlık adlarının tabloda daha kısa yer alması istenebilir. Bu durumda, başlığın bulunduğu yere bir komut eklenip, opsiyonel kısmına İçindekiler tablosunda yer alacağı şekli yazılır:

```
\chapter[Fillerin Anatomisi]{Fillerin değişik
kıtalardaki anatomilerinin karşılaştırılması}
```

Bir dokümanın ana başlığını oluşturmak üzere, önce başlık adı, yazarı, tarih, ... kolay anlaşılır komutlarla madde madde girilir:

```
\title{...}, \author{...} ve istenirse \date{...}
```

Burada `\title` başlık adı, `\author` yazar adı ve `\date` günün tarihi olur. Yazar adında çok sayıda isim varsa, bunlar `\and` komutuyla ayrılarak eklenebilirler.

Hepsi tamam olunca, başlığı oluşturan komut girilir:

```
\maketitle
```

Tüm bu komutların kullanıldığı bir örnek Sayfa 8 deki Şekil 1.2 de gösterilmiştir.

Yukardaki bölme komutları dışında, `book` sınıfının bölümlenmesi için üç tane daha komut vardır. Bu komutlar bölüm başlıklarını ve sayfa numaralarını bir kitapta görmeye alıştığımız şekilde dizebilmenizi sağlarlar:

`\frontmatter` (ön taraf) komutu yazılan metnin başladığı (`\begin{document}`) komutundan hemen sonra verilmelidir. Bu komut, baş taraftaki İçindekiler, Önsöz gibi kısımların sayfa numaralandırmasını Roma rakamıyla yapar.

`\mainmatter` (ana metin) komutu kitabın ilk bölüm başlığından hemen sonra gelmelidir. Buradan itibaren sayfa numaralandırmasını yeniden başlatıp rakamlara geçer.

`\appendix` (ekler) komutu kitabınızın eklerindeki bölümleri harflerle numaralandırır (Ek A, Ek B, ...).

`\backmatter` (arka taraf) komutu kitabınızda herşeyin, Kaynakça ve Dizin bittikten sonra kullanılır. Fakat, bilinen doküman sınıflarında görünürde hiçbir etkisi yoktur.

2.8 İç Atıflar

Kitap, rapor ve makalelerde daima şekillere, tablolara veya denklemlere iç atıflar bulunur. \LaTeX bu tür atıfları şu komutlarla düzenler:

```
\label{işaret}, \ref{işaret} ve \pageref{işaret}
```

Burada *işaret* kullanıcının seçtiği bir kelime veya kısaltmadır. Bir şekil veya denkleme `\label` komutu eklendiğinde \LaTeX onun numarasını saklar. Daha sonraki derlemede, `\ref` komutunun geçtiği yere bu şekil veya denklemin numarasını koyar. `\pageref` komutunda ise, `\label` komutunun bulunduğu sayfa numarasını koyar. Başlıklarda olduğu gibi, kullanılan numaralar bir önceki derlemenin sonuçlarıdır, doğru olması için en az iki kere derlenmelidir.

Bu kısma atıf yapmak için şöyle yazın: `\label{sec:this}`
 ‘‘bakınız Sayfa~`\pageref{sec:this}`,
 Kısım~`\ref{sec:this}`.’’

Bu kısma atıf yapmak için şöyle yazın:
 ‘‘bakınız Sayfa 31, Kısım 2.8.’’

2.9 Dipnotlar

Bir sayfanın dipnot bölgesine bir dipnot yazılmak istendiğinde,

```
\footnote{dipnot metni}
```

komutu kullanılır. Dipnotlar mutlaka açıklama yaptıkları kelimenin⁴ veya cümlelerin sonuna konulmalıdır. Cümle veya cümle parçasına konulan dipnot nokta veya virgülden sonra yer almalıdır.⁵

⁴Mesela böyle.

⁵Dipnotlar okuyucunun dikkatini dağıtır, fazla kullanmayın. Eğer herkes dipnotları okuyacaksa, ana metin içinde neden yer almasın ki?⁶

⁶Dipnot içinde dipnot, bunun sonu gelir mi?

Dipnot`\footnote{Bu bir dipnottur.}` kullanmak `\LaTeX{}`'de kolaydır.

Dipnot^a kullanmak `\LaTeX{}`'de kolaydır.

^aBu bir dipnottur.

2.10 Vurgulanmış Kelimeler

Daktiloda yazarken önemli kelimelerin altı çizilerek vurgulanır. `\LaTeX{}`'te bunu şu komutla yaparsınız:

```
\underline{metin}
```

Fakat, matbaada basılı kitaplarda vurgulanmak istenen kelimeler *italik* yazıtipiyle dizilirler. `\LaTeX{}`'te bunu,

```
\emph{metin}
```

komutuyla yaparsınız. Fakat, bu komutu hangi metne uyguladığınız önemlidir. Şu örneğe bakın:

```
\emph{Vurgulanmış bir metinde
tekrar vurgulama yaparsanız,
\LaTeX{ } onu \emph{düz}
yazıtipine çevirir.}
```

Vurgulanmış bir metinde tekrar vurgulama yaparsanız, `\LaTeX{}` onu düz yazıtipine çevirir.

Keza, `\LaTeX{}`'in bir metni *vurgulaması* ile, başka bir *yazıtipinde* dizmesi arasında fark vardır:

```
\textit{İtalik dizilmiş bir
metinde \emph{vurgu}
yapmak isterseniz,
bunu \textsf{serifsiz
yazıtipinde \emph{vurgu}}, veya
\texttt{daktilo yazıtipinde
\emph{vurgu}} olarak
yapabilirsiniz.}
```

İtalik dizilmiş bir metinde vurgu yapmak isterseniz, bunu serifsiz yazıtipinde vurgu, veya daktilo yazıtipinde vurgu olarak yapabilirsiniz.

2.11 Ortamlar

```
\begin{ortam} metin \end{ortam}
```

Burada *ortam* kullanılan ortamın adıdır. Ortamlar birbiri içinde açılabilirler, ama doğru sırada yerelmalındırlar, yani en içtekenden itibaren kapatılmalındırlar:

```

\begin{aaa}
...
  \begin{bbb}
    ...
  \end{bbb}
...
\end{aaa}

```

Şimdi, en önemli ortamların açıklamasına geçelim.

2.11.1 Sıralandırma, Numaralandırma, ve Maddeleme

Bir listeyi madde madde girmek için değişik *ortamlar* kullanılır. Basit listeler için `enumerate` ortamı uygun olur. Numaralı listeler için `enumerate` ortamı kullanılır. Tanımlayıcı maddelerle başlayan listeler için de `description` ortamı uygun olur.

```

\flushleft
\begin{enumerate}
\item Ortamları istediğiniz gibi
birlikte kullanabilirsiniz:
\begin{itemize}
\item Bu biraz basit oldu.
\item[-] İsterseniz eksi işaretiyle.
\end{itemize}
\item O zaman unutmayın:
\begin{description}
\item[Kuşlar] havada uçar.
\item[Balıklar] denizde yüzer.
\end{description}
\end{enumerate}

```

1. Ortamları istediğiniz gibi birlikte kullanabilirsiniz:
 - Bu biraz basit oldu.
 - İsterseniz eksi işaretiyle.
2. O zaman unutmayın:

Kuşlar havada uçar.

Balıklar denizde yüzer.

Örnekte görüldüğü gibi, `itemize` ortamında her madde bir yuvarlak (●) ile başlar, istenirse bu madde işareti değiştirilebilir. `enumerate` ortamında numaralar otomatik olarak artar. `description` ortamında da köşeli parantez içine alınan ilk kelime kalın dizilir.

2.11.2 Sola Yaslama, Sağa Yaslama, ve Ortalama

`flushleft` ve `flushright` ortamları, sırasıyla sola ve sağa yaslanmış paragraflar dizer. `center` ortamı ise ortalanmış paragraflar içindir. Eğer `\\` komutuyla satırları siz kesmezseniz, `LaTeX` satırları kendi bildiği yerden keser.

```

\begin{flushleft}
Bu yazı\\ sola yaslanmıştır.
\LaTeX{} her satırı eşit
uzunlukta dizmeye çalışmaz.
\end{flushleft}

```

Bu yazı
sola yaslanmıştır. `LaTeX` her satırı eşit
uzunlukta dizmeye çalışmaz.

```
\begin{flushright}
Bu yazı sağa \\yaslanmıştır.
\LaTeX{} her satırı eşit
uzunlukta dizmeye çalışmaz.
\end{flushright}
```

Bu yazı sağa
yaslanmıştır. \LaTeX her satırı eşit
uzunlukta dizmeye çalışmaz.

```
\begin{center}
Dünya'nın\\ortasında ne var?
\end{center}
```

Dünya'nın
ortasında ne var?

2.11.3 Alıntı Yapmak ve Şiir Dizmek

Başka bir yazardan alıntı yapmak veya önemli cümleler dizmek için `quote` ortamı kullanılır. Bu ortamda metin daha dar bir alana dizilir:

```
Matbaacılıkta satır uzunluğu
için pratik kural şudur:
\begin{quote}
Bir satırda, ortalama olarak
66 harften fazla olmamalıdır.
\end{quote}
Bu yüzden \LaTeX{} dokümanlarının
sayfalarında marjlar geniş
olur ve yine bu yüzden
gazeteler çok sütunlu
basılırlar.
```

Matbaacılıkta satır uzunluğu için pratik kural şudur:

Bir satırda, ortalama olarak
66 harften fazla olmamalıdır.

Bu yüzden \LaTeX dokümanlarının sayfalarında marjlar geniş olur ve yine bu yüzden gazeteler çok sütunlu basılırlar.

Buna benzer iki ortam daha vardır: `quotation` ve `verse` (şiir) ortamları. `quotation` ortamı daha uzun, birçok paragraftan oluşan alıntılar içindir ve her paragrafın ilk satırı içerden başlar. `verse` ortamı, satır kesmenin önemli olduğu şiirleri dizmek için kullanılır. Şiir satırları `\\` komutuyla sonlandırılır ve her kıta sonunda boş bir satır bırakılır.

```
Türkçenin büyük bir
ustasından dizeler:
\begin{verse}
Ben giderim adım kalır,\\
Dostlar beni hatırlasın.\\
Düğün olur, bayram gelir,\\
Dostlar beni hatırlasın.\\[1ex]

Gün ikindi akşam olur,\\
Gör ki başa neler gelir,\\
Veysel gider, adı kalır\\
Dostlar beni hatırlasın.\\
\end{verse}
```

Türkçenin büyük bir ustasından dizeler:

Ben giderim adım kalır,
Dostlar beni hatırlasın.
Düğün olur, bayram gelir,
Dostlar beni hatırlasın.

Gün ikindi akşam olur,
Gör ki başa neler gelir,
Veysel gider, adı kalır
Dostlar beni hatırlasın.

2.11.4 Özet (Abstrak)

Bilimsel yayınların başında, okuyucuya içerik hakkında bir fikir vermek için özet (abstrak) koymak adettir. \LaTeX bu amaçla kullanmak üzere `abstract` ortamı sunar. `abstract` ortamı genellikle makale sınıfı dokümanlarda kullanılır.

```
\begin{abstract}
Özetin de özeti \ldots
\end{abstract}
```

```
Özetin de özeti ...
```

2.11.5 Yazıldığı Gibi (Verbatim) Basmak

Bir metni tıpatıp daktiloda yazıldığı gibi, yani satır sonları ve boşluk miktarları ne kadarsa, içinde \LaTeX komutları varsa onları da işletmeden, olduğu gibi dizmek için `\begin{verbatim}` ve `\end{verbatim}` komutları arasına koyarsınız.

```
\begin{verbatim}
10 PRINT "MERHABA DÜNYA ";
20 GOTO 10
\end{verbatim}
```

```
10 PRINT "MERHABA DÜNYA ";
20 GOTO 10
```

Aynı davranışı sadece bir paragraf içindeki birkaç kelimeye uygulamak isterseniz,

```
\verb|metin|
```

komutu kullanırsınız. Burada `| |` sınırlar için sadece bir örnektir, harfler dışında, örneğin `*` veya `+` veya boşluk da olabilir. Bu kitaptaki pekçok örnek `\verbatim` ortamında dizilmişlerdir.

Belki `\verb|\ldots|` komutu `\ldots`

```
Belki \ldots komutu ...
```

`\verbatim` ortamını yıldızlı (`*`) kullandığımızda, boşluklar için özel işaret koyar:

```
\begin{verbatim*}
verbatim ortamının
yıldızlı          sürümü
metin içindeki
boşlukları       vurgular
\end{verbatim*}
```

```
verbatim_ortamının
yıldızlı_üüüüüüüüüüüü_sürümü
metin_üü_içindeki
boşlukları_üüüü_vurgular
```

Paragraf içindeki `\verb` komutunu da yıldızlı kullanılabilir:

```
\verb*|ište böyle :-) |
```

```
ište_böyle_:-)_
```

`\verb` ortamı ve `\verb` komutu başka bir komutun parametreleri içinde kullanılamazlar.

2.11.6 Tablolar

Yatay ve dikey çizgilerle ayrılmış tablolar ve cetveller `tabular` ortamında dizilirler. Sütun genişliklerini \LaTeX kendisi ayarlar.

```
\begin{tabular}[konum]{özellikler}
```

Bu komutun içindeki *özellikler* argümanı tablonun formatını belirler. Sola yanaşık bir sütun için `l`, sağa yanaşık bir sütun için `r`, ortalanmış bir sütun için `c` parametreleri kullanılır. `|` işareti dikey bir çizgi çizdirir.

Bir sütun içindeki metin fazla uzunsa \LaTeX onu bölüp alt satıra yazmaz. `p{genişlik}` komutu kullanırsanız, verilen genişlikte bir sütun açar ve normal bir paragraf gibi yazıyı böler.

Keyfi olan *konum* argümanı tablonun, çevresindeki metne göre dikey konumunu belirler. `t`, `b` ve `c` harfleri, tabloyu sırasıyla üst, alt ve orta konuma yerleştirir.

`tabular` ortamında `&` işareti bir sonraki sütuna yazdırır, `\\` komutu yeni bir satır başlatır ve `\hline` komutu yatay bir çizgi çeker. Sadece birkaç sütundaki maddelerin altını çizdirmek isterseniz `\cline{i-j}` komutu kullanırsınız. Burada *i* ve *j*, çizginin başlayıp bittiği sütun numaralarıdır.

```
\begin{tabular}{|r|l|}
\hline
7C0 & hegzadesimal \\
3700 & oktal \\ \cline{2-2}
11111000000 & ikili \\
\hline \hline
1984 & ondalık \\
\hline
\end{tabular}
```

7C0	hegzadesimal
3700	oktal
11111000000	ikili
1984	ondalık

```
\begin{tabular}{|p{3.7cm}|c|}
\hline
Boxy'nin yerine
hoşgeldiniz. & 3 ytl \\
\hline
\end{tabular}
```

Boxy'nin yerine hoşgeldiniz.	3 ytl
------------------------------	-------

Bazı tablo girdilerinin iki veya daha çok sütunu kapsaması isteniyorsa `\multicolumn` komutu kullanılır:

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Ene	
Mene	Muh!

Sütun ayracı olarak kullanılan `@{...}` komutu çok marifetlidir, sütunların başında ve sonundaki boşluğu yokeder, yerine çengelli parantez içinde ne varsa onu koyar. Bu özellik, sütun baş ve sonlarındaki boşluğu yoketmekte kullanılabilir. Aradaki farkı şu örnekte görebilirsiniz:

```
\begin{tabular}{l}
\hline
başta ve sonda boşluklar \\
\hline
\end{tabular}
```

<u>başta ve sonda boşluklar</u>

```
\begin{tabular}{@{} l @{}}
\hline
hiç boşluk yok \\
\hline
\end{tabular}
```

<u>hiç boşluk yok</u>

`@{...}` ayraçının diğer bir kullanımı, ondalık kesirleri hizaya getirmesidir. `tabular` ortamında sayısal girdileri ondalık (.) noktasına göre hizaya getirecek bir komut yoktur. Bunun yerine şöyle bir “aldatmaca” yapabiliriz: ondalık kesri iki sütunmuş gibi gireriz, tamsayı kısmını sağa yaslar, kesirli kısmını sola yaslar ve iki sütun ayracı olarak da `@{.}` kullanırız. Böylece iki taraf birbirine yaslanmış, sütun arası boşluk yok edilmiş ve yerine ondalık noktası (.) konulmuş olur. Ondalık sayının iki tarafını sütun ayracı (&) ile ayırmayı unutmayın! Bu “sütuna” bir başlık koymak gerekirse `\multicolumn` komutu kullanılabilir.

```
\begin{tabular}{c r @{.} l}
Zaman & \\
\multicolumn{2}{c}{Yükseklik} \\
\hline
2 & 3 & 865 \\
4 & 17 & 48 \\
6 & 180 & 9 \\
\end{tabular}
```

Zaman	Yükseklik
2	3.865
4	17.48
6	180.9

`tabular` ortamında girilen metinler daima aynı sayfada kalacak şekilde dizilirler. Daha uzun tablolar dizmek istiyorsanız `longtable` paketini kullanabilirsiniz.

2.12 Yüzer-Gezer Nesnelere

Her doküman içinde pekçok şekil ve tablo bulunur. Bu nesnelere dizeken özel bir önem gerekir, çünkü bunlar iki sayfaya bölünemezler. İlk çözüm, her şekil veya tabloyu ayrı bir sayfaya basmak olabilirdi. Fakat, bu yaklaşımda pekçok sayfa yarı boş kalır ve görüntü kötü olur.

Bu sorunu çözenin doğru yolu, şekil veya tabloları ‘yüzer-gezer’ hale getirmektir. Böylece, bir sayfadaki yerine sığmayan şekli ileri bir sayfaya erteleyip onun bırakacağı boşluğu metinle doldururuz. \LaTeX yüzer-gezer nesnelere için, biri şekiller, diğeri tablolar olmak üzere, iki ortam sunar. Bu iki ortamı doğru kullanabilmek için \LaTeX 'in yüzer-gezerleri nasıl işlediğini bilmek gerekir. Bu bilinmezse yüzer-gezerler başağrısı olmaya devam ederler, çünkü \LaTeX onları sizin istediğiniz yere bir türlü koyamaz.

Önce \LaTeX 'in yüzer-gezerler için sunduğu komutlara bakalım: **figure** (şekil) veya **table** (tablo) ortamında dizilen herşey yüzer-gezer nesne olarak algılanır.

```
\begin{figure}[konum] veya \begin{table}[...]
```

Her iki ortamın opsiyonel *konum* parametresi \LaTeX 'e yüzer-gezeri nereye taşıyabileceğini söyler. Tablo 2.5 de konum parametresi için seçenekler gösterilmiştir.

Bir tablonun şöyle bir komutla başlatıldığını düşünelim:

```
\begin{table}[!hbp]
```

Buradaki konum parametresi `[!hbp]` \LaTeX 'e tabloyu bulunduğu yere (**h**), veya yerleştireceği sayfanın dibine (**b**), veya yüzer-gezerler için özel bir sayfaya (**p**), veya kötü görünse de bunlardan birine (!) yerleştirmesini söyler. Yerleştirme parametresi verilmemişse, `[tbp]` seçeneği olduğu varsayılır.

Tablo 2.5: Yüzer-gezer Yerleştirme Seçenekleri.

Seçenek	Açıklama
h	<i>buraya</i> , metinde yazılı yere. Bu genellikle küçük yüzer-gezerler için kullanılır.
t	<i>üste</i> , sayfanın üst tarafına.
b	<i>alta</i> , sayfanın alt tarafına.
p	<i>sayfaya</i> , sadece yüzer-gezerlerin bulunduğu özel bir sayfaya.
!	(mutlaka diğere seçeneklerden birine), iç kısıtlamaların ^a engel olmasına aldırmadan.

^aÖrneğin, bir sayfaya girebilecek maksimum yüzer-gezer sayısı sınırlanmış olabilir.

\LaTeX her yüzer-gezeri yazarın verdiği seçeneklere göre yerleştirmek için elinden geleni yapar. Yüzer-gezer bulunduğu sayfaya yerleşmiyorsa, *figures* veya *tables* kuyruklarından birine alınır. (Bu kuyruksa ilk giren ilk çıkar.) Yeni bir sayfa açıldığında \LaTeX önce kuyruksaaki yüzer-gezerlerle dolu özel bir ‘yüzer-gezer sayfası’ yapıp yapamayacağına bakar. Bu mümkün değilse, her kuyruksaaki ilk yüzer-gezer alınıp onun konum parametresine göre yerleştirmeye çalışır (‘h’ seçeneği artık geçersiz olduğundan dikkate alınmaz). Diğer yeni yüzer-gezerler de kuyruksa alınıp işlem görürler. \LaTeX her türden yüzer-gezerin orijinal sırasını asla bozamaz. Bu yüzden, doğru yerine yerleşmeyen bir yüzer-gezer, diğerlerini de çok daha ötelere iter. Bu nedenle:

Eğer \LaTeX yüzer-gezerleri istediğiniz yere koyamıyorsa bunun sebebi, iki yüzer-gezer kuyruksaından birindeki trafik tıkanıklığının diğerini de etkiliyor olmasıdır.

Konum parametresi olarak \LaTeX ’e tek bir seçenek verilebilir, fakat bu sorun yaratır. Eğer yüzer-gezer oraya sığamazsa tıkanıp kalır ve sonraki yüzer-gezerlerin önünü tıkar. Özellikle [h] seçeneği asla yalnız başına verilmez; o kadar sorun yaratır ki \LaTeX ’in yeni sürümlerinde bu seçenek otomatik olarak [ht] olarak işlem görür.

Olabilecek sorunları açıkladıktan sonra, şimdi `figure` ve `table` ortamlarının kullanılışı hakkında bilgi verelim.

Yüzer-gezerlere açıklayıcı bir alt yazı koymak isterseniz şu komutu kullanırsınız:

```
\caption{altyazı}
```

Bu komut, resimler için “Şekil”, tablolar için “Tablo” yazıp önüne bir numara koyar ve *altyazı* olarak girdiğiniz metni dizer.

Dokümanınızın başına Şekiller ve Tablolar listesi koymak için, şu iki komutu kullanırsınız:

```
\listoffigures ve \listoftables
```

Bu komutlar da tıpkı İçindekiler tablosunun `\tableofcontents` komutu gibi çalışır ve sırasıyla, Şekiller ve Tablolar listesi çıkarır. Bu listelerde şekil veya tablonun alt yazısı olduğu gibi zikredilir; bu yüzden, eğer uzun alt yazılar kullanmışsanız, daha kısa bir sürümünü `\caption` komutundan sonraki köşeli parantez içinde verebilirsiniz:

```
\caption[Kısa]{Uzzzzzzuuuuunnnnn}
```

Yüzer-gezerinize atıfta bulunmak isterseniz `\label` komutu eklersiniz. Keza, yüzer-gezer içinde başka bir şeye atıfta bulunmak için `\ref` komutu kullanabilirsiniz.

Aiağıdaki örnekte bir kare çizdirilip doküman içine eklenmektedir. (Şekil çizdirme konusu daha sonra işlenecektir.) Bunu, dokümanınızın son halinde ekleyeceğiniz bir şekile şimdiden yer ayırmak istediğinizde kullanabilirsiniz.

```
Figure~\ref{white} Bu bir modern tablodur.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Beş çarpı beş santimetre.\label{white}}
\end{figure}
```

Bu örneği \LaTeX önce *buraya* (h) yerleştirmek için *gerçekten* (!) uğraşır (tabii, şekil kuyruğunda başka birşey yoksa). Bu mümkün olmazsa, aynı sayfanın *altına* (b) yerleştirmeyi dener. Bu da olmuyorsa, bu şekli tablolar kuyruğunda birikmiş tablolarla birlikte özel bir yüzer-gezer sayfasına dizmeye çalışır. Fakat, özel sayfa için birikmiş yeterince malzeme yoksa, yeni bir sayfa açar ve şekil komutu yeni verilmiş gibi işlem yapar.

Bazı durumlarda şu iki komutu kullanmak zorunlu olabilir:

```
\clearpage ve hatta \cleardoublepage
```

Bu komutlar \LaTeX 'e yeni bir sayfa açıp kuyruklardaki yüzer-gezerleri hemen yerleştirmesini emreder. `\cleardoublepage` komutu, sağ sayfadan başlayabilmek için, gerekirse iki sayfa açmasını söyler.

Bu kitapta POSTSCRIPT formatında çizilmiş şekilleri \LaTeX dokümanınıza nasıl ekleyebileceğinizi daha sonra anlatacağız.

2.13 Kırılğan Komutları Korumak

`\caption` veya `\section` gibi komutlara argüman olarak yazılan metinler doküman içinde birden fazla yerde yeralabilirler (örneğin, hem metinde hem de İçindekiler tablosunda). Bazı komutlar diğer bir komutun argümanı içinde (örneğin, `\section` komutu içinde) yer aldıklarında çökebilirler ve dokümanınızın derlenmesi başarısız olur. Bu tür komutlara kırılğan denir: örneğin, `\section` (kısım) komutu içine `\footnote` (dipnot) koyduğunuzda hata verir. Bu kırılğan komutların korunmaya ihtiyacı vardır (hangimizin yok ki?). Onları korumak için, önlerine `\protect` komutu koyarsınız.

```
\section{Ben tedbirli adamım}
\protect\footnote{dipnotlarımı korurum}}
```

`\protect` komutu sadece ardından gelen komutu korur, onun argümanlarını korumaz. Çoğu durumlarda fazla bir `\protect` komutundan zarar gelmez.

Bölüm 3

Matematik Formülleri

Tamam, artık hazırsınız! \TeX 'in en kuvvetli olduğu alana artık girebiliriz: matematik formüllerini dizmek. Fakat, baştan söyleyeyim, bu bölümde konunun sadece yüzeyini kazımış olacağız. Burada anlatılanlar pekçok kişinin makale veya kitap yazması için yeterli olacaktır. Yine de, matematik dizgisinde çözemediğiniz bir sorunla karşılaşırsanız, umutsuzluğa kapılmayın. Sorununuz belki de daha kapsamlı olan $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ ¹ ile çözülebilir.

3.1 Genel

$\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 'in matematik formülleri dizmek için özel bir kipi vardır. Matematik formülleri iki türlü dizilebilir: Birincisi, paragrafı bozmadan metin içine, ikincisi de paragrafı yarıda kesip ayrı bir satıra dizilebilir.

Bir paragraf içine dizilen matematik formülü ya $\backslash($ ile $\backslash)$ arasına, yahut $\$$ ile $\$$ arasına, veyahut da $\backslash\text{begin}\{\text{math}\}$ ile $\backslash\text{end}\{\text{math}\}$ arasına dizilir.

$\$a\$$ kare ile $\$b\$$ karenin toplamı $\$c\$$ kare olur. Veya, $\$c^{\{2\}}=a^{\{2\}}+b^{\{2\}}\$$ matematik formülüyle ifade edilir.

a kare ile b karenin toplamı c kare olur. Veya, $c^2 = a^2 + b^2$ matematik formülüyle ifade edilir.

$\backslash\text{TeX}\{\}$ 'in yunanca yazılışı $\backslash(\tau\epsilon\chi\backslash)$.
 $100\text{-m}\$^{\{3\}}\$$ su
Bir $\backslash\text{begin}\{\text{math}\}\heartsuit\backslash\text{end}\{\text{math}\}$ iki beyine eşittir.

\TeX 'in yunanca yazılışı $\tau\epsilon\chi$.
 100 m^3 su
Bir \heartsuit iki beyine eşittir.

¹ *Amerikan Matematik Derneği* tarafından hazırlanan bu program $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 'in çok daha güçlü bir sürümü olup, \TeX dağıtımının yeni sürümleriyle birlikte verilmektedir. eğer bilgisayarınızda kurulmamışsa, `macros/latex/required/amslatex` adresinden indirebilirsiniz. Bu kitaptaki örneklerin çoğu $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ ile hazırlanmıştır.

Daha uzun formüller veya denklemler olduğunda, bunları *sergilemek*, yani paragraftan ayrı dizmek gerekir. Bu durumda, formülünüzü ya `\[` ile `\]` arasına, yahut da `\begin{displaymath}` ile `\end{displaymath}` arasına girersiniz.

```
$a$ kare ile $b$ karenin
toplamaı $c$ kare olur. Veya,
\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}
olur. Örneđin:
\[ 3^2+4^2=5^2 \]
```

a kare ile b karenin toplamı c kare olur.
Veya,

$$a^2 + b^2 = c^2$$

olur. Örneđin:

$$3^2 + 4^2 = 5^2$$

L^AT_EX'in denklemlerinizi numaralandırmasını istiyorsanız `equation` (denklem) ortamına geçmeniz gerekir. O zaman, denkleminize bir de `\label` (işaret) koyar ve metin içinde `\ref` komutuyla ona atıfta bulunursunuz. (`amsmath` paketinde atıflar `\eqref` komutu ile yapılır.)

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
Denklem \ref{eq:eps} gözönüne
alınırsa \ldots{} Denklem
\eqref{eq:eps} de aynı işi görür.
```

$$\epsilon > 0 \quad (3.1)$$

Denklem 3.1 gözönüne alınırsa ... Denklem (3.1) de aynı işi görür.

Paragraf içi denklemlerle, sergilenmiş denklemler arasındaki dizgi farkına dikkat edin:

```
Burada $\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$ ifadesi\ldots
```

Burada $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$ ifadesi...

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

L^AT_EX'de *matematik kipi* ile *metin kipi* arasında önemli farklar vardır. Örneđin, *matematik* kipinde:

1. Boşlukların ve satır kesimlerinin genelde bir önemi yoktur, çünkü L^AT_EX tüm boşlukları matematik ifadenin kurgusundan veya `\,`, `\quad` ve `\qquad` gibi ayraçlardan kendisi oluşturur.
2. Boş satırlara izin yoktur. Her formüle sadece bir paragraf.

3. Her harf bir değişkenin adı gibi ele alınır ve öyle dizilir. Eğer, formül içine düz yazıyla ve normal aralıklarla bir metin yazacaksanız, bunu `\textrm{...}` komutuyla girmeniz gerekir (bu konuda Sayfa 50 deki Kısım 3.7 de daha fazla bilgi vardır.)

```
\begin{equation}
\forall x \in \mathbf{R}:
\quad x^2 \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R} : \quad x^2 \geq 0 \quad (3.2)$$

```
\begin{equation}
\textrm{Her } x \in \mathbf{R}
\textrm{ için:} \quad x^2 \geq 0
\end{equation}
```

$$\text{Her } x \in \mathbf{R} \text{ için:} \quad x^2 \geq 0 \quad (3.3)$$

Matematikçiler kullandıkları sembollerin yazımında titizdirler: Yukardaki formülde kalın **R** yerine, ‘karatahta kalını’, yani çift çizgili R kullanmak isterler. Bunun için `amsfonts` veya `amssymb` paketlerinde `\mathbb` yazıtipi vardır. Son örneği bununla yazarsak:

```
\begin{displaymath}
\textrm{Her } x \in \mathbb{R}
\textrm{ için:} \quad x^2 \geq 0
\end{displaymath}
```

$$\text{Her } x \in \mathbb{R} \text{ için:} \quad x^2 \geq 0$$

3.2 Matematik Kipinde Gruplandırma

Matematik kipindeki komutların çoğu kendisinden sonra gelen ilk karaktere etki ederler. Bir komutun çok sayıda karaktere uygulanmasını istiyorsanız, çengelli parantez `{...}` kullanarak onları gruplandırmanız gerekir.

```
\begin{equation}
a^{x+y} \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \quad (3.4)$$

3.3 Bir Matematik Formülünün Yapıtaşları

Bu kısımda matematik formülleri dizmenin en önemli komutlarını anlatacağız. Matematik sembollerini dizmekte kullanılan komutların bir listesini görmek için, Sayfa 54 deki Kısım 3.10 a bir bakın.

Küçük Yunanca harfler `\alpha`, `\beta`, `\gamma`, ..., komutlarıyla, büyük harfleri ise `\Gamma`, `\Delta`, ...² komutlarıyla girilir.

²Yunancada büyükharf Alfa tanımlanmamıştır, çünkü bildiğimiz A harfiyle aynıdır. Yeni matematik kodlamasında bu durum değişecektir.

`\lambda, \xi, \pi, \mu, \Phi, \Omega`

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

Üsler ve İndisler, sırasıyla `^` ve `_` karakteriyle girilirler.

`a_{1} \quad x^{2} \quad $e^{-\alpha t}$ \quad a_{ij}^3 \quad $e^{x^2} \neq e^{x^2}$`

$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$
 $e^{x^2} \neq e^{x^2}$

Karekök `\sqrt` komutuyla girilir; n . kök yazmak isterseniz, `\sqrt[n]` girersiniz. Karekök işaretinin boyunu L^AT_EX otomatik ayarlar. Sadece kök işareti gerekiyorsa, `\surd` komutunu kullanabilirsiniz.

`\sqrt{x}$ \quad $\sqrt{x^2+\sqrt{y}}$ \quad $\sqrt[3]{2}$ \quad $\surd[x^2+y^2]$`

$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$
 $\sqrt{x^2 + y^2}$

`\overline` ve `\underline` komutları bir ifadenin üstüne veya altına yatay bir çizgi çekerler.

`\overline{m+n}`

$\overline{m+n}$

`\overbrace` ve `\underbrace` komutları bir ifadenin üstüne veya altına yatay bir çengel atarlar.

`\underbrace{a+b+\cdots+z}_{26}`

$\underbrace{a+b+\cdots+z}_{26}$

Değişken adlarının üstüne ok veya tilda gibi aksanlar eklemek için Sayfa 54 Tablo 3.1 deki komutları kullanabilirsiniz. Birden fazla karakter üzerine şapka veya tilda işareti koymak isterseniz, `\widehat` ve `\widetilde` komutları kullanırsınız. ' ' sembolü türev işaretini verir.

`\begin{displaymath} y=x^2 \quad y'=2x \quad y''=2 \end{displaymath}`

$y = x^2 \quad y' = 2x \quad y'' = 2$

Vektörler değişkenin üstüne küçük bir ok işareti koyularak gösterilirler. Bu işi `\vec` komutu yapar. A dan B ye vektörü göstermek için `\overrightarrow` ve `\overleftarrow` adlı iki komut daha vardır.

`\begin{displaymath} \vec{a} \quad \overrightarrow{AB} \end{displaymath}`

$\vec{a} \quad \overrightarrow{AB}$

İki değişkenin çarpımı genellikle araya nokta koymadan yanyana yazılarak yapılır. Fakat, bazan okuyucunun formüldeki gruplandırmaları görebilmesi için noktayı açıkça göstermek gerekebilir. Bu durumlarda `\cdot` komutu kullanmahsınız:

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2
   {\tau}_1 \cdot {\tau}_2
\end{displaymath}
```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

Matematik kipinde değişkenler italik harflerle gösterilirse de, logaritma veya sinüs gibi fonksiyon adları düz yazıtipinde dizilirler. En önemli fonksiyon adları için \LaTeX şu komutları sunar:

```
\arccos \cos \csc \exp \ker \limsup
\arcsin \cosh \deg \gcd \lg \ln
\arctan \cot \det \hom \lim \log
\arg \coth \dim \inf \liminf \max
\sinh \sup \tan \tanh \min \Pr
\sec \sin
```

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

Modülo fonksiyonu için iki komut vardır: “ $a \bmod b$ ” türü ikili işlemci için `\bmod` komutu, ve “ $x \equiv a \pmod{b}$ ” türü ifadeler için `\pmod` komutu.

```
\$a\bmod b\$
\$x\equiv a \pmod{b}\$
```

$$a \bmod b$$

$$x \equiv a \pmod{b}$$

Kesirler `\frac{...}{...}` komutuyla dizilirler. Birinci grup `{ }` içine pay, ikinci grup `{ }` içine payda yazılır. Küçük kesirler için, örneğin $1/2$ yazmak daha güzel gösterir.

```
\$1\frac{1}{2}\$~saat
\begin{displaymath}
\frac{x^2}{k+1} \qquad
x^{\frac{2}{k+1}} \qquad
x^{1/2}
\end{displaymath}
```

$$1\frac{1}{2} \text{ saat}$$

$$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

Binom katsayıları ve benzerlerini dizmek için `amsmath` paketindeki `\binom` komutunu kullanabilirsiniz.

```
\begin{displaymath}
\binom{n}{k} \qquad \mathrm{C}_n^k
\end{displaymath}
```

$$\binom{n}{k} \quad C_n^k$$

İkili işlemcilerde bazı semboller üst üste bindirilebilir. `\stackrel` komutu birinci argümanındaki sembolü, normal boyda yazılan ikincinin üzerine yazar.

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

integral işareti `\int` komutuyla, **sigma toplama işareti** `\sum` ile, ve **çarpım işareti** `\prod` ile dizilir. Alt ve üst limitler `\sim` ve `_` işaretleriyle, yani üs ve indis gibi girilirler.³

```
\begin{displaymath}
\sum_{i=1}^n \qquad \int_0^{\frac{\pi}{2}} \qquad \prod_{\epsilon}
\end{displaymath}
```

$$\sum_{i=1}^n \quad \int_0^{\frac{\pi}{2}} \quad \prod_{\epsilon}$$

Karmaşık ifadelerde indisler ve üsleri daha iyi yerleştirmek için `amsmath` paketinde iki seçenek daha vardır: `\substack` komutu ve `subarray` ortamı:

```
\begin{displaymath}
\sum_{\substack{0 < i < n \\ 1 < j < m}}
P(i, j) =
\sum_{\subarray{1}
i \in I \\
1 < j < m}
\end{subarray}} Q(i, j)
\end{displaymath}
```

$$\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) = \sum_{\substack{i \in I \\ 1 < j < m}} Q(i, j)$$

\TeX her türlü parantez ve diğer gruplandırıcı işaretler için (örneğin, `[` `<` `||` `↓`) çok sayıda sembole sahiptir. Yuvarlak ve köşeli parantezler klavyedeki yerlerinden, çengelli parantez `\{` olarak girilebilirse de, diğer tüm gruplandırıcılar özel komutlarla (örneğin, `\updownarrow`) girilirler. Kullanılabilecek tüm gruplandırıcı işaretler Sayfa 56 deki Tablo 3.7 de listelenmiştir.

```
\begin{displaymath}
\{a, b, c\} \neq \{a, b, c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

Grup açıcı bir sembolün önüne `\left` (sol) komutu, veya grup kapatıcı bir sembolün önüne `\right` komutu getirirseniz, \TeX onları otomatik olarak

³ $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 'de çok katlı üs ve indis bulunur.

en uygun boyda dizer. Her `\left` komutuna karşılık gelen bir `\right` komutu mutlaka bulunmalıdır. Fakat, bunların doğru boyda dizilmesi için, ikisi de aynı satırda yer almalıdır. Sağ tarafta gruplandırıcı işaret istemiyorsanız, görünmeyen `'\right.'` komutu kullanırız.

```
\begin{displaymath}
1 + \left( \frac{1}{1-x^2} \right) ^3
\end{displaymath}
```

$$1 + \left(\frac{1}{1-x^2} \right)^3$$

Ama bazan, gruplandırıcı sembolün boyunu elle ayarlamak gerekebilir. Bunun için, gruplandırıcı komutun önüne `\big`, `\Big`, `\bigg` veya `\Bigg` komutlarından birini koyarsınız.⁴

```
$$\Big( (x+1) (x-1) \Big) ^2$$\
$\big(\Big(\bigg(\Bigg(\quad
$\big\}\Big\}\bigg\}\Bigg\}$
\quad
$\big\|\Big\|\bigg\|\Bigg\|\$
```

$$\left((x+1)(x-1) \right)^2$$

$$\left(\left(\left(\left(\right) \right) \right) \right) \quad \left\| \left\| \left\| \left\| \left\| \right. \right.$$

Formüllerde **üç nokta** koymak için değişik komutlar vardır. `\ldots` komutu satır tabanına, `\cdots` komutu satırın orta yüksekliğine üç nokta koyar. Bunlara ek olarak, `\vdots` komutu dikey doğrultuda ve `\ddots` komutu çapraz doğrultuda üç nokta dizer. Başka bir örnek için [Kısım 3.5](#) e bakınız.

```
\begin{displaymath}
x_1, \ldots, x_n \quad \quad \quad x_1 + \cdots + x_n
\end{displaymath}
```

$$x_1, \dots, x_n \quad x_1 + \cdots + x_n$$

3.4 Matematikte Boşluklar

TeX'in bir formül içindeki bıraktığı boşluklar uygun düşmüyorsa, özel boşluk komutlarıyla ayarlanabilirler. Küçük aralıklar koyan komutlar şunlardır: `\,` $\frac{3}{18}$ quad (U) uzunlukta, `\:` $\frac{4}{18}$ quad (U) uzunlukta ve `\;` $\frac{5}{18}$ quad (U) uzunlukta boşluk açar. Geribölü `_` işareti orta boy aralık, `\quad` (□) ve `\qqquad` (□□□□) komutları daha geniş boşluklar içindir. Bir `\quad` aralık, kullanılan yazıtipindeki 'M' harfinin genişliğine eşittir. `\!` komutu eksi bir aralıktır, yani $-\frac{3}{18}$ quad (U) kadar aralığı küçültür.

⁴Eğer punto değiştiren bir komut kullanılmışsa, veya doküman sınıfında `11pt`, `12pt` opsiyonlarından biri belirtilmişse, bu komutlar beklenen boyda dizilmezler. Bu sorunu gidermek için `exscale` veya `amsmath` paketlerini kullanın.

```

\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\int_D g(x,y)\ud x \ud y
\end{displaymath}
yerine
\begin{displaymath}
\int\!\!\!\int_D g(x,y)
\ , \ \ud x\ , \ \ud y
\end{displaymath}

```

yerine

$$\int\int_D g(x,y)dx dy$$

$$\int\!\!\!\int_D g(x,y) dx dy$$

Burada diferansiyel işareti ‘d’ nin düz dizildiğine dikkat edelim. Bazı kitaplarda buna gerek görülmez.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ ’de çok katlı integraller arasındaki mesafeyi doğru ayarlayan `\iint`, `\iiint`, `\iiiint`, ve `\idotsint` komutları vardır. `amsmath` paketini yüklerseniz, yukardaki örneği şöyle dizebilirsiniz:

```

\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_D \ , \ \ud x \ , \ \ud y
\end{displaymath}

```

$$\iint_D dx dy$$

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ paketiyle dağıtılan `testmath.tex` dokümanında veya *The $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ Companion* [3] kitabının 8. Bölümünde daha fazla bilgi bulabilirsiniz.

3.5 Düşey Hizalanmış İfadeler

Matrisler ve tablo halinde sıralanmış ifadeler için, `array` ortamı kullanılır. Kullanımı aynen `tabular` ortamı gibidir. Satırlar `\right` komutuyla kesilir.

```

\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}

```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

`array` ortamı bir taraftan gruplandırılmış büyük ifadeler için de kullanılır. Gruplandırmayı sağ tarafta, görünmeyen “`\right.`” komutuyla tamamlarsınız:

```

\begin{displaymath}
y = \left\{ \begin{array}{ll}
a & \text{\textit{ $d > c$ ise}} \\
b+x & \text{\textit{ sabahları}} \\
l & \text{\textit{ gün boyunca}}
\end{array} \right.
\end{displaymath}

```

$$y = \begin{cases} a & d > c \text{ ise} \\ b+x & \text{sabahları} \\ l & \text{gün boyunca} \end{cases}$$

tabular ortamında olduğu gibi, array ortamında da matris elemanlarını birbirinden ayıran çizgiler çekebilirsiniz:

```
\begin{displaymath}
\left(\begin{array}{c|c}
1 & 2 \\ \hline
3 & 4
\end{array}\right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}\right)$$

Birkaç satıra taşan formüller veya denklem sistemleri için, equation ortamı yerine, eqnarray ve eqnarray* ortamları kullanılır. eqnarray ortamında her denkleme bir numara verilir, eqnarray* numaralandırma yapmaz.

eqnarray ve eqnarray* ortamlarında denklemler 3 sütunlu bir tablo gibi dizilirler. Birinci sütuna denklemin sol tarafı, orta sütun eşit (=) işareti veya kullanacağımız diğer bir işlemci, üçüncü sütuna ise denklemin sağ tarafı girilir. Sütunlar & işaretiyle ayrılır ve \\ komutu satırı keser.

```
\begin{eqnarray}
f(x) & = & \cos x \\
f'(x) & = & -\sin x \\
\int_0^x f(y)dy & = & \sin x
\end{eqnarray}
```

$$\begin{aligned} f(x) &= \cos x & (3.5) \\ f'(x) &= -\sin x & (3.6) \\ \int_0^x f(y)dy &= \sin x & (3.7) \end{aligned}$$

Dikkat ederseniz, eşit işaretinin iki tarafında biraz daha büyük boşluk oluşur. Bu fazlalığı, aşağıdaki örnekteki gibi, \setlength\arraycolsep{2pt} komutu ile azaltabilirsiniz.

Uzun denklemler kendiliğinden uygun parçalara bölünmezler. Yazar, bunların nereden bölüneceğini ve ne kadar içerden başlayacağını belirtmelidir. Aşağıda örnekte bu sorunu halletmenin iki farklı yolu gösterilmiştir.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
& & \frac{x^7}{7!} + \dots
\end{eqnarray}}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (3.8)$$

```

\begin{eqnarray}
\lefteqn{ \cos x = 1 }
-\frac{x^2}{2!} + \{ \}
\quad \quad \quad \backslashnonumber\{ \}
& & \{ \} + \frac{x^4}{4!}
-\frac{x^6}{6!} + \{ \} \cdots
\end{eqnarray}

```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \quad (3.9)$$

`\nonumber` (numarasız) komutu \LaTeX 'e bu denkleme numara vermemesini söyler.

Bu yöntemlerle denklemleri düşey yönde hizalandırmak zordur. `amsmath` paketinde çok daha güçlü seçenekler (`align`, `flalign`, `gather`, `multline` ve `split`) vardır.

3.6 Hayaletler

Hayaletler görünmez, ama bazı insanların aklını yine de meşgul ederler. \LaTeX 'teki hayaletler de böyledir. Görünmeyen, ama sayfada yer işgal eden bu komutla, bazı ince ayarlamalar yapabiliriz.

\LaTeX , `^` ve `_` komutlarıyla üs ve indisleri yerleştirdiğinde, düşey doğrultuda hizalanışları göze hoş gelmeyebilir. `` (hayalet) komutuyla, sayfa üzerinde görünmeyecek olan *metin* kadar genişlikte bir yer ayırmış oluruz. En iyisi, bunu iki örnekle anlatayım:

```

\begin{displaymath}
\{ \}^{\{12\}}_{\{6\}} \text{\texttrm{C}}
\quad \quad \quad \backslashqquad \text{\texttrm{yerine}} \quad \quad \quad \backslashqquad
\{ \}^{\{12\}}_{\{ \phantom{\{1\}6} \}} \text{\texttrm{C}}
\end{displaymath}

```

$${}_{6}^{12}\text{C} \quad \text{yerine} \quad {}_{6}^{12}\text{C}$$

```

\begin{displaymath}
\Gamma_{\{ij\}}^{\{k\}}
\quad \quad \quad \backslashqquad \text{\texttrm{yerine}} \quad \quad \quad \backslashqquad
\Gamma_{\{ij\}}^{\{ \phantom{ij} \}k}
\end{displaymath}

```

$$\Gamma_{ij}^k \quad \text{yerine} \quad \Gamma_{ij}^{k}$$

3.7 Matematik Yazıtipi Puntosu

\TeX matematik kipinde yazıtipinin punto büyüklüğünü, ele aldığı ifadeye göre seçer. Örneğin, üsler ve indisler daha küçük puntoda dizilirler. Bir denklemin bazı yerlerini düz harflerle dizmek istiyorsanız `\texttrm{ }` komutu kullanamazsınız, çünkü bu komut geçici olarak metin kipine geçiş demektir, üs ve indislerin otomatik punto ayarını artık yapamaz. Bunun yerine,

kısa ifadeler için `\mathrm` komutu kullanılır. Bu komut kısa ifadeler içindir, boşluklar yine gözönüne alınmaz ve aksanlı harfler yine girilmez.⁵

```
\begin{equation}
P_{\text{atm}}=\rho g h \quad \quad \quad
P_{\mathrm{atm}}=\rho g h
\end{equation}
```

$$P_{\text{atm}} = \rho g h \quad P_{\mathrm{atm}} = \rho g h \quad (3.10)$$

Bazan uygun puntoyu sizin \LaTeX 'e bildirmeniz gerekebilir. Matematik kipinde bunu yapan dört stil komutu vardır:

```
\displaystyle (123), \textstyle (123), \scriptstyle (123) ve
\scriptscriptstyle (123).
```

Stil değiştirmek alt ve üst sınırların dizilişini de etkiler.

```
\begin{displaymath}
\frac{\sum_{i=1}^n(x_i-\overline{x})
(y_i-\overline{y})}{
\left[\sum_{i=1}^n(x_i-\overline{x})^2
\sum_{i=1}^n(y_i-\overline{y})^2\right]^{1/2}}
\end{displaymath}
```

$$\frac{\sum_{i=1}^n(x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n(x_i - \bar{x})^2 \sum_{i=1}^n(y_i - \bar{y})^2\right]^{1/2}}$$

Bu örnekte kesrin payı ile paydası farklı stillerde dizilmiştir. Paydadaki köşeli parantezler de `\left[` ve `\right]` komutlarının yapabileceğinden daha büyüktürler. `\biggl` ve `\biggr` komutları, sırasıyla sol ve sağ parantezler içindir.

3.8 Teoremler, Yasalar, ...

Matematik dokümanları yazarken, bazı paragraflara çoğu zaman “Teorem”, “Tanım”, “Aksiyoim” gibi başlıklar atmak isteyeceksinizdir.

```
\newtheorem{kısa adı}[sayaç]{tam adı}[section]
```

Bu komut dokümanın sahanlık kısmına bir defa konur. Burada *kısa adı*, teoremi tanımaya yarayan kısa bir anahtar kelimedir. *tam adı* teoremin kağıda basılacak adı olur. Köşeli parantezler içine konulanlar opsiyonel parametrelerdir. Örneğin, *sayaç* içine bir önceki teoremin *kısa adını* koyarsanız, teoremin numaralandırması oradan devam eder. *section* (kısım) kullanıldığında, teoremin kısım bilgileri ile numaralandırmasını ister.

Bu komutu sahanlık kısmına koyduktan sonra, artık metin içinde teorem yazmak istediğinizde şöyle girersiniz:

⁵ $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ 'teki `amsmath` paketinde `\textrm` komutu yazı puntosunu ayarlar.

```
\begin{kısa adı}[tam adı]
Bu teorem çok kısadır.
\end{kısa adı}
```

Bu kısa anlatım yeterli sayılır. Aşağıdaki örneklerden görüleceği üzere, `\newtheorem` ortamı karmaşık ve anlaması zor bir konudur.

```
\newtheorem{mur}{Murphy Yasası}[section]
\begin{mur}
Bir işi yapmak için
birden fazla yol
varsa, ve bu
yollardan biri
felakete yolaçuyorsa,
o yolu deneyen birisi
mutlaka çıkacaktır.
\end{mur}
```

Murphy Yasası 3.8.1. *Bir işi yapmak için birden fazla yol varsa, ve bu yollardan biri felakete yolaçuyorsa, o yolu deneyen birisi mutlaka çıkacaktır.*

Bu örnekte “Murphy” teoreminin numaralandırması, `[section]` komutuyla mevcut bölüm ve kısım numarasına bağlanmıştır. Başka bir birim, örneğin bölüm veya altkısım da kullanabilirsiniz.

```
\newtheorem{yasa}{Yasa}
\newtheorem{yon}[yasa]{Yönetmelik}
% Doküman içinde kullan
\begin{yasa} \label{patron}
Patron haklıdır.
\end{yasa}
\begin{yon}[önemli]
Patron daima haklıdır.
\end{yon}
\begin{yasa}
Eğer patron haksızsa,
\ref{patron} No.lu
yasaya bakın.
\end{yasa}
```

Yasa 1. *Patron haklıdır.*

Yönetmelik 2 (önemli). *Patron daima haklıdır.*

Yasa 3. *Eğer patron haksızsa, 1 No.lu yasaya bakın.*

Bu örnekte önce, kısa adları (`yasa`) ve (`yon`) olan “Yasa” ve “Yönetmelik” adlı iki teorem tanımlanmıştır. Yönetmeliğin opsiyonel sayaç parametresi `[yasa]` olarak seçildiği için, onun numarası da `yasa` numarasını takip edecektir. Ayrıca, yasanın ilk kullanıldığı yere (`\label{patron}`) komutuyla bir işaret konularak, daha sonra ona atıfta bulunulmuştur.

`amsthm` paketinde `\newtheoremstyle{stil}` komutuyla, önceden tasarlanmış üç ayrı stil kullanılabilir: `definition` (tanım: kalın başlık, düz metin), `plain` (sade: kalın başlık, italik metin) ve `remark` (yorum: italik başlık, düz metin).

Bu pakette önce teoremler seçilen bir stilde tanımlanırlar:

```
\theoremstyle{definition} \newtheorem{yasa}{Yasa}
\theoremstyle{plain}      \newtheorem{lema}[yasa]{Lema}
\theoremstyle{remark}    \newtheorem*{ibo}{İbrahim}
```

amsthm paketinde `proof` (kanıt, ispat) adlı bir teorem daha vardır.

```
\begin{proof}
Sadeleştirme yapılırsa
\math[E=mc^2]
\end{proof}
```

Kanıt. Sadeleştirme yapılırsa

$$E = mc^2$$

□

İspat sonunu bildiren □ işareti bazan son satırda yalnız kalırsa, `\qedhere` komutuyla onu doğru yere oturtabilirsiniz:

```
\begin{proof}
Sadeleştirme yapılırsa
\math[E=mc^2] \qedhere
\end{proof}
```

Kanıt. Sadeleştirme yapılırsa

$$E = mc^2$$

□

3.9 Kalın Semboller

L^AT_EX'te kalın matematik sembolleri dizmek zordur; bunun nedeni belki de amatör dizgicilerin onu aşırı kullanmasını önlemek için olabilir. Yazıtipi değiştirme komutu olan `\mathbf` kalın harfleri verir, fakat bunlar düz harflerdir, oysa matematik harfleri italik olmalıdır. Ayrı bir `\boldmath` komutu vardır, ama o *sadece matematik kipi dışında kullanılabilir*. Bu komut, sembolleri de kalın dizer.

```
\begin{displaymath}
\mu, M \quad \mathbf{M} \quad \mu, M
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

$$\mu, M \quad \mathbf{M} \quad \mu, M$$

Bu örnekte, önce matematik kipinde `\mathbf` yazıtipiyle kalın yazılmış, sonra `\mbox{ }` komutuyla matematik dışına çıkılıp `\boldmath` yazıtipiyle aynı semboller italik yazılabilmıştır. Aradaki virgülün de, istemediğimiz halde, kalın dizildiğine dikkat edin.

`amsmath` içinde gelen `amsbsy` ve `bm` paketlerinde bu iş daha kolaydır, `\boldsymbol` komutuyla istenilen harf veya sembol kalın dizilir.

```
\begin{displaymath}
\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}$$

3.10 Matematik Semboller Listesi

Aşağıdaki tablolarda *matematik kipinde* kullanılacak tüm sembolleri bulabilirsiniz. 3.11–3.15 arası tabloları kullanabilmek için⁶ dokümanınızın preamble kısmında `amssymb` paketi yüklenmiş ve sistemde AMS matematik yazıtipleri kurulmuş olmalıdır. AMS matematik paketi ve yazıtipleri bilgisayarınızda kurulu değilse `macros/latex/required/amslatex` sitesinden indirebilirsiniz. Daha geniş bir semboller listesi `info/symbols/comprehensive` sitesinde bulunabilir.

Tablo 3.1: Matematik Kipi Aksanları.

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\tilde{a}	<code>\tilde{a}</code>
\grave{a}	<code>\grave{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>
\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>	\widehat{A}	<code>\widehat{A}</code>
\acute{a}	<code>\acute{a}</code>	\breve{a}	<code>\breve{a}</code>	\widetilde{A}	<code>\widetilde{A}</code>

Tablo 3.2: Yunan Harfleri.

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	υ	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

⁶Bu tablolar David Carlisle'in `symbols.tex` dokümanından uyarlanmış ve Josef Tkadlec'in önerileriyle ciddi değişiklikler yapılmıştır.

Tablo 3.3: İkili Bağıntılar.

Aşağıdaki sembollerin önüne `\not` komutu koyarsanız, üstüne çapraz bir çizgi çekerek olumsuz şeklini dizer.

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> or <code>\le</code>	\geq	<code>\geq</code> or <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset	<code>\sqsubset</code> ^a	\sqsupset	<code>\sqsupset</code> ^a	\bowtie	<code>\bowtie</code> ^a
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code> , <code>\owns</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\mid	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq	<code>\neq</code> or <code>\ne</code>

^aBu sembolü kullanabilmek için `latexsym` paketi yüklenmiş olmalıdır.

Tablo 3.4: İkili İşlemciler.

$+$	<code>+</code>	$-$	<code>-</code>	\triangleleft	<code>\triangleleft</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\triangleright	<code>\triangleright</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\star	<code>\star</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	$*$	<code>\ast</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\circ	<code>\circ</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\bullet	<code>\bullet</code>
\vee	<code>\vee</code> , <code>\lor</code>	\wedge	<code>\wedge</code> , <code>\land</code>	\diamond	<code>\diamond</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\uplus	<code>\uplus</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\amalg	<code>\amalg</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\dagger	<code>\dagger</code>
\triangleleft	<code>\bigtriangleleft</code>	\triangleright	<code>\bigtriangleright</code>	\ddagger	<code>\ddagger</code>
\triangleleft	<code>\lhd</code> ^a	\triangleright	<code>\rhd</code> ^a	\wr	<code>\wr</code>
\triangleleft	<code>\unlhd</code> ^a	\triangleright	<code>\unrhd</code> ^a		

Tablo 3.5: BÜYÜK İşlemciler.

Σ	<code>\sum</code>	\cup	<code>\bigcup</code>	\vee	<code>\bigvee</code>
\prod	<code>\prod</code>	\cap	<code>\bigcap</code>	\wedge	<code>\bigwedge</code>
\coprod	<code>\coprod</code>	\sqcup	<code>\bigsqcup</code>	\oplus	<code>\bigoplus</code>
\int	<code>\int</code>	\oint	<code>\oint</code>	\odot	<code>\bigodot</code>
\oplus	<code>\bigoplus</code>	\otimes	<code>\bigotimes</code>		

Tablo 3.6: Oklar.

\leftarrow	<code>\leftarrow</code> or <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>
\rightarrow	<code>\rightarrow</code> or <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff	<code>\iff</code> (bigger spaces)
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>
\Uparrow	<code>\Uparrow</code>	\uparrow	<code>\uparrow</code>
\Downarrow	<code>\Downarrow</code>	\Downarrow	<code>\Downarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>
\leadsto	<code>\leadsto</code> ^a		

^aBu sembolü kullanabilmek için `latexsym` paketi yüklenmiş olmalıdır.

Tablo 3.7: Gruplandırıcılar.

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>
$[$	<code>[</code> or <code>\lbrack</code>	$]$	<code>]</code> or <code>\rbrack</code>	\downarrow	<code>\downarrow</code>
$\{$	<code>\{</code> or <code>\lbrace</code>	$\}$	<code>\}</code> or <code>\rbrace</code>	\updownarrow	<code>\updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> </code> or <code>\vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>	\Updownarrow	<code>\Updownarrow</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>	$\ $	<code>\ </code> or <code>\Vert</code>
\lceil	<code>\lceil</code>				

Tablo 3.8: Büyük Gruplandırıcılar.

(<code>\lgroup</code>)	<code>\rgroup</code>	{	<code>\lmoustache</code>
	<code>\arrowvert</code>		<code>\Arrowvert</code>		<code>\bracevert</code>
}	<code>\rmoustache</code>				

Tablo 3.9: Değişik Semboller.

...	<code>\dots</code>	...	<code>\cdots</code>	:	<code>\vdots</code>	⋯	<code>\ddots</code>
\hbar	<code>\hbar</code>	\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>	\wp	<code>\wp</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\mho ^a	<code>\mho</code>	∂	<code>\partial</code>
'	<code>'</code>	'	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>	\square	<code>\Box</code> ^a	\diamond	<code>\Diamond</code> ^a
\perp	<code>\bot</code>	\top	<code>\top</code>	\angle	<code>\angle</code>	\surd	<code>\surd</code>
\diamond	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg	<code>\neg</code> or <code>\lnot</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>

^aBu sembolü kullanabilmek için `latexsym` paketi yüklenmiş olmalıdır.

Tablo 3.10: Matematiksel Olmayan Semboller.

Bu semboller metin kipinde de kullanılabilirler.

†	<code>\dag</code>	§	<code>\S</code>	©	<code>\copyright</code>	®	<code>\textregistered</code>
‡	<code>\ddag</code>	¶	<code>\P</code>	£	<code>\pounds</code>	%	<code>\%</code>

Tablo 3.11: AMS Gruplandırıcıları.

┌	<code>\ulcorner</code>	┐	<code>\urcorner</code>	└	<code>\llcorner</code>	┘	<code>\lrcorner</code>
	<code>\lvert</code>		<code>\rvert</code>		<code>\lVert</code>		<code>\rVert</code>

Tablo 3.12: AMS Yunanca ve İbranice.

\digamma	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>	\gimel	<code>\gimel</code>	\daleth	<code>\daleth</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	----------	---------------------	-----------	----------------------

Tablo 3.13: AMS İkili Bağıntılar.

\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\doteqdot	<code>\doteqdot</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\eqslantless	<code>\eqslantless</code>	\eqslantgtr	<code>\eqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll or \llless	<code>\lll</code> or <code>\llless</code>	\ggg	<code>\ggg</code>	\circeq	<code>\circeq</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\thicksim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\gtreqqless	<code>\gtreqqless</code>	\thickapprox	<code>\thickapprox</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\approxeq	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsim	<code>\backsim</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\subseteqq	<code>\subseteqq</code>	\supseteqq	<code>\supseteqq</code>	\Vdash	<code>\Vdash</code>
\shortparallel	<code>\shortparallel</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\blacktriangleleft	<code>\blacktriangleleft</code>	\sqsupset	<code>\sqsupset</code>	\backepsilon	<code>\backepsilon</code>
\vartriangleright	<code>\vartriangleright</code>	\because	<code>\because</code>	\varpropto	<code>\varpropto</code>
\blacktriangleright	<code>\blacktriangleright</code>	\Subset	<code>\Subset</code>	\between	<code>\between</code>
\trianglerighteq	<code>\trianglerighteq</code>	\smallfrown	<code>\smallfrown</code>	\pitchfork	<code>\pitchfork</code>
\vartriangleleft	<code>\vartriangleleft</code>	\shortmid	<code>\shortmid</code>	\smallsmile	<code>\smallsmile</code>
\trianglelefteq	<code>\trianglelefteq</code>	\therefore	<code>\therefore</code>	\sqsubset	<code>\sqsubset</code>

Tablo 3.14: AMS Oklar.

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>
\multimap	<code>\multimap</code>	\Uparrow	<code>\Uparrow</code>
\downdownarrows	<code>\downdownarrows</code>	\Uparrow	<code>\Uparrow</code>
\upharpoonright	<code>\upharpoonright</code>	\Downarrow	<code>\Downarrow</code>
\rightsquigarrow	<code>\rightsquigarrow</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>

Tablo 3.15: AMS Olumsuz İkili Bağıntılar ve Oklar.

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\nsubseteqeq
\nleqslant	\ngeqslant	\nsupseteqeq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\nleqq	\ngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\nprecneqq	\nsuccneqq	\nvDash
\nprecnsim	\succnsim	\nVdash
\nprecnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\nsubseteq	\nsupseteq	\ntrianglelefteq
\subseteqeq	\supseteqeq	\ntrianglerighteq
\nleftarrow	\rightarrow	\nleftrightarrow
\nLeftarrow	\nrightarrow	\nLeftrightarrow

Tablo 3.16: AMS İkili İşlemciler.

\dotplus	\centerdot	
\ltimes	\rtimes	\divideontimes
\doublecup	\doublecap	\smallsetminus
\veebar	\barwedge	\doublebarwedge
\boxplus	\boxminus	\circleddash
\boxtimes	\boxdot	\circledcirc
\intercal	\circledast	\rightthreetimes
\curlyvee	\curlywedge	\leftthreetimes

Tablo 3.17: AMS Değişik Semboller.

\hbar	<code>\hbar</code>	\hbar	<code>\hslash</code>	\mathbb{k}	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\textcircled{S}	<code>\circledS</code>
\triangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
∇	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\Game	<code>\Game</code>
\diamond	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\bigstar	<code>\bigstar</code>
\sphericalangle	<code>\angle</code>	\sphericalangle	<code>\measuredangle</code>		
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\backprime	<code>\backprime</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\varnothing	<code>\varnothing</code>
\eth	<code>\eth</code>	\sphericalangle	<code>\sphericalangle</code>	\mho	<code>\mho</code>

Tablo 3.18: Matematik Yazıtipleri.

Örnek	Komut	Gerekli paket
$ABCDEabcde1234$	<code>\mathrm{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathit{ABCDE abcde 1234}</code>	
$ABCDEabcde1234$	<code>\mathnormal{ABCDE abcde 1234}</code>	
\mathcal{ABCDE}	<code>\mathcal{ABCDE abcde 1234}</code>	
\mathscr{ABCDE}	<code>\mathscr{ABCDE abcde 1234}</code>	<code>mathrsfs</code>
$\mathfrak{ABCDEabcde1234}$	<code>\mathfrak{ABCDE abcde 1234}</code>	<code>amsfonts</code> or <code>amssymb</code>
$\mathbb{ABCDE}\mathbb{K}\mathbb{H}\mathbb{Z}$	<code>\mathbb{ABCDE abcde 1234}</code>	<code>amsfonts</code> or <code>amssymb</code>

Bölüm 4

Özel Konular

Büyük bir dokümanı bir araya getirmeye çalıştığınızda, izin çıkarma, kaynakça oluşturma gibi pekçok zahmetli işte \LaTeX size yardımcı olur. \LaTeX 'in bu özel araçları ve diğer zenginleştirilmiş yönleri konusunda daha geniş bilgi *\LaTeX Manual* [1] ve *The \LaTeX Companion* [3] da bulunabilir.

4.1 PostScript Formatında Grafik Ekleme

\LaTeX 'de resim, grafik veya tablo gibi yüzer-gezer nesneler, `figure` ve `table` ortamlarında ele alınırlar.

\LaTeX 'te bir grafiği çizdirmek için birkaç yol vardır. Bunlardan bazıları Bölüm 5 de anlatılacaktır. Bu konuda daha fazla bilgi için yine *The \LaTeX Companion* [3] ve *\LaTeX Manual* [1] kaynaklarına başvurun.

Fakat, bir dokümana grafik eklemek için başka bir yol daha vardır: Özel bir çizim programı (örneğin, CorelDraw, Freehand, GNUPlot, ...) ile grafiği çizersiniz, sonra bu grafiği dokümanınıza eklersiniz. Bu yolu seçerseniz yine \LaTeX size pekçok araç sunar. Bu kitapta sadece Encapsulated POSTSCRIPT (EPS) formatındaki grafiklerin nasıl ekleneceğini anlatacağız; çünkü en yaygın grafik formatı budur ve dokümana eklenmesi kolaydır. İlke olarak, EPS formatındaki grafikleri yazıcıda basabilmek için, POSTSCRIPT dilinden anlayan bir yazıcınız olmalıdır. Fakat, `support/ghostscript` sitesinden indirebileceğiniz GHOSTSCRIPT programıyla her yazıcıda basabilirsiniz.

Grafik ekleme konusunda en kapsamlı program D. P. Carlisle'in yazdığı `graphicx` ve `graphics` paketleridir. Bunlar “grafik bohçası” denilen programlar kümesi içinde yer alırlar.¹

Burada `graphics` paketini nasıl kullanacağımızı anlatacağım. Bilgisayarınızda `graphics` paketinin kurulu olduğunu ve POSTSCRIPT yazıcınız (yoksa, `ghostscript` programının kurulmuş) olduğunu varsayıyorum. Aşağıdaki adımları sırayla izleyerek, dokümanınıza grafik ekleyebilirsiniz:

¹`macros/latex/required/graphics`

1. Herhangi bir çizim programında oluşturduğunuz grafiği EPS formatında saklayın (save) veya ihraç edin (export).²
2. Girdi dosyanızın sahanlık kısmına şu komutla `graphics` paketini yükleyin:

```
\usepackage[dvips]{graphics}
```

Burada opsiyonel `[dvips]` seçeneği dvi dosyasına grafiği eklerken `dvips` sürücüsünün kullanılmasını ister. \TeX 'in grafik eklemek için belli bir standardı yoktur, ama sürücü adını bilirse, dvi dosyasına `.eps` uzantılı grafik dosyasını yazıcının anlayacağı şekilde ekleyebilir.

3. Artık, grafiğinizi dokümanınız içinde eklemek istediğiniz yere, şu komutlarla girersiniz:

```
\begin{figure}[!hbt]
\centering
\includegraphics*{dosya}
\caption{altyazı}
\end{figure}
```

Burada *dosya* grafik dosyanızın adı, *altyazı* ise resmin altyazısı olarak girmek istediğiniz metindir. `\centering` komutu grafiği ortalar. Komutun sonuna yıldız (*) konulmuşsa, grafiğin ayrılan bölge dışına taşan kısımları da basılır, yıldız konulmazsa fazlalıklar kesilir.

`graphics` paketinde, resimleri istediğiniz gibi küçültmek, döndürmek, ... için komutlar vardır. Bunlardan en önemlileri Tablo 4.1 de gösterilmiştir.

Şu örnek konunun daha iyi anlaşılmasına yardımcı olacaktır:

```
\begin{figure}
\centering
\rotatebox{35}{\scalebox{0.4}{\includegraphics{test}}}
\caption{Bu bir test.}
\end{figure}
```

Bu örnekte `test.eps` dosyasında bulunan grafiğin önce boyu % 40 oranında küçültülmüş, sonra 35 derece döndürülmüştür.

4.2 Kaynakça

Dokümanınıza bir kaynakça koymak için `thebibliography` ortamında çalışmanız gerekir:

²Çizim programınız EPS formatında saklayamıyorsa, şöyle bir izleyin: Bilgisayarımıza POSTSCRIPT yazıcılardan birinin (örneğin Apple LaserWriter) sürücüsünü ekleyin. Sonra, bu sürücüyü kullanarak çizimi bir dosyaya yazdırın. Şansınız varsa, bu dosya EPS formatında olacaktır. Dikkat edin, EPS grafiği bir sayfadan daha büyük olmamalıdır.

Tablo 4.1: graphics Paketinin Önemli Komutları.

<code>\includegraphics* [x,ly] [{ux,uy}]{dosya}</code>
(<i>x,ly</i>) boyutlarındaki <i>dosya</i> adlı grafiği sol alt köşesi (<i>ux,uy</i>) koordinatlı yere gelecek şekilde yerleştirir.
<code>\rotatebox{açı}{dosya}</code>
<i>dosya</i> adlı grafiği verilen <i>açı</i> kadar saat yönü tersine döndürür.
<code>\reflectbox{dosya}</code>
<i>dosya</i> adlı grafiğin aynadan yansımış şeklini basar.
<code>\scalebox{rx}[ry]{dosya}</code>
Grafiği <i>rx</i> ve <i>ry</i> oranlarında yatay ve dikey yönde küçültür. Oranlar ondalık kesirdir, 0.5 yarıyarıya küçültür, 2.0 iki kat büyütür. Dikey oran verilmemişse, yatay oranla aynı varsayılır.

`\begin{thebibliography} . . . \end{thebibliography}`

Sonra, kitap veya makale kaynaklarının herbiri şöyle girilir:

`\bibitem[label]{işaret} kitap veya makale bilgileri . . .`

Doküman içinde bu kaynağa atıfta bulunmak için *işaret* parametresi kullanılır:

`\cite{işaret}`

Opsiyonel *label* parametresi konulmazsa, kaynaklar giriş sırasına göre numaralandırılır. `\begin{thebibliography}` komutundan sonra bir sayı konularak maksimum label sayısı bildirilebilir. Aşağıdaki örnekte, L^AT_EX'e kaynakçada en fazla {99} kaynağın yeralacağı bildirilmiştir:

```
Bu konuda Partl~\cite{pa}
şöyle diyor \ldots
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German TeX},
TUGboat Cilt-9, Sayı-1 (1988)
\end{thebibliography}
```

Bu konuda Partl [1] şöyle diyor ...

Kaynakça

[1] H. Partl: *German TeX*, TUGboat Cilt 9, Sayı 1 (1988)

Daha hacımlı kaynakçalar için BibTeX programını kullanmayı düşünebilirsiniz. Standart TeX dağıtımıyla gelen bu programla, size lazım olabilecek tüm kaynakları bir veritabanında toplar ve daha sonra bunlardan istediğiniz birkaçını çekip makalenizde kullanırsınız. Ayrıca BibTeX değişik yazım standartlarına göre kaynakçanın formatlamasını yapabilir.

4.3 Dizin

Bir kitabı kullanışlı yapan en önemli bölümü dizindir. L^AT_EX ve onun destek programı `makeindex` (veya, `makeidx`) ile dizin çıkarmak çok kolay bir iştir. Bu kitapta dizin çıkaran temel komutları öğreteceğiz. Daha derin bir açıklama için *The L^AT_EX Companion* [3] kitabına bakın.

L^AT_EX'in dizin yapabilmesi için, dokümanınızın sahanlığında iki komut vermeniz gerekir. Önce, `makeidx` paketi yüklenir:

```
\usepackage{makeidx}
```

Sonra, dizinleme komutlarını etkinleştirmek için, yine sahanlık kısmına şu komut girilir:

```
\makeindex
```

Artık metin içinde, dizine girmesini istediğiniz her kelime için şu komutu girersiniz:

```
\index{terim}
```

Burada *terim* dizin maddesi olan terim, kavram veya tanımdır. Dizinleme komutunu, o terimin geçtiği her yere değil, bakılmasını istediğiniz sayfalardaki yerlere koymanız doğru olur. Tablo 4.2 de terimlerin madde veya altmadde olarak nasıl girileceği örneklerle gösterilmiştir.

L^AT_EX girdi dosyanızı derlerken bu indis maddelerini, buldukları sayfa numaralarıyla birlikte özel bir dosyaya kaydeder. Bu dosyanın adı sizin girdi dosyanızla aynı, fakat uzantısı farklıdır (`.idx`). Bu dosya şimdi `makeindex` programından geçirilir:

```
makeindex dosyaadi
```

`makeindex` programı dizin maddelerini alfabetik sıraya göre dizip, `.ind` uzantılı diğer bir dosyaya aktarır. (Fakat, Türkçedeki aksanlı harflerle başlayan kelimeler doğru sırada yer almazlar. Bir metin yazıcı programla `.ind` dosyasını açıp bunları elle doğru yerlere taşımanız gerekir.) L^AT_EX girdi dosyanız tekrar derlendiğinde, bu sıralanmış dizin dokümanınızda yer alır. Bu-

Tablo 4.2: Dizin Maddesi Girme Örnekleri.

Örnek	Dizin Maddesi	Açıklama
<code>\index{kuvvet}</code>	kuvvet, 1	Düz madde
<code>\index{kuvvet!magnetik}</code>	magnetik, 3	'kuvvet'in altmaddesi
<code>\index{Türev@\textsl{Türev}}</code>	<i>Türev</i> , 2	Formatlı madde
<code>\index{Limit@\textbf{Limit}}</code>	Limit , 7	yukardaki gibi
<code>\index{Açı \textbf}}</code>	Açı, 3	Formatlı sayfa numarası
<code>\index{İvme \textit}}</code>	İvme, 5	yukardaki gibi
<code>\index{ecole@\'ecole}</code>	école, 4	aksanlı madde girişi

nun için, dizinin görünmesini istediğiniz yere (yani, doküman sonuna) şu komutu eklersiniz:

```
\printindex
```

L^AT_EX'le gelen `showidx` paketi dizin maddelerini, metnin sol marjı içine basılı olarak gösterir. Düzeltmeler yaparken ve dizini kontrol ederken, böyle bir çıktı üzerinde çalışmak çok kullanışlı olur.

`\index` komutu doğru kullanılmazsa dokümanınızın dizgisini etkileyebilir.

```
Enerji \index{enerji}.
Enerji\index{enerji}.
Noktanın yerine dikkat edin.
```

```
Enerji . Enerji. Noktanın yerine dikkat edin.
```

4.4 Tepelik ve Diplikler

Piet van Oostrum'un hazırladığı `fancyhdr` paketi³ birkaç basit komutla, doküman sayfalarınızın tepelik ve diplik bölgelerini özelleştirmenizi sağlar. Bu sayfanın tepesine bakarsanız, bu paketin bir uygulamasını görürsünüz.

Tepelik ve diplikleri özelleştirmede en büyük sorun bölüm ve kısım adlarını doğru numaralarla oraya koyabilmektir. L^AT_EX bunu iki aşamada yapar. Tepelik ve diplik tanımlarında, `\leftmark` komutuyla sol sayfalara bölüm adını, `\rightmark` komutuyla da sağ sayfalara kısım adını koyarsınız. Ne zaman bölüm veya kısım değişse, bu iki komutun aldığı değerler değişir. `\chapter` (bölüm) komutu `\rightmark` (sağ) ve `\leftmark` (sol) tepelikleri kendisi değiştirmez, `\chaptermark`, `\sectionmark`, veya `\subsectionmark`)

³ macros/latex/contrib/supported/fancyhdr. adresinden indirilebilir.

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% Şu iki komutla tepelikteki bölüm ve kısım
% başlıklarını küçük harfe çeviririz.
\renewcommand{\chaptermark}[1]{%
    \markboth{#1}{}}
\renewcommand{\sectionmark}[1]{%
    \markright{\thesection\ #1}}
\fancyhf{} % mevcut tepelik ve başlığı kaldır
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % çizgi için yer ayır
\fancypagestyle{plain}{%
    \fancyhead{} % sade sayfaların tepeliğini ve
    \renewcommand{\headrulewidth}{0pt} % çizgisini kaldır
}

```

Şekil 4.1: fancyhdr Paketiyle Tepelik Örneği.

komutlarını çağırarak `\rightmark` ve `\leftmark` içeriklerini onlara değiştirir. Tepelikteki bölüm adının görünümünü değiştirmek istiyorsanız, sadece `\chaptermark` komutunu yenilersiniz.

Şekil 4.1 de `fancyhdr` paketiyle bu kitaptaki tepeliklerin nasıl hazırlandığı gösterilmiştir. Daha fazla bilgi için, dipnotta verdiğim adresteki dokümanları incelemenizi tavsiye ederim.

4.5 Verbatim Paketi

Bu kitapta daha önce `verbatim ortamını` öğrenmiştiniz, yazılan bir metni, içinde \LaTeX komutları olsa bile, ‘olduğu gibi’ diziordu. Oysa, çok güçlü bir `verbatim paketi` daha vardır. Bu paket bazı işleri daha kolay yapabilirse de, asıl avantajı yeni bazı olanaklar tanımasıdır. `verbatim` paketinde şöyle bir komut vardır:

```
\verbatiminput{dosya adi}
```

Bu komutla herhangi bir girdi *dosyasının* içeriğini `verbatim` ortamına, tümüyle aktarabilirsiniz.

`verbatim` paketi ‘araçlar bohçası’ denilen bir grubun içinde bilgisayarınıza önceden kurulmuş olmalıdır. Bu paket hakkında daha fazla bilgiyi [10] da bulabilirsiniz.

4.6 Diğer Paketleri Kurmak

En çok kullanılan paketler \LaTeX ile birlikte otomatik olarak bilgisayarınızda kurulmuş olarak bulunurlar. Fakat internette sunulan pekçok işe yarar paket daha vardır. Değişik stil paketlerinin bulunduğu başlıca adres şudur:

CTAN (<http://www.ctan.org/>).

`geometry`, `hyphenat`, ... gibi diğer tüm paketler tipik olarak iki dosyadan oluşur: biri `.ins` uzantılı, diğeri `.dtx` uzantılı olur. Bazan da paket hakkında bilgi veren `readme.txt` (benioku) dosyayı olur, önce bu dosyayı okumanızda yarar vardır.

Paket dosyalarını sabit diskinizde bir yere kopyaladıktan sonra, yapmanız gereken iki iş vardır: (a) \TeX programına yeni paketin varlığını ve yerini bildirmek, (b) Paketin dokümantasyonunu çıkarmak. Birinci işi şöyle yaparsınız:

1. `.ins` dosyasını \LaTeX ile derleyin. Bu işlem sonunda sabit diskinizde `.sty` uzantılı bir dosya çıkacaktır.
2. `.sty` dosyasını \LaTeX 'in bulabileceği bir yere kopyalayın. Bu, genellikle `.../localtexmf/tex/latex` alt klasöründe bir yer olur (Windows veya OS/2 kullanıcıları bölü işaretini yerine geribölü işaretini kullanırlar.)

3. \LaTeX kurulumunun dosya-adlarını sakladığı veritabanını tazeleyin. Hangi komutu kullanacağımız, \LaTeX dağıtımına göre değişir:
 TeX , fpTeX için `texhash`; web2c için `maktexlsrc`; MikTeX için `initexmf -update-fndb`, veya grafik arabirimini kullanın.

Sonra, `.dtx` dosyasından dokümantasyonu şöyle çıkarırsınız:

1. `.dtx` dosyasını \LaTeX ile derleyin. Bu işlem sonucu sabit diskinizde `.dvi` dosyası oluşacaktır. Atıfları da dizmek için \LaTeX 'den birkaç kez geçirmeniz gerekir.
2. Derleme sonucu `.idx` uzantılı bir dosya çıkıp çıkmadığını kontrol edin. Bu dosya gözüküyorsa, dizin oluşturmaya gerek yok demektir ve doğrudan 5. adıma geçebilirsiniz.
3. Dizin oluşturmak için, şu komutu girin:

```
makeindex -s gind.ist dosya
```

(burada `dosya` girdi dosyanızın adıdır, uzantısı gerekmez).
4. `.dtx` dosyasını bir kez daha \LaTeX 'den geçirin, `.dvi` uzantılı dokümantasyon dosyası oluşur.
5. Son olarak, okuma zevkinizi artırmak istiyorsanız, `dvips` veya `pdflatex` programlarıyla, dokümantasyonu `.ps` veya `.pdf` formatında çıkarabilirsiniz.

Bazen bu derlemeler sonucu `.glo` uzantılı bir dosya oluştuğunu görebilirsiniz. Terimler listesi (glossary) çıkarmaya yarayan bu dosya için, 4 ile 5 adımları arasında şu komutu girin:

```
makeindex -s gglo.ist -o dosya.gls dosya.glo
```

Sonra, `.dtx` dosyasını bir kez daha \LaTeX ile derleyip, 5. adıma geçebilirsiniz.

4.7 PDF Formatında Doküman Çıkarmak

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

PDF bir hiper-metin doküman formatıdır. Bir web sayfasındaki gibi, altı çizili görünen bazı kelimeler hiper-bağlantı olarak işaretlenmişlerdir, o dokümanda ve hatta başka bir dokümandaki bir yere bağlantıları vardır. Böyle bir hiper-bağlantı üzerini tıklarsanız, sizi bağlantının varacağı yere götürür. \LaTeX bağlamında konuşacak olursak, `\ref` ve `\pageref` komutlarının geçtiği her yer hiper-bağlantı demektir. Keza, İçindekiler tablosu, Dizin gibi benzer yapılar da birer hiper-bağlantı yerleridir.

Günümüzde web sayfaları çoğunlukla HTML (*HyperText Markup Language*) dilinde yazılmışlardır. Bilimsel doküman yazımında bu formatın iki önemli sorunu vardır:

1. HTML dokümanlarında matematik formülleri yazmak zordur. Bu iş için bir standardın var olmasına rağmen, çoğu gezdirici programlar bunu ya desteklemezler, yahut da gerekli yazıtiplerini bulundurmazlar.
2. HTML dokümanlarını yazıcıda bastırduğunuzda, sonuçlar her sisteme veya gezdiriciye göre değişir. Bunların kalitesi, \LaTeX ortamında alıştığımız düzeyin çok altındadır.

\LaTeX dokümanlarını HTML formatına çevirmek üzere pekçok kişi program yazmaya girişti. Bazıları \LaTeX girdi dosyasından internet ortamında okunabilecek kalitede HTML dosyası yapmayı başardılar. Fakat, hepsi de kuralları kısıyından köşesinden çiğneyerek bunu yapabildiler. Nitekim, biraz karmaşık \LaTeX komutları veya paketler işin içine girdiğinde, hepsinde sorunlar çıktı. Dokümanlarının dizgi kalitesini internet ortamında korumak isteyen yazarlar, sonunda PDF (*Portable Document Format*) formatında karar kıldılar. Bu formatta hem doküman dizgisi korunabiliyor, hem de hipermetin bağlantılarında gezinebiliyordu. Günümüzdeki gezdirici programların tümünde, PDF dokümanlarını doğrudan gösterebilen ilave araçlar (plug-in) bulunmaktadır.

Her işletim sisteminde DVI ve PS formatındaki dokümanları önizleme programları olduğu gibi, PDF dokümanlarını görebilmek için Acrobat Reader ve Xpdf gibi programlar da yaygın olarak kullanılmaktadır. Bu yüzden, dokümanınızın PDF formatında bir kopyasını çıkarabilerseniz, daha geniş bir okuyucu kitlesine hitap etmiş olursunuz.

Bilgisayarınızdaki \LaTeX kurulumuyla gelen bazı programlar kullanarak, PDF formatında doküman elde etmek için 3 farklı yol izleyebilirsiniz:

1. **pdf \LaTeX** programı girdi dosyanızdan doğrudan PDF dokümanı çıkarır. Bunun için, girdi dosyanızı \LaTeX ile derlemek yerine,

```
pdflatex dosyaadi.tex
```

komutuyla pdf \LaTeX 'ten geçirmeniz yeterlidir. Bu işlem sırasında, dokümanınızdaki hipermetin bağlantıları da PDF dokümanına aynen aktarılır. Klavyeden komut girmeyi kaldıran bazı grafik arabirimlerinde bu iş için özel bir tuş bulabilirsiniz.

Bu, işin kolay tarafı. Şimdi zor tarafına gelelim: POSTSCRIPT formatındaki grafikleri Pdf \LaTeX kabul etmez. Sadece .png, .pdf, .jpg ve .mps formatında grafikleri alabilir. Bu sorunu aşmanın basit yolu, **epstopdf** programını kullanarak EPS grafik dosyalarınızı PDF formatına çevirmektir. Bu durumda, sabit diskinizde aynı grafiği iki farklı formatta saklamanız gerekir.

2. **DVIPS** ve **EPSTOPDF** programlarını birlikte kullanarak PDF dokümanı çıkarabilirsiniz. Bunun için, önce .dvi uzantılı dosyanızdan

DVIPS programıyla bir `.ps` dosyası elde etmek üzere şu komutu girersiniz:

```
dvips -o dosyaadı.ps dosyaadı.dvi
```

Burada, `-o` parametresi, yazıcıya göndermeyip sabit diskte `.ps` uzantılı bir dosyaya yazmasını söyler. Sonra, bu `.ps` dosyasını EPSTOPDF programına şu komutla gönderirsiniz:

```
epstopdf dosyaadı.ps
```

ve sabit diskte `.pdf` uzantılı doküman elde etmiş olursunuz.

Bu yöntemin sorunu ise, hiper-metin bağlantılarını yoketmesidir.

3. **DVIPDFM** programı, yukardaki iki yöntemin sorunlarını ortadan kaldıran en elverişli yoldur. Bu program `.dvi` uzantılı dosya üzerinde çalışır. Bu da bir avantajdır, zira elinizde `.tex` uzantılı L^AT_EX girdi dosyası bulunmuyor olabilir, veya başkaları size sadece `.dvi` uzantılı bir dosya göndermiş olabilir. Programı çalıştırmak için, şu komutu girersiniz:

```
dvipdfm dosyaadı.dvi
```

Böylece, sabit diskte PDF dokümanı oluşur. Bu yöntemle, hem POSTSCRIPT formatında grafikleri ve hem de hiper-bağlantıları olduğu gibi korumuş olursunuz.

Bölüm 5

Grafik Çizimleri

Pekçok kişi \LaTeX 'i metin yazmakta kullanır. Oysa, \LaTeX 'in, sınırlı olanaklarla da olsa, verilen komutlarla grafik çizme yeteneği de vardır. Bu yetenek ilave paketlerle daha da genişletilmiştir. Bu bölümde grafik çiziminin temellerini anlatacağız.

5.1 Genel Bakış

\LaTeX içinde grafik çizimi `picture` (resim) ortamında yapılır. Bu ortamın ayrıntılı bir açıklaması *\LaTeX Manual* [1]de bulunur. Bu ortamda, bir yandan doğru parçalarının eğimi, çemberlerin yarıçapları, ... için katı kısıtlamalar gelir. Diğer yandan da, `picture` ortamıyla birlikte yeni komutlar girer (`\put`, `\qbezier`, ...). Çember, elips veya zincir gibi çok sık kullanılan bazı eğriler kuadratik Bézier eğrileriyle yaklaşık olarak çizilebilirler. Buna ek olarak, Java türü bir programlama diliyle `\qbezier` blokları oluşturulursa, `picture` ortamı çok daha güçlü olur.

\LaTeX 'te çizimleri doğrudan programlamak çok kısıtlı ve yorucu olsa da, bunu yapmak için bazı sebepler yine de vardır. Böyle yazılan dokümanlar sabit diskte daha az yer kaplar ve yanlarında diğer grafik dosyaları taşımaya gerek bırakmazlar.

The \LaTeX Companion [3]'da açıklanan `epic`, `eepic` veya `pstricks` gibi paketler `picture` ortamının zorluklarını giderip \LaTeX 'in grafik gücünü artırmayı hedeflemişlerdir. İlk iki paket kolaylık sağlarken, `pstricks` paketinin kendi `pspicture` ortamı vardır. Bu ortam POSTSCRIPT dilinin avantajlarını kullanır. Özel bazı çizimler için de ayrı paketler yazılmıştır. Bunlardan biri, bölüm sonunda anlatacağımız `Xy-pic` paketidir.

Fakat, \LaTeX 'in en güçlü grafik aracı şüphesiz `MetaPost` programıdır. Donald E. Knuth'un `METAFONT` programının ikizi olan bu program, `METAFONT` programlama dilinin tüm inceliklerini kullanarak en zor işleri yapabilir. Sonuçları bitmap olarak üreten `METAFONT`'un tersine, `MetaPost` programı POSTSCRIPT formatında grafik ürettiği için, sonuçlar doğrudan \LaTeX girdi dosyasına

konulabilir. Giriş düzeyinde MetaPost öğreten *A User's Manual for MetaPost* [15] elkitabına, veya [17] no.lu kaynağa bakabilirsiniz.

L^AT_EX ve T_EX için grafik (ve yazıtipi) stratejilerinin ayrıntılı bir öğretisini *T_EX Unbound* [16] kitabından okuyabilirsiniz.

5.2 picture (resim) Ortamı

By Urs Oswald <osurs@bluewin.ch>

5.2.1 Temel Komutlar

İnanması zor ama, `picture` ortamı, başka hiçbir paket veya program gerektirmeden, doğrudan L^AT_EX'in içinden çalışır. Bu ortam şu iki komuttan biriyle açılır:

```
\begin{picture}(x,y)...\end{picture}
```

veya

```
\begin{picture}(x,y)(x_0,y_0)...\end{picture}
```

Burada x , y , x_0 , y_0 sayıları `\unitlength` (uzunluk birimi) cinsinden verilmiş değerlerdir. Bu birim, `picture` ortamı dışında olmak kaydıyla, her an şöyle bir komutla değiştirilebilir:

```
\setlength{\unitlength}{1.2cm}
```

Varsayılan `\unitlength` birim değeri `1pt` dir. Ortamdaki ilk sayı çifti, (x, y) , resim için ayrılan dikdörtgen bölgenin boyutlarıdır. Opsiyonel ikinci çift, (x_0, y_0) , ayrılan bölgenin sol alt köşesinin koordinatlarının ne olacağını bildirir.

Çizim komutları genellikle şu iki yapıdan birinde olurlar:

```
\put(x,y){nesne}
```

veya

```
\multiput(x,y)(\Delta x,\Delta y){n}{nesne}
```

Fakat Bézier eğrileri farklıdır. Onlar şu komutla çizilirler:

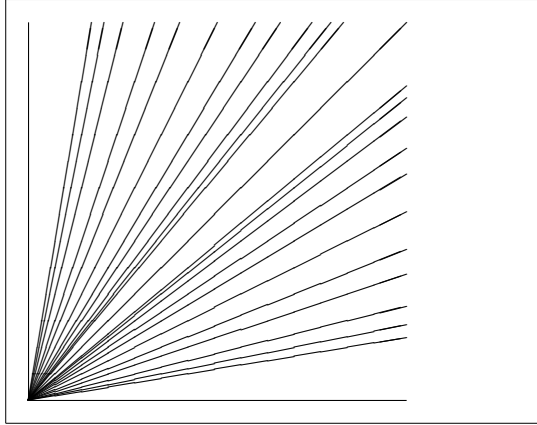
```
\qbezier(x_1,y_1)(x_2,y_2)(x_3,y_3)
```


5.2.2 Doğru Parçaları

```

\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(0,1){1}}
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(1,1){1}}
  \put(0,0){\line(1,2){.5}}
  \put(0,0){\line(1,3){.3333}}
  \put(0,0){\line(1,4){.25}}
  \put(0,0){\line(1,5){.2}}
  \put(0,0){\line(1,6){.1667}}
  \put(0,0){\line(2,1){1}}
  \put(0,0){\line(2,3){.6667}}
  \put(0,0){\line(2,5){.4}}
  \put(0,0){\line(3,1){1}}
  \put(0,0){\line(3,2){1}}
  \put(0,0){\line(3,4){.75}}
  \put(0,0){\line(3,5){.6}}
  \put(0,0){\line(4,1){1}}
  \put(0,0){\line(4,3){1}}
  \put(0,0){\line(4,5){.8}}
  \put(0,0){\line(5,1){1}}
  \put(0,0){\line(5,2){1}}
  \put(0,0){\line(5,3){1}}
  \put(0,0){\line(5,4){1}}
  \put(0,0){\line(5,6){.8333}}
  \put(0,0){\line(6,1){1}}
  \put(0,0){\line(6,5){1}}
\end{picture}

```



Bir doğru parçası şu komutla çizdirilir:

```
\put(x,y){\line(x1,y1){uzunluk}}
```

`\line` (çizgi) komutu iki argüman alır:

1. bir doğrultman vektörü,
2. bir uzunluk.

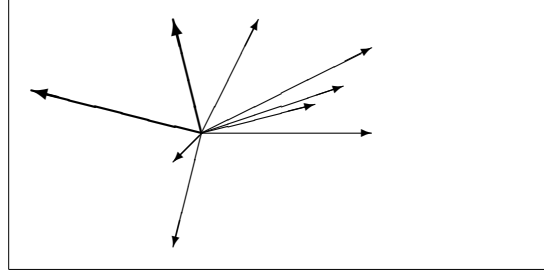
Doğrultman vektörünün (x_1, y_1) bileşenleri tamsayı ve aralarında asal olmak zorundadırlar, yani ortak bölenleri olmamalıdır [örneğin, $(3,5)$, $(2,7)$ gibi]. Yukardaki şekilde düzlemin birinci çeyreğinde farklı eğimde çizilebilecek tüm doğrular görülmektedir. Bunların uzunluğu `\unitlength` birimi cinsindedir (bu örnekte 5 cm). Düşey doğru parçası için uzunluk argümanı düşey koordinat kadar, diğerleri için yatay koordinat olarak alınmıştır.

5.2.3 Oklar

```

\setlength{\unitlength}{0.75mm}
\begin{picture}(60,40)
  \put(30,20){\vector(1,0){30}}
  \put(30,20){\vector(4,1){20}}
  \put(30,20){\vector(3,1){25}}
  \put(30,20){\vector(2,1){30}}
  \put(30,20){\vector(1,2){10}}
  \thicklines
  \put(30,20){\vector(-4,1){30}}
  \put(30,20){\vector(-1,4){5}}
  \thinlines
  \put(30,20){\vector(-1,-1){5}}
  \put(30,20){\vector(-1,-4){5}}
\end{picture}

```



Oklar şu komutla çizilir:

```
\put(x,y){\vector(x1,y1){uzunluk}}
```

Okların doğrultman vektörleri çok daha kısıtlı olup, sadece şu tamsayı-
lardan ikisi, aralarında asal olacak şekilde seçilebilir:

$$-4, -3, \dots, 3, 4.$$

Şekilde, üst sol tarafa yönelik ok için kullanılan `\thicklines` (kalın çizgi) komutunun etkisine dikkat edin.

5.2.4 Çemberler

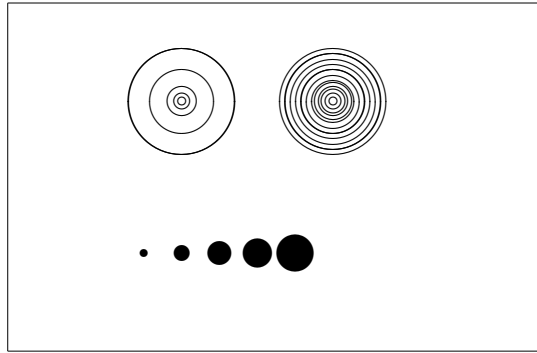
```

\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
  \put(20,30){\circle{1}}
  \put(20,30){\circle{2}}
  \put(20,30){\circle{4}}
  \put(20,30){\circle{8}}
  \put(20,30){\circle{16}}
  \put(20,30){\circle{32}}

  \put(40,30){\circle{1}}
  \put(40,30){\circle{2}}
  \put(40,30){\circle{3}}
  \put(40,30){\circle{4}}
  \put(40,30){\circle{5}}
  \put(40,30){\circle{6}}
  \put(40,30){\circle{7}}
  \put(40,30){\circle{8}}
  \put(40,30){\circle{9}}
  \put(40,30){\circle{10}}
  \put(40,30){\circle{11}}
  \put(40,30){\circle{12}}
  \put(40,30){\circle{13}}
  \put(40,30){\circle{14}}

  \put(15,10){\circle*{1}}
  \put(20,10){\circle*{2}}
  \put(25,10){\circle*{3}}
  \put(30,10){\circle*{4}}
  \put(35,10){\circle*{5}}
\end{picture}

```



Çember çizdiren komut

```
\put(x,y){\circle{çap}}
```

olup, merkezi (x, y) noktasıdır ve çapı (yarıçapı değil) verilir. `picture` ortamında çapı en fazla 14 mm olan çemberler çizilebilir, ayrıca bu değer altındaki her çap çizilemeyebilir. Çemberin içini doldurmak isterseniz, yıldızlı `\circle*` komutu kullanırsınız.

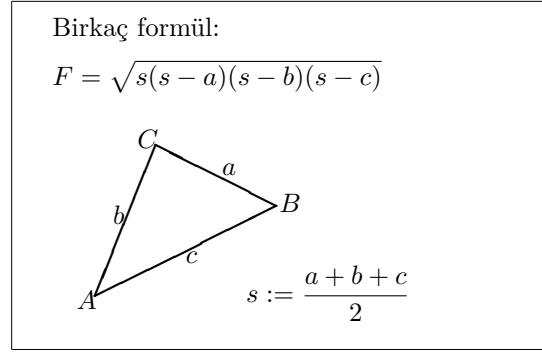
Doğru parçalarında olduğu gibi, daha iyi çemberler için `eepic` veya `pstricks` gibi ilave paketlere ihtiyaç duyabilirsiniz. Bunların açıklaması *The L^AT_EX Graphics Companion* [4]'da vardır.

`picture` ortamında bir seçenek daha vardır. Eğer gerekli hesapları yapmaktan (veya, onları bir programa yaptırmaktan) kaçmıyorsanız, istediğiniz boyda çember ve elipsleri kuadratik Bézier ağrilerini birleştirerek çizdirebi-

lirsiniz. Bunun örnekleri ve Java programları için *Graphics in L^AT_EX 2_ε* [17] e bakın.

5.2.5 Metin ve Formüller

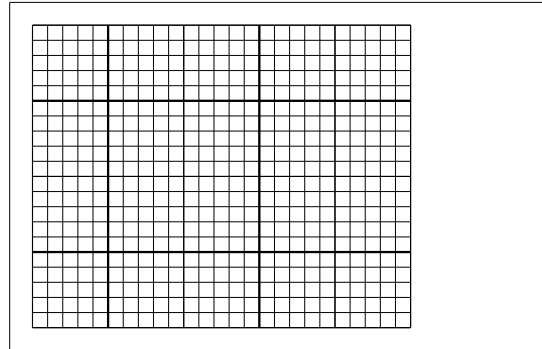
```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
  \thicklines
  \put(0.3,4.8){Birkaç formül:}
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.7,0.3){\mathit{A}}
  \put(4.05,1.9){\mathit{B}}
  \put(1.7,2.95){\mathit{C}}
  \put(3.1,2.5){\mathit{a}}
  \put(1.3,1.7){\mathit{b}}
  \put(2.5,1.05){\mathit{c}}
  \put(0.3,4){\mathit{F}=}
  \put(3.5,0.4){\mathit{s}:=\frac{a+b+c}{2}}
  \put(3.5,0.4){\mathit{s}:=\frac{a+b+c}{2}}
\end{picture}
```



picture ortamına yazı yazdırmak veya formül girmek için, yine \put komutu kullanırsınız. Sanırım yukardaki örnek gayet açıktır.

5.2.6 \multiput ve \linethickness Komutları

```
\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){26}%
  {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
  {\line(1,0){25}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){6}%
  {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
  {\line(1,0){25}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){2}%
  {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
  {\line(1,0){25}}
\end{picture}
```



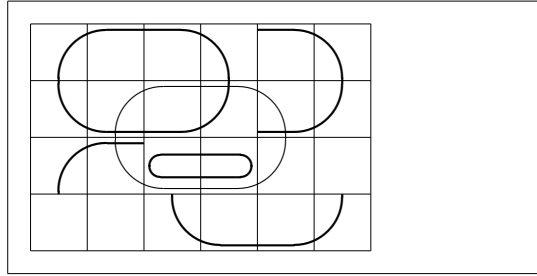
`\multiput` komutunun 4 argümanı vardır:

```
\multiput(x,y)(\Delta x,\Delta y){n}{nesne}
```

Bunlar sırasıyla, başlangıç noktası, bir nesneden diğerine öteleme vektörü, nesne sayısı ve çizilecek nesnenin cinsi olurlar. `\linethickness` (çizgi kalınlığı) komutu yatay ve dikey doğru parçalarına uygulanır, ama eğimli çizgilere veya çemberlere uygulanmaz. (Fakat nedense, kuadratik Bézier eğrilerine uygulanırlar.) Yukardaki örnekte nasıl milimetrik kağıt yapabileceğiniz görülmektedir.

5.2.7 Ovaler

```
\setlength{\unitlength}{0.75cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}%
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}%
    {\line(1,0){6}}
  \thicklines
  \put(2,3){\oval(3,1.8)}
  \thinlines
  \put(3,2){\oval(3,1.8)}
  \thicklines
  \put(2,1){\oval(3,1.8)[t1]}
  \put(4,1){\oval(3,1.8)[b]}
  \put(4,3){\oval(3,1.8)[r]}
  \put(3,1.5){\oval(1.8,0.4)}
\end{picture}
```



Oval çizmek için,

```
\put(x,y){\oval(w,h)}
```

veya

```
\put(x,y){\oval(w,h)[konum]}
```

komutları kullanılır. Merkezi (x, y) , genişliği w ve yüksekliği h kadar olur. Opsiyonel *konum* argümanı için **b** (alt), **t** (üst), **l** (sol) ve **r** (sağ) seçenekleri birlikte kullanılabilirler. Yukardaki örnekte olduğu gibi.

Çizgi kalınlığı iki tür komutla değiştirilebilir:

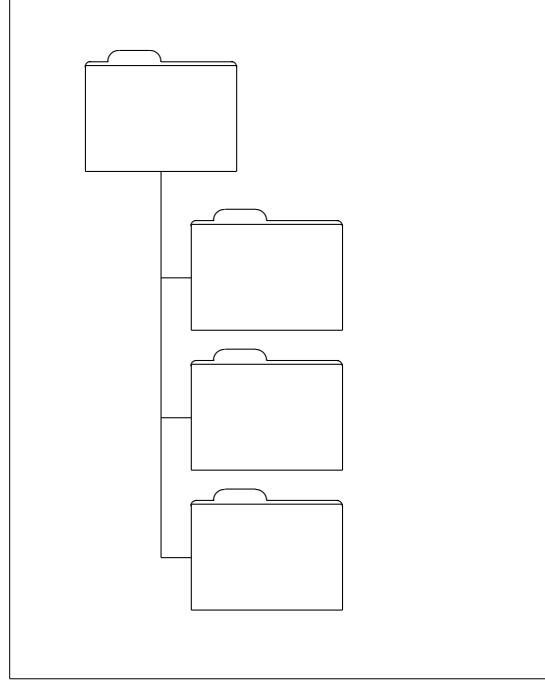
`\linethickness{kalınlık}`, veya `\thinlines` (ince çizgi) ve `\thicklines` (kalın çizgi) ikilisiyle. `\linethickness{kalınlık}` komutu sadece yatay ve dikey çizgilere (ve Bézier eğrilerine) uygulanır. `\thinlines` ve `\thicklines` ikilisi eğik çizgiler, çember ve ovaler için de geçerlidirler.

5.2.8 Önceden Tanımlanmış Resim Kutuları Kullanma

```

\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\aklasor}
\savebox{\aklasor}
  (40,32)[bl]{% Tanım
  \multiput(0,0)(0,28){2}
    {\line(1,0){40}}
  \multiput(0,0)(40,0){2}
    {\line(0,1){28}}
  \put(1,28){\oval(2,2)[t1]}
  \put(1,29){\line(1,0){5}}
  \put(9,29){\oval(6,6)[t1]}
  \put(9,32){\line(1,0){8}}
  \put(17,29){\oval(6,6)[tr]}
  \put(20,29){\line(1,0){19}}
  \put(39,28){\oval(2,2)[tr]}
  }
\newsavebox{\bklasor}
\savebox{\bklasor}
  (40,32)[l]{% Tanım
  \put(0,14){\line(1,0){8}}
  \put(8,0){\usebox{\aklasor}}
  }
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\aklasor}}
\multiput(34,86)(0,-37){3}
  {\usebox{\bklasor}}
\end{picture}

```



Bir resim kutusu şu komutla *bildirilir*:

```
\newsavebox{isim}
```

ve sonra, şu komutla *tanımlanır*:

```
\savebox{isim}(genişlik,yükseklik)[konum]{içerik}
```

Daha sonra *çizdirmek* istendiğinde, şu komut verilir:

```
\put(x,y)\usebox{isim}
```

Opsiyonel *konum* parametresi `\savebox` komutunun demir atma noktasını belirler. Yukardaki örnekte, `bl` alınmış, yani `savebox`'un sol alt köşesi seçilmiştir. Diğer seçenekler, `t` (üst) ve `r`(sağ) olurlar.

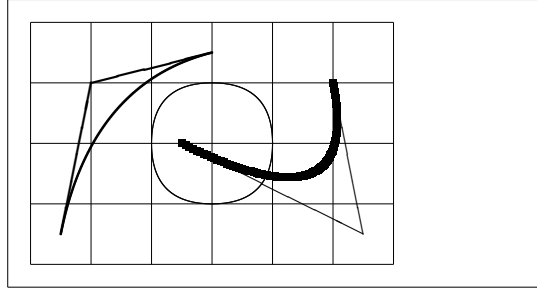
isim argümanı \LaTeX 'in saklama kovanına aktarıldığı için, bir komut gibi geribölü işaretiyle verilir (örnekte, `\aklasor` gibi). Kutulanmış resimler iç

içe yuvalanabilirler. Bu örnekte, `\bklassor` tanımı içinde `\aklassor` kullanılabilmiştir.

Çizimde `\line` yerine `\oval` komutu kullanılmıştır, çünkü `\line` komutu 3 mm den küçük çizgi çizemez.

5.2.9 Kuadratik Bézier Eğrileri

```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}
    {\line(1,0){6}}
  \thicklines
  \put(0.5,0.5){\line(1,5){0.5}}
  \put(1,3){\line(4,1){2}}
  \qBezier(0.5,0.5)(1,3)(3,3.5)
  \thinlines
  \put(2.5,2){\line(2,-1){3}}
  \put(5.5,0.5){\line(-1,5){0.5}}
  \linethickness{1mm}
  \qBezier(2.5,2)(5.5,0.5)(5,3)
  \thinlines
  \qBezier(4,2)(4,3)(3,3)
  \qBezier(3,3)(2,3)(2,2)
  \qBezier(2,2)(2,1)(3,1)
  \qBezier(3,1)(4,1)(4,2)
\end{picture}
```



Bu örnekte görüldüğü üzere, bir çemberi 4 tane kuadratik Bézier eğrisiyle oluşturmak iyi sonuç vermez. Aynı şekilde, `\linethickness` (çizgi kalınlığı) komutunun eğik çizgiler üzerindeki etkisi de görülmektedir. Ayrıca, her iki tür komutun da Bézier eğrilerinde kullanılabildiği, her komutun kendinden öncekilerin yerini aldığı görülmektedir.

Uç noktaları $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ ve eğimleri m_1, m_2 olan iki Bézier eğrisi olsun. Ara kontrol noktasının $S = (x, y)$ koordinatları şu denklemin çözümleri olurlar:

$$\begin{cases} x = \frac{m_2 x_2 - m_1 x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y = y_i + m_i(x - x_i) \quad (i = 1, 2). \end{cases} \quad (5.1)$$

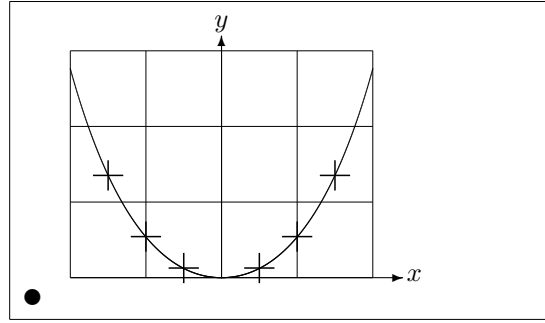
Gerekli `\qBezier` komut satırını oluşturan Java programını, *Graphics in L^AT_EX 2_ε* [17] kitabında bulabilirsiniz.

5.2.10 Zincir Eğrisi

```

\setlength{\unitlength}{1cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){\mathit{x}}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){\mathit{y}}}
\qBezier(0.0,0.0)(1.2384,0.0)
(2.0,2.7622)
\qBezier(0.0,0.0)(-1.2384,0.0)
(-2.0,2.7622)
\linethickness{.075mm}
\multiput(-2,0)(1,0){5}
{\line(0,1){3}}
\multiput(-2,0)(0,1){4}
{\line(1,0){4}}
\linethickness{.2mm}
\put(.3,.12763){\line(1,0){.4}}
\put(.5,-.07237){\line(0,1){.4}}
\put(-.7,.12763){\line(1,0){.4}}
\put(-.5,-.07237){\line(0,1){.4}}
\put(.8,.54308){\line(1,0){.4}}
\put(1,.34308){\line(0,1){.4}}
\put(-1.2,.54308){\line(1,0){.4}}
\put(-1,.34308){\line(0,1){.4}}
\put(1.3,1.35241){\line(1,0){.4}}
\put(1.5,1.15241){\line(0,1){.4}}
\put(-1.7,1.35241){\line(1,0){.4}}
\put(-1.5,1.15241){\line(0,1){.4}}
\put(-2.5,-0.25){\circle*{0.2}}
\end{picture}

```



Bu şekilde, $y = \cosh x - 1$ zincir eğrisinin iki simetrik yarısının her biri kuadratik Bézier eğrisiyle oluşturulmuştur. Eğrinin sağ tarafının bittiği $(2, 2.7622)$ noktasındaki eğimi $m = 3.6269$ dir. Yine, (5.1) denklemini kullanarak, ara kontrol noktalarını hesaplayabiliriz. Hesap sonuçları $(1.2384, 0)$ ve $(-1.2384, 0)$ verir. Çarpı işaretleri *gerçek* zincir eğrisinin noktaları olup, hata payları yüzde birin altında olup, gözle farkedilemeyecek kadar küçüktür.

Bu örnek `\begin{picture}` komutunun opsiyonel argümanının nasıl kullanılacağını da gösteriyor. Çizim bildiğimiz “matematik” koordinatla yapıyor, ama komut

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

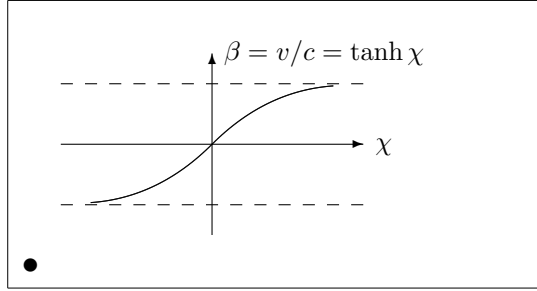
Siyah daire ile gösterilen sol köşeyi $(-2.5, -0.25)$ koordinatları ile tanımlıyor.

5.2.11 Özel Görelilik Teorisinde Hızlılık

```

\setlength{\unitlength}{0.8cm}
\begin{picture}(6,4)(-3,-2)
  \put(-2.5,0){\vector(1,0){5}}
  \put(2.7,-0.1){$\chi$}
  \put(0,-1.5){\vector(0,1){3}}
  \multiput(-2.5,1)(0.4,0){13}
    {\line(1,0){0.2}}
  \multiput(-2.5,-1)(0.4,0){13}
    {\line(1,0){0.2}}
  \put(0.2,1.4)
    {${\beta=v/c=\tanh\chi}$}
  \qbezier(0,0)(0.8853,0.8853)
    (2,0.9640)
  \qbezier(0,0)(-0.8853,-0.8853)
    (-2,-0.9640)
  \put(-3,-2){\circle*{0.2}}
\end{picture}

```



Her iki Bézier eğrisinin kontrol noktaları (5.1) denkleminde hesaplanır. Pozitif dal $P_1 = (0, 0)$, $m_1 = 1$ ve $P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$ noktaları arasındadır. Yine, resim matematik koordinatlarıyla tanımlanır ve sol alt köşeye (siyah daire) $(-3, -2)$ koordinatları verilir.

5.3 Xy-pic Paketi

By Alberto Manuel Brandão Simões <albie@alfarrabio.di.uminho.pt>

Xy-pic diyagram çizimlerinde kullanılan özel bir pakettir. Onu kullanabilmek için, doküman sahanlığına şu satırı girmeniz gerekir:

```
\usepackage[opsiyonlar]{xy}
```

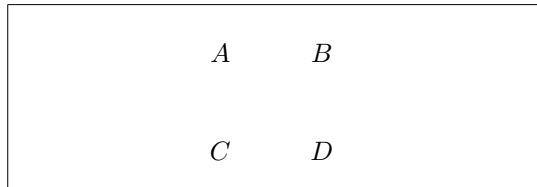
Xy-pic'in yüklemek istediğiniz fonksiyonları *opsiyonlar* içinde yer alırlar. Size `all` (hepsi) opsiyonuyla, tüm Xy komutlarını yüklemenizi öneririm.

Xy-pic diyagramları bir matris düzenindeki zemin kullanırlar ve her matris elemanı yerine bir diyagram elemanı konur.

```

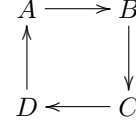
\begin{displaymath}
\xymatrix{A & B \\
          C & D }
\end{displaymath}

```



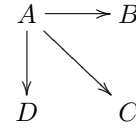
Matematik kipinde `\xymatrix` komutu kullanılmalıdır. Bu örnekte iki satır ve iki sütun kullandık. Bu matrisi diyagrama dönüştürmek için, `\ar` komutuyla oklar eklenmelidir.

```
\begin{displaymath}
\begin{matrix}
A \xrightarrow{\quad} B \\
\uparrow \qquad \downarrow \\
D \xleftarrow{\quad} C
\end{matrix}
\end{displaymath}
```



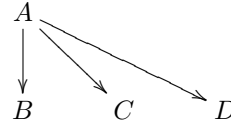
\ar (ok) komutu başladığı hücrede verilir, argümanı hangi yöne olacağını belirtir: u (yukarı), d (aşağı), r (sağa) ve l (sola).

```
\begin{displaymath}
\begin{matrix}
A \xrightarrow{\quad} B \\
\downarrow \searrow \\
D \qquad \qquad C
\end{matrix}
\end{displaymath}
```



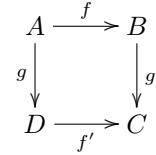
Çapraz oklar için birden fazla yön verilir. Daha büyük oklar için, yön komutlarını tekrar edersiniz.

```
\begin{displaymath}
\begin{matrix}
A \xrightarrow{\quad} B \\
\downarrow \searrow \swarrow \\
B \qquad \qquad C \qquad D
\end{matrix}
\end{displaymath}
```



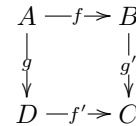
Okları harflendirmek için, bildiğimiz üs ve indis işlemcilerini kullanırız.

```
\begin{displaymath}
\begin{matrix}
A \xrightarrow{f} B \\
g \downarrow \qquad \downarrow g' \\
D \xrightarrow{f'} C
\end{matrix}
\end{displaymath}
```



Dikkat edilirse, matematik kipinde olmasına rağmen, üs işareti ‘ok üzerinde’ ve indis işareti ‘okun altında’ anlamına gelir. Üçüncü bir işlemci dikey çubuktur: | Harfi okun içine yazar.

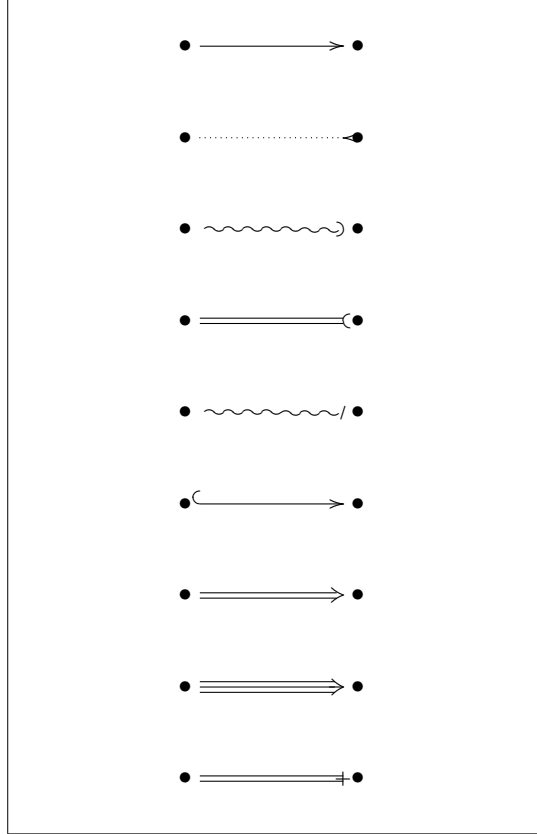
```
\begin{displaymath}
\begin{matrix}
A \xrightarrow{f} B \\
\downarrow |g \qquad \downarrow |g' \\
D \xrightarrow{f'} C
\end{matrix}
\end{displaymath}
```



İçinde delik olan bir ok çizmek için \ar[...]|hole komutu kullanılır.

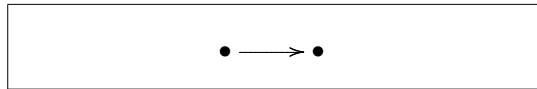
Bazı durumlarda değişik ok türlerini ayırdetmek gerekir. Bu işi, üzerlerine harfler koyarak yaptığımız gibi, okların görünümünü değiştirerek de yapabilirsiniz:

```
\shorthandoff{"}
\begin{displaymath}
\xymatrix{
\bullet\ar@{->}[rr] && \bullet\\
\bullet\ar@{.<}[rr] && \bullet\\
\bullet\ar@{~}[rr] && \bullet\\
\bullet\ar@{=} [rr] && \bullet\\
\bullet\ar@{~/}[rr] && \bullet\\
\bullet\ar@{^}({->}[rr] && \bullet\\
\bullet\ar@2{->}[rr] && \bullet\\
\bullet\ar@3{->}[rr] && \bullet\\
\bullet\ar@{=+}[rr] && \bullet
}
\end{displaymath}
\shorthandon{"}
```

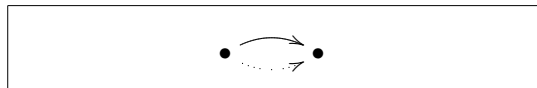


Şu iki diyagram arasındaki farka dikkat edin:

```
\begin{displaymath}
\xymatrix{
\bullet \ar[r]
\ar@{.>}[r] &
\bullet
}
\end{displaymath}
```



```
\begin{displaymath}
\xymatrix{
\bullet \ar/^/[r]
\ar@/_@{.>}[r] &
\bullet
}
\end{displaymath}
```



Bölümler arasındaki işlemciler eğrilerin nasıl çizileceğini belirlemekteler. XY-pic paketinde eğri çizimleri için başka komutlar da vardır. Daha fazla bilgi için, XY-pic dokümanlarına bakın.

Bölüm 6

L^AT_EX'i Özelleştirmek

Buraya kadar öğrendiğiniz komutlarla yazacağınız dokümanlar çok geniş kitlelerin beğenisini kazanır düzeyde olacaklardır. Fazla albenili olmasalar da, rahat okunabilme ve göze hoş gelme bakımından, iyi dizilmiş bir dokümanın kurallarına uygun olurlar.

Fakat, L^AT_EX'in hazır şablonları sizin isteklerinize her zaman cevap veremeyebilir, veya bazı komutlar sizin düşüncenizi tam uygulamayabilirler.

Bu bölümde L^AT_EX'e yeni marifetler öğretebilmenizin yollarını anlatacağız. Böylece, varsayılan formatlardan daha farklı dizilmiş dokümanlar yazabileceksiniz.

6.1 Yeni Komutlar, Ortamlar ve Paketler

Dikkat ettiyseniz, bu kitapta açıkladığım tüm komutlar bir kutu içinde yazılmış ve kitabın sonundaki Dizin içinde gösterilmişlerdir. Bunu yapabilmek için, L^AT_EX'in komutlarını doğrudan kullanmak yerine, kendim yeni komutlar ve ortamlar tanımladım, sonra da bunları bir paket içinde topladım. Böylece, artık bazı şeyleri kısaca yazabiliyorum. Örneğin:

```
\begin{lscommand}  
\ci{dum}  
\end{lscommand}
```



\dum

Bu örnekte `lscommand` adında yeni bir ortamda çalışıyorum, bu ortamda yazdığım herşey kutu içine alınıyor. Ayrıca, `\ci` adında yeni bir komut kullanıyorum, bu komut içine yazdığım şeyi hem bir komut gibi geribölü işaretiyle diziyo, hem de onu kitabın sonundaki Dizin içine atıyor. Dizine baktığınızda `\dum` için bir madde girildiğini ve onun geçtiği sayfaları gösterdiğini görebilirsiniz.

Eğer günün birinde komutları kutu içinde yazmaktan vazgeçersem, sadece `lscommand` ortamının tanımını değiştirir ve yeni bir görünüm yaparım.

Bu iş, doküman içinde kutuya konmuş komutları bulup tek tek kaldırmaktan çok daha kolaydır.

6.1.1 Yeni Bir Komut Tanımlamak

Kendiniz yeni bir komut yazmak isterseniz, dokümanın sahanlık kısmında şöyle tanımlarsınız:

```
\newcommand{ad}[sayı]{tanım}
```

Bir komutta mutlaka olması gereken iki argüman vardır: Komutun *adı* ve *tanımı*. Köşeli parantez içindeki opsiyonel *sayı* parametresi yeni komutun alabileceği argüman sayısını bildirir. Her komut en fazla 9 argüman alabilir, hiç sayı girilmemişse 0 demektir, yani alacağı argüman yoktur.

Aşağıdaki iki örnek kafanızda bir fikir oluşmasına yardımcı olacaktır. Birinci örnekte, `\iblk` adlı yeni bir komut tanımlanıyor, bu komutu kullandığımızda “İnce Bir L^AT_EX 2_ε Elkitabı” kelimelerini diziyor. Böyle bir komut sizi, kitabın adını her defasında uzun uzun yazmaktan kurtarabilir.

```
\newcommand{\iblk}{İnce Bir
\LaTeXe{}} Elkitabı}
% Doküman içinde kullanılışı:
Kitabın adı ‘‘\iblk’’ \dots{}}
evet, ‘‘\iblk’’
```

Kitabın adı “İnce Bir L^AT_EX 2_ε Elkitabı”
... evet, “İnce Bir L^AT_EX 2_ε Elkitabı”

Diğer örnek 1 argüman alan yeni bir komut tanımlıyor. Tanım içindeki #1 işaretçisinin bulunduğu yere, sizin kullanırken vereceğiniz argüman giriyor. Eğer iki argümanlı bir komut tanımlamak isterseniz, #2 kullanırsınız.

```
\newcommand{\txsit}[1]
{\emph{#1} Bir \LaTeX{}} Elkitabı}
% Doküman içinde kullanılışı:
\begin{itemize}
\item Bu \txsit{İnce}'dir.
\item Bu \txsit{Kalın}'dir.
\end{itemize}
```

- Bu *İnce* Bir L^AT_EX Elkitabı'dır.
- Bu *Kalın* Bir L^AT_EX Elkitabı'dır.

L^AT_EX kendi komutlarından biriyle aynı adı taşıyan yeni bir komut yazmanıza izin vermez. Fakat, mevcut bir komutun yaptığı işi değiştirmek istiyorsanız, bunu `\renewcommand` (yenile) komutuyla yapabilirsiniz. Onun da imla kuralı aynıdır.

Bazı durumlarda `\providecommand` komutu kullanılabilir. Bu da `\newcommand` gibi işlev görür, fakat komut adı önceden tanımlanmış ise L^AT_EX sesini çıkarmadan onu gözardı eder.

L^AT_EX komutlarında boş yerler bırakılmasıyla ilgili önemli bazı noktalar vardır. Bu konuda daha fazla bilgi için Sayfa 5 e bakın.

6.1.2 Yeni Ortam Oluşturmak

Yeni komut için nasıl bir `\newcommand` komutu varsa, yeni bir ortam oluşturmak için de `\newenvironment` (yeni ortam) komutu vardır:

```
\newenvironment{ad}[sayı]{öncesi}{sonrası}
```

Yine, `\newenvironment` komutu da opsiyonel bir argüman alabilir. *öncesi* içine yazılanlar, ortama geçmeden önce işlenirler, *sonrası* içine yazılanlar da ortamdan çıkıldıktan sonra. Aşağıdaki örnekte `\newenvironment` komutunun nasıl kullanılacağını görebilirsiniz.

```
\newenvironment{kral}
  {\rule{1ex}{1ex}%
   \hspace{\stretch{1}}}
  {\hspace{\stretch{1}}%
   \rule{1ex}{1ex}}
```

■ Aziz vatandaşlarım ... ■

```
\begin{kral}
Aziz vatandaşlarım \ldots
\end{kral}
```

Opsiyonel *sayı* argümanı `\newcommand` komutundaki gibi işlev görür. \LaTeX kendi mevcut ortamlarıyla aynı adı taşıyan ortam yapmanıza izin vermez. İlla ki onun ortamlarından birinde değişiklik yapmak isterseniz, bunun için `\renewenvironment` (ortam yenile) komutunu kullanabilirsiniz. Onun da kullanımı `\newenvironment` gibidir.

Bu örnekteki anlamadığımız komutları daha sonra açıklayacağız. `\rule` (çizgi) komutu için Sayfa 102 ye, `\stretch` (uzat) komutu için Sayfa 95 ya ve `\hspace` (yatay boşluk) komutu için Sayfa 95 ya bakabilirsiniz.

6.1.3 Fazladan Boşluklar

Yeni bir ortam oluştururken istemediğiniz boşluklar kolayca oluşur ve bunlar derleme sırasında hatalara yol açabilirler. Örneğin, kendisinin ve kendinden sonraki paragrafın içerden başlamasını engelleyen bir ortam yapmak isterseniz, `\ignorespaces` (boşlukları geç) komutu kullanabilirsiniz. Bu komutu ortamın başında kullandığınızda gerçekten de o paragrafın başındaki boşluğu kaldırır. Fakat, sonunda kullandığınızda, \LaTeX 'in ortamı sonlandırmak için kullandığı iç komutlarla çelişir ve sonraki paragrafın boşlukları kalır. Bunun yerine, `\ignorespacesafterend` komutu kullanılması gerekir. Aşağıdaki iki örnek bu sorunu daha iyi anlatacaktır:

```
\newenvironment{basit}%
{\noindent}%
{\par\noindent}

\begin{basit}
Soldaki boşluğa\dikkat edin.
\end{basit}
Burada da\aynısı.
```

Soldaki boşluğa
dikkat edin.

Burada da
aynısı.

```
\newenvironment{dogrusu}%
{\noindent\ignorespaces}%
{\par\noindent%
\ignorespacesafterend}

\begin{dogrusu}
Solda boşluk\kalmadı.
\end{dogrusu}
Burada da\aynısı.
```

Solda boşluk
kalmadı.

Burada da
aynısı.

6.1.4 Kendi Paketinizi Oluşturmak

Böyle bir sürü yeni komutlar ve ortamlar yazıyorsanız, dokümanınızın sahanlık kısmı giderek fazla uzamaya başlar. Bu durumda, tüm yeni ortam ve komutlarınızı bir L^AT_EX paketi haline getirebilirsiniz. Sonra, sadece `\usepackage` komutuyla onları dokümanınızda kullanır hale gelirsiniz.

```
% Tobias Oetiker'in demo paketi
\ProvidesPackage{demopak}
\newcommand{\iblk}{İnce Bir \LaTeXe{} Elkitabı}
\newcommand{\txsit}[1] {\emph{#1} Bir \LaTeX{} Elkitabı}
\newenvironment{kral}{\begin{quote}}{\end{quote}}
```

Şekil 6.1: Bir Paket Örneği.

Bir paket yazmak, aslında dokümanınızın sahanlığındaki yeni komut ve ortamları ayrı bir dosyaya aktarıp, bunu `.sty` uzantılı bir dosya halinde saklamaktan ibarettir. Paketin başındaki özel komut

```
\ProvidesPackage{paket adı}
```

paketin adını L^AT_EX'e tanıtan komuttur. Aynı paketi iki kez yüklerseniz, L^AT_EX hata mesajıyla sizi uyarır. Şekil 6.1 de, daha önceki örneklerde tanımladığımız komut ve ortamları içeren bir paket örneği görüyorsunuz.

6.2 Yazıtipleri ve Puntolar

6.2.1 Yazıtipi Değiştiren Komutlar

\LaTeX , dokümanınızın mantıksal yapısına (kısımlar, dipnotlar, ...) bakarak onu uygun bir yazıtipinde dizer. Bazı durumlarda yazıtipini ve onun punto boyunu elle değiştirmek isteyebilirsiniz. Bu değişimler Tablo 6.1 ve 6.2'deki komutlarla yapılır. Her yazıtipinin boyu doküman sınıfı ve opsiyonlarına bağlı olarak değişir. Yazıtiplerinin mutlak boyutlarını bilmek isterseniz Tablo 6.3'den bakabilirsiniz.

```
{\small Küçük ve
\textbf{kalin kafalı} Romalılar}
{\Large kocaman bir
\textit{Italya} kurdular.}
```

Küçük ve **kalin kafalı** Romalılar kocaman bir *Italya* kurdular.

\LaTeX 'in önemli bir özelliği de yazıtipi değişimlerinin bağımsız oluşudur. Örneğin, puntoyu veya yazıtipini değiştiren komutlar verseniz de, daha önceden seçmiş olduğunuz kalın veya italik özelliklerini muhafaza eder.

Matematik kipinde çalışırken, değişik yazıtipinde düz metin girmek isterseniz, `\mbox{}` komutuyla geçici olarak matematik kipten çıkmanız gerekir. Formüllerin dizildiği yazıtipini değiştirmek için kullanılan komutlar ayrıdır. Tablo 6.4'de değişik matematik yazıtipleri komutları gösterilmiştir.

Yazıtipi punto değiştiren komutlarda, çengelli parantezler önemli rol oynarlar. Bu parantezlerle *gruplandırma* yaparak, \LaTeX komutlarının kapsamı sınırlandırılmış olur.

```
{\LARGE büyük ve {\small küçük}
harfler var}.
```

büyük ve küçük harfler var.

Yazıtipi puntosunu değiştiren komutlar buna uygun olarak, satır aralığını da değiştirirler. Fakat bunu ancak, paragraf komutunun içinde biterse

Tablo 6.1: Yazıtipleri.

<code>\textrm{...}</code>	düz	<code>\textsf{...}</code>	serifsiz
<code>\texttt{...}</code>	daktilo		
<code>\textmd{...}</code>	orta kalın	<code>\textbf{...}</code>	kalın
<code>\textup{...}</code>	dik	<code>\textit{...}</code>	<i>italik</i>
<code>\textsl{...}</code>	eğimli	<code>\textsc{...}</code>	KÜÇÜK BÜYÜKHARF
<code>\emph{...}</code>	<i>vurgulu</i>	<code>\textnormal{...}</code>	dokümanın ana yazıtipi

Tablo 6.2: Yazıtipi Puntoları.

<code>\tiny</code>	mini yazıtipi	<code>\Large</code>	daha büyük
<code>\scriptsize</code>	çok küçük yazıtipi	<code>\LARGE</code>	çok büyük
<code>\footnotesize</code>	oldukça küçük yazıtipi	<code>\huge</code>	aşırı büyük
<code>\small</code>	küçük yazıtipi	<code>\Huge</code>	en büyük
<code>\normalsize</code>	normal yazıtipi		
<code>\large</code>	büyük yazıtipi		

Tablo 6.3: Standart Doküman Sınıflarının Mutlak Puntoları.

size	10pt (varsayılan)	11pt opsiyonu	12pt opsiyonu
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Tablo 6.4: Matematik Yazıtipi.

<code>\mathrm{...}</code>	Düz yazıtipi
<code>\mathbf{...}</code>	Kalin yazıtipi
<code>\mathsf{...}</code>	Serifsiz yazıtipi
<code>\mathtt{...}</code>	Daktilo yazıtipi
<code>\mathit{...}</code>	<i>İtalik yazıtipi</i>
<code>\mathcal{...}</code>	$\mathcal{E}\mathcal{L}\mathcal{Y}\mathcal{A}\mathcal{Z}\mathcal{I}\mathcal{S}\mathcal{J}\mathcal{Y}\mathcal{A}\mathcal{Z}\mathcal{I}\mathcal{T}\mathcal{T}\mathcal{P}\mathcal{J}$
<code>\mathnormal{...}</code>	<i>Normal yazıtipi</i>

(Türkçenin Latin alfabesinde olmayan ı, ü, ğ, ... gibi harfleri matematik kipinde kullanılmaz.)

yapabilirler. Bu nedenle, parantezi erken kapamamak gerekir. Aşağıdaki iki örnekte, boş bir satır görevi yapan `\par` (paragraf sonu) komutunun yerine dikkat edin, farklı satır aralığına yolaçarlar.

```
{\Large Bunu okumayın!  
Bu doğru değil.  
Bana inanın!\par}
```

Bunu okumayın! Bu doğru değil. Bana inanın!

```
{\Large Bu da doğru değil.  
Fakat yalancı olduğumu  
unutmayın.}\par
```

Bu da doğru değil. Fakat yalancı olduğumu unutmayın.

Tüm bir paragrafın puntosunu değiştirmek isterseniz, değiştirme komutunu bir ortam olarak verebilirsiniz.

```
\begin{Large}  
Bu doğru değil.  
Ama, günümüzde ne doğru ki  
  \dots  
\end{Large}
```

Bu doğru değil. Ama, günümüzde ne doğru ki ...

Bu sizi ikide bir çengelli parantez kullanmaktan kurtarır.

6.2.2 Dikkat, Başınıza Taş Düşebilir

Bu bölümün başında belirttiğimiz gibi, hazır doküman şablonlarını böyle komutlarla elle değiştirmek L^AT_EX'in felsefesine aykırı bir iştir. Bunu mutlaka yapmanız gerekiyorsa, yazıtipi değiştirme işini `\newcommand` ile yeni bir komut şeklinde tanımlayıp kullanmanız daha doğru olur.

```
\newcommand{\oops}[1]{%  
  \textbf{#1}}  
Bu odaya \oops{girme},  
içerde kaynağı ve amacı  
bilinmeyen makinalar var.
```

Bu odaya **girme**, içerde kaynağı ve amacı bilinmeyen makinalar var.

Bu yöntemin avantajı, daha sonra `\textbf` (kalın) yerine başka bir yazıtipi kullanmak isterseniz, sadece komutun içeriğini değiştirir, doküman içinde kalın yazılmış yerleri bir bir aramak zahmetinden kurtulursunuz.

6.2.3 Tavsiye

Yazıtipi ve puntolar diyarına yaptığımız bu yolculuğu bitirken, küçük bir tavsiyede bulunmak isterim:

Unutmayın! *Doküman* içinde **ne kadar** çok yazı tipi kullanırsanız, *eseriniz* daha GÜZEL ve *okunaklı olacaktır*.

6.3 Aralıklar

6.3.1 Satır Aralığı

Satır aralıklarını daha büyük yapmak isterseniz dokümanın sahanlığında şu komutu kullanırsınız:

```
\linespread{çarpın}
```

Buradaki *çarpın* parametresi normal satır aralığının çarpımı kadar bir aralığı belirtir. Varsayılan değeri 1 dir. “Bir buçuk” denilen aralığı elde etmek için `\linespread{1.3}`, ve “çift aralık” için `\linespread{1.6}` kullanılır.

Fakat, `\linespread` komutu yayınlanmak üzere hazırlayacağınız dokümanların bazı yerlerinde kaba etkiler yapabilir. Gerçekten satır aralığını değiştirmek istiyorsanız, şu komutu tercih edin:

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
{1.5\baselineskip}
Bu paragraf baseline skip çarpımını
1.5 olarak dizilmiştir. Paragraf
sonundaki komuta dikkat edin.\par}
```

Bu paragrafın amacı da, çengelli parantez kapatıldığında herşeyin normale döndüğünü göstermektir.

Bu paragraf baseline skip çarpımını 1.5 olarak dizilmiştir. Paragraf sonundaki komuta dikkat edin.

Bu paragrafın amacı da, çengelli parantez kapatıldığında herşeyin normale döndüğünü göstermektir.

6.3.2 Paragraf Formatlama

L^AT_EX'in paragraf dizgisini ayarlayan iki komutu daha vardır. Dokümanınızın sahanlığına şöyle iki uzunluk ayarı koyabilirsiniz:

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

Bir paragrafı biraz içerden başlatmaya çentik atmak denir. Bu komutların birincisi paragraf çentiğini sıfırlar, diğeri iki paragraf arasındaki boşluğu artırır.

İkinci komut içindeki `plus` (artı) ve `minus` (eksi) değerleri, \LaTeX 'e paragraflar arası boşluğu, icabında belirtilen miktarlarda azaltıp çoğaltabileceğini söyler, böylece paragrafların sayfaya oturduğu daha düzgün olur.

Bazı Kıta Avrupası ülkelerinde paragraflar çentiksiz başlar, buna karşılık iki paragraf arasında biraz boşluk bırakılır. Fakat dikkat, bu durumda İçindekiler tablosu da geniş aralıklı dizilecektir. Bunu önlemek için, yukardaki iki komutu sahanlıktan çıkarıp doküman içinde `\tableofcontents` komutundan daha sonraki bir yere alırsınız, veya hiç kullanmazsınız. Gerçekten de profesyonel dizilmiş kitapların çoğunda paragraflar arası fazladan boşluk bırakılmaz, sadece her paragrafa çentik atılır.

Çentiksiz bir paragrafa çentik açmak için, paragrafın başında şu komut verilir:

```
\indent
```

Tabii bu komut, `\parindent` komutunun belirlediği çentik miktarı sıfırdan farklı ise bir anlam taşır.¹

Bir paragrafın çentiğini yoketmek istiyorsanız, paragraf başında şu komutu kullanırsınız:

```
\noindent
```

Bu komut, başlık falan atmayıp, doğrudan doküman metni yazmak istediğinizde işe yarayabilir.

6.3.3 Yatay Aralıklar

\LaTeX kelimeler ve cümleler arasındaki boşlukları otomatik olarak ayarlar. Yatay aralığı artırmak isterseniz şu komutu kullanırsınız:

```
\hspace{uzunluk}
```

Satır başına veya sonuna rasgelse dahi, bu boşluğu korumak istiyorsanız, yıldızlı `\hspace*` komutu kullanırsınız. Buradaki *length*, birimiyle birlikte verilen bir değerdir. En çok kullanılan birimler Tablo 6.5 de gösterilmiştir.

Bu aralık `\hspace{1.5cm}`
1.5 cm genişliktedir.

Bu aralık 1.5 cm genişliktedir.

¹Dokümanlarda her bölüm ve kısmın ilk paragrafı çentiksiz başlatılır. Bunu önlemek, yani ilk paragrafa da çentik açmak isterseniz, `indentfirst` paketini kullanabilirsiniz.

Tablo 6.5: T_EX Uzunluk Birimleri.

mm	milimetre $\approx 1/25$ inch	□
cm	santimetre = 10 mm	□
in	inç = 25.4 mm	□
pt	punto $\approx 1/72$ inch $\approx \frac{1}{3}$ mm	□
em	Kullanılan yazıtipinde 'M' harfinin genişliği	□
ex	Kullanılan yazıtipinde 'x' harfinin yüksekliği	□

Boşluğu esnek yapmak için,

`\stretch{n}`

komutu kullanılır. Bu komut, satırın geriye kalan kısmını kaplayacak şekilde genişleyen bir aralık bırakır. Aynı satırda iki tane `\hspace{\stretch{n}}` komutu kullanılmışsa, herbiri belirtilen oranda genişler.

`x\hspace{\stretch{1}}`
`x\hspace{\stretch{3}}x`

x x x

Metin içinde yatay boşluk bırakırken, miktarını kullanılan yazıtipiyle orantılı vermek daha tutarlı bir yoldur. Bu amaçla, metin puntosuna bağlı iki uzunluk birimi, `em` ve `ex` kullanılır:

`{\Large}Büyük\hspace{1em}y\{\`
`{\tiny}mini\hspace{1em}y}`

Büyük y
 mini y

6.3.4 Düşey Aralıklar

Paragraf, kısım, altkısım, ... gibi birimler arasındaki her türlü düşey aralıkları L^AT_EX otomatik olarak ayarlar. Eğer gerekiyorsa, iki paragraf arasındaki boşluğu şu komutla artırabilirsiniz:

`\vspace{uzunluk}`

Bu komut iki boş satır arasında verilmelidir . Bir sayfanın başında veya sonunda yine de bu boşluğu açmak istiyorsanız, komutu `\vspace*` şeklinde yıldızlı vermelisiniz.

`\stretch` (esnet) komutu `\pagebreak` (sayfa kes) komutuyla birlikte kullanıldığında iki işi yapar: sayfanın son satırına metin girmek, veya bir metni düşey doğrultuda ortaya yerleştirmek.

Bir takım satırlar `\ldots`

`\vspace{\stretch{1}}`

Bu satır sayfanın dibine gider.`\pagebreak`

Aynı bir paragrafın iki satırı arasında, veya bir tablonun satırları arasında ilave aralık açmak için,

`\[uzunluk]`

komutu kullanılır.

Ayrıca, `\bigskip` (büyük atlat) ve `\smallskip` (küçük atlat) komutları, miktar hesaplamayla uğraşmadan, belli oranda düşey boşluklar açarlar.

6.4 Sayfa Düzeni

$\text{\LaTeX} 2_{\epsilon}$ size `\documentclass` komutuyla birlikte sayfa boyutunu seçme olanağı verir. Bunu seçtiğiniz zaman, marjları, tepelik ve diplik konumlarını, ... kendisi otomatik ayarlar. Elbette bu düzenlemeyi değiştirebilirsiniz. Şekil 6.2 de değiştirebileceğiniz parametreler gösterilmiştir. Bu şekil `layout` paketi ile hazırlanmıştır.²

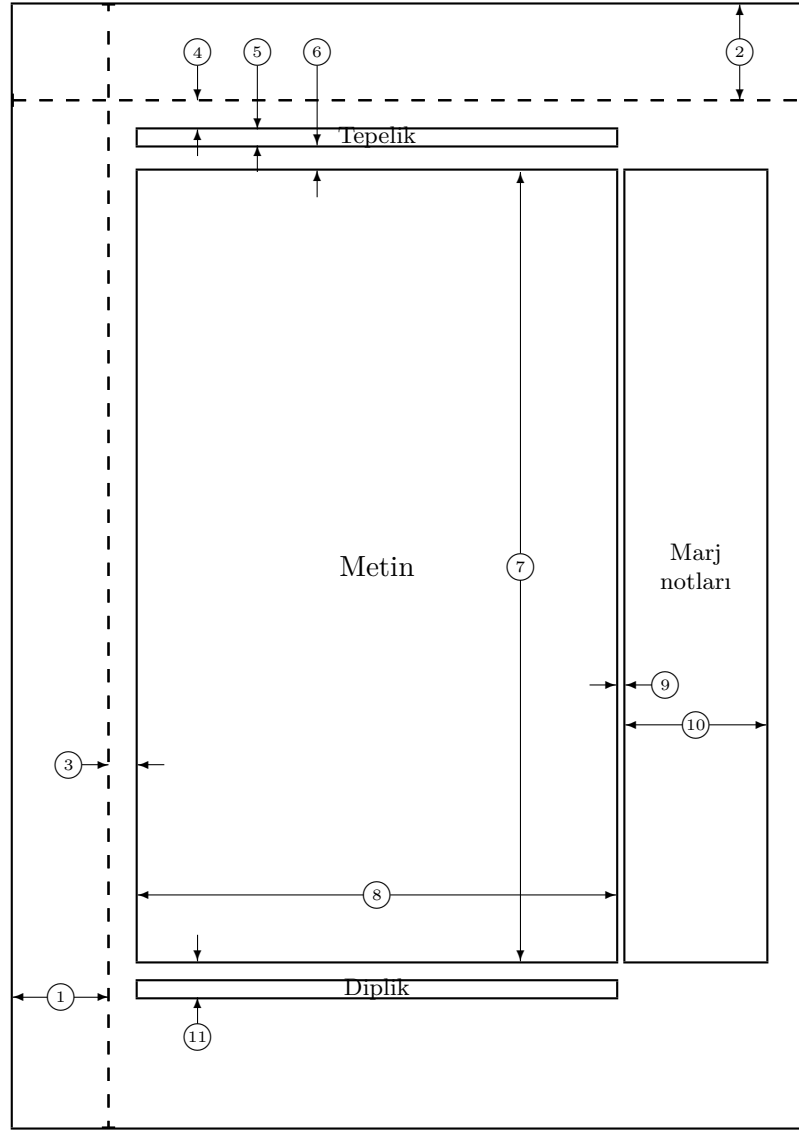
DURUN! “Şu sayfayı biraz geniş yapayım,” diye havalanmadan önce, biraz düşünün. \LaTeX 'te her şey gibi, sayfa düzeninin de bir mantığı vardır.

Bu sayfa düzeni belki MS Word programındaki gibi geniş değil, pek dar görünüyor. Ama, elinizin altında bulunan kaliteli bir kitaba bakın ve bir satırdaki harf sayısını hele bir sayın. Her satırda 66 harften fazla bulunmadığını görürsünüz. Aynı şeyi \LaTeX 'de dizilmiş bir sayfa için tekrarlayın. Onun da 66 harften fazla olmadığını göreceksiniz. Tecrübeler bir satırda daha fazla harf bulunursa, okumanın güçleştiğini söylüyor. Çünkü, göz bir satır sonundan diğerin başına dönmekte zorlanır. Bu yüzden gazeteler dar sütunlar halinde dizilirler.

İşte bu yüzden, satır genişliğini artırırsanız, okuyucunun makalenizi okumasını zorlaştırırsınız. Neyse, sizi uyardıktan sonra, bu işi nasıl yapacağınızı anlatabayım.

Bu parametreleri değiştirmek için \LaTeX 'in iki komutu vardır. Genellikle sahanlık kısmında kullanılırlar.

²`macros/latex/required/tools`



1	bir inç + <code>\hoffset</code>	2	bir inç + <code>\voffset</code>
3	<code>\oddsidemargin = 22pt</code> or <code>\evensidemargin</code>	4	<code>\topmargin = 22pt</code>
5	<code>\headheight = 12pt</code>	6	<code>\headsep = 19pt</code>
7	<code>\textheight = 595pt</code>	8	<code>\textwidth = 360pt</code>
9	<code>\marginparsep = 7pt</code>	10	<code>\marginparwidth = 106pt</code> <code>\marginparpush = 5pt</code> (gösterilmedi)
11	<code>\footskip = 27pt</code> <code>\hoffset = 0pt</code> <code>\paperwidth = 597pt</code>		<code>\voffset = 0pt</code> <code>\paperheight = 845pt</code>

Şekil 6.2: Sayfa Düzeni Parametreleri.

Birinci komut belli bir parametreye sabit bir değer verir:

```
\setlength{parametre}{uzunluk}
```

İkinci komut belli bir parametrenin değerini artırır:

```
\addtolength{parametre}{uzunluk}
```

Aslında ikinci komut `\setlength` komutundan daha kullanışlıdır, çünkü mevcut duruma bağlı olarak çalışmak daha kolaydır. Örneğin, satır genişliğini 1 cm artırmak için doküman sahanlığına şu komutu girerim:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

Bu bağlamda `calc` paketine bir göz atmanızda yarar olabilir. Bu paket `\setlength` ve diğer komutlar içinde aritmetik işlemler yapmanıza olanak verir.

6.5 Aralıklarla Diğer İşlemler

\LaTeX 'de yazarken, elimden geldiğince parametrelere mutlak değerler vermekten kaçınırım. Bunun yerine, satır genişliği veya sayfa yüksekliği gibi değerler cinsinden çalışırım. Örneğin, bir şeklin genişliğini `\textwidth` (satır genişliği) cinsinden yazarım.

Aşağıdaki 3 komut bir metin parçasının genişlik, yükseklik ve derinliğini bağlı olarak belirtmenizi sağlarlar:

```
\settoheight{değişken}{metin}
\settodepth{değişken}{metin}
\settowidth{değişken}{metin}
```

Şu örnek bu komutları nasıl kullanabileceğinizi gösteriyor:

```

\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[Opt][r]{#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Burada}$a$,
$b$ -- bir dik üçgenin
dik açısına komşu kenarlardır.

$c$ -- üçgenin hipotenüsüdür ve
yalnızlık çeker.

$d$ -- hiç görünmüyor.
Tuhaf değil mi?
\end{vardesc}

```

$$a^2 + b^2 = c^2$$

Burada: a , b – bir dik üçgenin dik açısına komşu kenarlardır.

c – üçgenin hipotenüsüdür ve yalnızlık çeker.

d – hiç görünmüyor. Tuhaf değil mi?

Bu örnekte, daha sonraki paragrafların çentiği “Burada:” kelimesinin kapladığı yere göre ayarlanmaktadır.

6.6 Kutular

L^AT_EX bir sayfayı düzenlerken, kutuları sağa sola yerleştirerek işe başlar. En temel düzeyde, her harf bir kutudur, bunu diğer harf kutularına yapıştırarak kelimeleri oluşturur. Bu kelime kutuları, esneme özelliği olan farklı bir yapışkanla, diğer kelimelere yapıştırılıp cümleler kurulur.

Bu bakış açısının biraz basit kaçtığının farkındayım, fakat gerçek şu ki T_EX'in çalışması kutular ve yapışkanlar ile anlaşılabilir. Sadece harfler değil, herşey kutular içine konulabilir, sonra kutu kutu içinde yer alır. O zaman L^AT_EX her kutuyu sanki tek bir harfmış gibi ele alır.

Aslında, daha önceki bölümlerde bilmeden pek çok kutu ile tanıştınız. Örneğin, `tabular` (tablo) ortamı veya `\footnote` (dipnot) komutu, hep birer kutu üretirler. Bu demektir ki siz iki tabloyu, birer kutu gibi, yan yana veya üst üste yerleştirebilirsiniz. Sadece, toplam boyutlarının sayfa boyutlarını aşmamasına dikkat edersiniz.

Herhangi bir paragrafı kutu içine almak için iki yol vardır. Birincisi,

```
\parbox[konum]{genişlik}{metin}
```

komutu, ve ikincisi

```
\begin{minipage}[konum]{genişlik} metin \end{minipage}
```

ortamı. Buradaki *konum* parametresi kutunun bulunduğu yere göre düşey

konumunu belirler ve **c**, **t** veya **b** harflerinden birini alır (orta, üst ve alt). *genişlik* kutu genişliğini belirten bir uzunluk parametresidir. `minipage` ile `\parbox` arasındaki başlıca fark şudur: Bir `parbox` içinde her komut veya ortamı kullanamazsınız, oysa `minipage` içinde herşey mümkündür.

Tüm bir paragrafı, satır kesimleriyle birlikte bir kutuya yerleştiren `\parbox` komutu yanısıra, sadece yatay hizalanmış malzeme için kullanılan kutulama komutları da vardır. Bunlardan birini daha önceden gördünüz: `\mbox` komutu. Bu komut bir dizi kelimeyi yatay yönde dizip L^AT_EX'in bunları aradan bölmesini engelliyordu. Bunun daha genişletilmiş olan şöyle bir komut vardır:

```
\makebox[genişlik][konum]{metin}
```

Burada *genişlik* kutunun dışardan görünen genişliğidir ve içerdeki metinden daha küçük olabilir. Hatta 0pt genişlikte bile yapabilirsiniz, böylece içindeki metin çevredeki kutuları etkilemeden dizilmiş olur. Genişlik parametresi içinde, `\width`, `\height`, `\depth` ve `\totalheight` komutları da kullanılabilir. Bunlar, *metin* olarak girilen malzemeyi ölçerek hesaplanırlar. *konum* parametresi bir harf değeri alır: **c** (ortala), **l** (sola yanaşık), **r** (sağa yanaşık) veya **s** (yayıllı).

`\framebox` komutu da bununla aynıdır, sadece metin çevresine gerçekten bir kutu çizer.

Aşağıdaki örnekte `\makebox` ve `\framebox` komutlarıyla neler yapabileceğinizi görebilirsiniz.

```
\makebox[\textwidth]{%
  o r t a d a }\par
\makebox[\textwidth][s]{%
  y a y ı l m ı ş}\par
\framebox[1.1\width]{Bir
  çerçeve içindeyim!} \par
\framebox[0.8\width][r]{Aman,
  fazla geniş kaldım} \par
\framebox[1cm][l]{Kafaya
  takmayalım}
Bunu okuyabilir misiniz?
```

```

          o r t a d a
y a y ı l m ı ş
  Bir çerçeve içindeyim!
Aman, fazla geniş kaldım
  Kafaya takmayalım
Bunu okuyabilir misiniz?
```

Yatay yönü kontrol altına aldıktan sonra, dikey yöne bakalım. L^AT_EX için problem değil. Bir kutunun dikey özelliklerini tanımlayan komut şöyledir:

```
\raisebox{yüksekti}[üst-sınır][alt-sınır]{metin}
```

Burada *yüksekti*, metnin satır tabanından ne kadar yukarda dizileceğini belirtir, eksi değer verilirse alta dizebilir. Diğer iki opsiyonel komut, metnin yukarda ve aşağıda en fazla uzanabileceği mesafeleri belirtirler. Yine, ilk üç

parametre içinde de `\width`, `\height`, `\depth`, ve `\totalheight` komutları kullanabilirsiniz.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}}
```

diye bağırdı ama alt satırda bile kimse ona ne olduğunu anlayamadı.

Aaaaaaaar diye bağırdı ama alt satırda bile kimse ona ne olduğunu anlayamadı.

6.7 Çizgiler ve Payandalar

Birkaç sayfa önce şöyle bir komut geçtiğini farketmişsinizdir:

```
\rule[yükselti]{genişlik}{yükseklik}
```

Normal kullanımda bu komut kara bir kutu oluşturur.

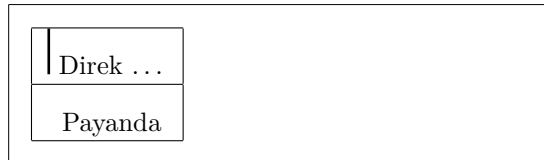
```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule[8mm]{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



Bu komut yatay ve dikey çizgiler çekmekte çok kullanışlıdır. Örneğin, bu sayfanın tepeliğindeki çizgi `\rule` komutuyla çizildi.

Çizgi komutunun özel bir hali, belli yükseklikte ama genişliği sıfır olan çizgidir. Dizgiciler buna payanda derler. Bu, sayfa içindeki bir nesnenin asgari yüksekliğini garantilemek için kullanılır. Örneğin, `tabular` (tablo) ortamında, bir satırın minimum bir yüksekliğe sahip olması için, görünmez bir payanda koyulur.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex} Direk \ldots\
\hline
\rule{0pt}{4ex} Payanda \
\hline
\end{tabular}
```



Kaynakça

- [1] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The T_EXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.
- [3] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley. *The L^AT_EX Companion, (2nd Edition)*. Addison-Wesley, Reading, Massachusetts, 2004, ISBN 0-201-36299-6.
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.
- [5] Her L^AT_EX kurulumunda *L^AT_EX Local Guide* (Yerel Rehber) adlı bir doküman bulunur, ve o yerel sisteme özel konuları açıklar. Normalde `local.tex` adlı bir dosyada bulunmalıdır. Fakat, bazı ihmalkar sistem operatörleri bunu hazırlamazlar. Bu durumda, çevrenizdeki L^AT_EX uzmanı varsa, ona danışın.
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. L^AT_EX 2_ε dağıtımında `usrguide.tex` adıyla gelir.
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε for Class and Package writers*. L^AT_EX 2_ε dağıtımında `clsguide.tex` adıyla gelir.
- [8] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. L^AT_EX 2_ε dağıtımında `fntguide.tex` adıyla gelir.
- [9] D. P. Carlisle. *Packages in the 'graphics' bundle*. L^AT_EX dağıtımında aldığımız yerde bulunan 'graphics' bohçasında `grfguide.tex` adıyla bulunur.
- [10] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of L^AT_EX's verbatim Environments*. L^AT_EX dağıtımında aldığımız yerde bulunan 'tools' bohçasında `verbatim.dtx` adıyla bulunur.

- [11] Vladimir Volovich, Werner Lemberg and L^AT_EX3 Project Team. *Cyrillic languages support in L^AT_EX*. Comes with the L^AT_EX 2_ε distribution as `cyrguide.tex`.
- [12] Graham Williams. *The TeX Catalogue* is a very complete listing of many T_EX and L^AT_EX related packages. Available online from CTAN: [/tex-archive/help/Catalogue/catalogue.html](http://tug.ctan.org/help/Catalogue/catalogue.html)
- [13] Keith Reckdahl. *Using EPS Graphics in L^AT_EX 2_ε Documents*, which explains everything and much more than you ever wanted to know about EPS files and their use in L^AT_EX documents. Available online from CTAN: [/tex-archive/info/epslatex.ps](http://tug.ctan.org/info/epslatex.ps)
- [14] Kristoffer H. Rose. *X_Y-pic User's Guide*. Downloadable from CTAN with X_Y-pic distribution
- [15] John D. Hobby. *A User's Manual for MetaPost*. Downloadable from <http://cm.bell-labs.com/who/hobby/>
- [16] Alan Hoenig. *T_EX Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)
- [17] Urs Oswald. *Graphics in L^AT_EX 2_ε*, içinde değişik boyda çember ve elips çizdirmek için Java kaynak kodları bulunur, ve *MetaPost - A Tutorial*. Her ikisi de <http://www.ursoswald.ch> sitesinden indirilebilir.

DİZİN

- \!, 47
- ", 21
- \$, 41
- \(, 41
- \), 41
- \., 42, 47
- , 22
- , 22
- \-, 20
- , 22
- , 22
- ., dan sonra boşluk, 28
- ..., 23
- \:, 47
- \;, 47
- \@, 28
- \[, 42
- \[, 19, 33, 34, 36, 97
- *, 19
- \], 42
- ~, 28

- A4 kağıt, 11
- A5 kağıt, 11
- å, 24
- abstract, 35
- Acrobat Reader, 71
- \addtolength, 99
- æ, 24
- akboşluk
 - komuttan sonraki, 5
 - satır başında, 4
- \aklasor, 80, 81
- aksanlı Fransız harfleri, 24
- Almanca, 27
- amsbsy, 53

- amsmath, 43, 61
- amsmath, 42, 45–48, 50, 51, 53
- amssymb, 43, 54
- amsthm, 52, 53
- ana başlığı, 30
- ana yazıtipi puntosu, 11
- \and, 30
- ansinew, 26
- \appendix, 30, 31
- applemac, 26
- \ar, 83, 84
- \arccos, 45
- \arcsin, 45
- \arctan, 45
- \arg, 45
- array, 48, 49
- article (makale) sınıfı, 10
- \author, 30

- B5 kağıt, 11
- babel, 20, 25
- \backmatter, 31
- \backslash, 5
- \begin, 32, 74, 82
- \bibitem, 65
- \Big, 47
- \big, 47
- \Bigg, 47
- \bigg, 47
- \biggl, 51
- \biggr, 51
- \bigskip, 97
- \binom, 45
- birimler, 95, 96
- \bklasor, 81
- bm, 53

- `\bmod`, 45
- `\boldmath`, 53
- `\boldsymbol`, 53
 - book (kitap) sınıfı, 10
 - boş, 10
 - boyutlar, 95
- calc, 99
- `\caption`, 39, 40
- `\cdot`, 45
- `\cdots`, 47
 - center, 33
- `\chapter`, 29
- `\chaptermark`, 67, 69
- `\ci`, 87
- `\circle`, 77
- `\circle*`, 77
- `\cite`, 65
- `\cleardoublepage`, 40
- `\clearpage`, 40
- `\cline`, 36
 - comment, 6
- `\cos`, 45
- `\cosh`, 45
- `\cot`, 45
- `\coth`, 45
 - cp1251, 26
 - cp850, 26
 - cp866nav, 26
- `\csc`, 45
 - çarpım işareti, 46
 - çengelli parantez, 91
 - çift satır aralığı, 94
 - çift sütun, 11
- `\date`, 30
- `\ddots`, 47
- `\deg`, 45
 - denklemler sistemi, 49
- `\depth`, 101, 102
 - Derece işaretini, 22
 - description, 33
- `\det`, 45
 - dik, 91
 - dikey üç nokta, 47
- `\dim`, 45
 - diplik, 10
 - displaymath, 42
- `\displaystyle`, 51
 - dizin, 66
 - doc, 12
- `\documentclass`, 9, 14, 20
 - Doküman başlığı, 11
 - doküman yazıtipi puntosu, 11
 - dosya türleri, 13
 - double sided, 11
- `\dq`, 28
- `\dum`, 87
 - düşey aralık, 96
 - düz, 91
 - dvips, 64, 70
- eepic, 73, 77
 - eğimli, 91
 - eksi işareti, 22
- `\emph`, 32, 91
 - Encapsulated POSTSCRIPT, 63
- `\end`, 32, 74
 - enerji, 67
 - enumerate, 33
 - epic, 73
 - eqnarray, 49
- `\eqref`, 42
 - equation, 42
 - eurosym, 23
- `\EURtm`, 23
 - executive kağıt boyutu, 11
- `\exp`, 45
 - exscale, 12, 47
 - extension
 - .aux, 14
 - .cls, 14
 - .dtx, 13
 - .dvi, 14
 - .eps, 64
 - .fd, 14
 - .idx, 14
 - .ilg, 14

- .ind, 14, 66
- .ins, 13
- .lof, 14
- .log, 14
- .lot, 14
- .sty, 13, 69
- .tex, 8, 13
- .toc, 14

- fancyhdr, 67–69
- \fbox, 21
- figure, 38, 39
- \flq, 28
- \flqq, 28
- flushleft, 33
- flushright, 33
- foiltex, 10
- font encodings, 26
- font size, 92
- fontenc, 12, 26
- \footnote, 31, 40, 100
- \footnotesize, 92
- \footskip, 98
- formüller, 41
- \frac, 45
- \framebox, 101
- \frenchspacing, 29
- \frontmatter, 30
- \frq, 28
- \frqq, 28
- \fussy, 20

- \gcd, 45
- geometry, 69
- geribölü , 5
- GhostScript, 63
- girdi dosyası, 7
- girdi kodlamaları
 - ansinew, 26
 - applemac, 26
 - cp1251, 26
 - cp850, 26
 - cp866nav, 26
 - koi8-ru, 26
 - latin1, 26
 - macukr, 26
 - utf8x, 26
- grafik, 9
- graphics, 63, 64
- graphicx, 63
- gruplandırma, 91

- \headheight, 98
- \headsep, 98
- \height, 101, 102
- hiper-metin, 70
- \hline, 36
- \hom, 45
- \hspace, 89, 95
- \Huge, 92
- \huge, 92
- hyphenat, 69
- \hyphenation, 20

- iç atıflar, 31
- \iblk, 88
- \idotsint, 48
- ifthen, 12
- \ignorespaces, 89
- \ignorespacesafterend, 89
- \iiiint, 48
- \iiint, 48
- \iint, 48
- \include, 14, 15
- \includeonly, 15
- \indent, 95
- indentfirst, 95
- \index, 66, 67
- indis, 44
- \inf, 45
- \input, 15
- inputenc, 12, 26
- \int, 46
- integral işareti, 46
- İskandinav harfleri, 24
- italik, 91
- \item, 33

- köşeli parantez, 5

- Küçük büyükharf, 91
kırılğan, 40
kağıt boyutu, 11
kalın, 91
kalın semboller, 43, 53
kalıplama, 24
karatahta kalını, 43
Karekök, 44
kaynakça, 64
\ker, 45
Knuth, Donald E., 1
kodlamalar
 girdi
 ansinew, 26
 applemac, 26
 cp1251, 26
 cp850, 26
 cp866nav, 26
 koi8-ru, 26
 latin1, 26
 macukr, 26
 utf8x, 26
 yazıtipi
 LGR, 27
 OT1, 26
 T1, 27
 T2A, 27
 T2B, 27
 T2C, 27
 X2, 27
koi8-ru, 26
komutlar
 \!, 47
 \(), 41
 \), 41
 \,, 42, 47
 \-, 20
 \:, 47
 \;, 47
 \@, 28
 \[, 42
 \\, 19, 33, 34, 36, 97
 *, 19
 \], 42
 \addtolength, 99
 \aklasor, 80, 81
 \and, 30
 \appendix, 30, 31
 \ar, 83, 84
 \arccos, 45
 \arcsin, 45
 \arctan, 45
 \arg, 45
 \author, 30
 \backmatter, 31
 \backslash, 5
 \begin, 32, 74, 82
 \bibitem, 65
 \Big, 47
 \big, 47
 \Bigg, 47
 \bigg, 47
 \biggl, 51
 \biggr, 51
 \bigskip, 97
 \binom, 45
 \bklasor, 81
 \bmod, 45
 \boldmath, 53
 \boldsymbol, 53
 \caption, 39, 40
 \cdot, 45
 \cdots, 47
 \chapter, 29
 \chaptermark, 67, 69
 \ci, 87
 \circle, 77
 \circle*, 77
 \cite, 65
 \cleardoublepage, 40
 \clearpage, 40
 \cline, 36
 \cos, 45
 \cosh, 45
 \cot, 45
 \coth, 45
 \csc, 45
 \date, 30

`\ddots`, 47
`\deg`, 45
`\depth`, 101, 102
`\det`, 45
`\dim`, 45
`\displaystyle`, 51
`\documentclass`, 9, 14, 20
`\dq`, 28
`\dum`, 87
`\emph`, 32, 91
`\end`, 32, 74
`\eqref`, 42
`\EURtm`, 23
`\exp`, 45
`\fbox`, 21
`\flq`, 28
`\flqq`, 28
`\footnote`, 31, 40, 100
`\footskip`, 98
`\frac`, 45
`\framebox`, 101
`\frenchspacing`, 29
`\frontmatter`, 30
`\frq`, 28
`\frqq`, 28
`\fussy`, 20
`\gcd`, 45
`\headheight`, 98
`\headsep`, 98
`\height`, 101, 102
`\hline`, 36
`\hom`, 45
`\hspace`, 89, 95
`\hyphenation`, 20
`\iblk`, 88
`\idotsint`, 48
`\ignorespaces`, 89
`\ignorespacesafterend`, 89
`\iiiint`, 48
`\iiint`, 48
`\iint`, 48
`\include`, 14, 15
`\includeonly`, 15
`\indent`, 95
`\index`, 66, 67
`\inf`, 45
`\input`, 15
`\int`, 46
`\item`, 33
`\ker`, 45
`\label`, 31, 42
`\LaTeX`, 21
`\LaTeXe`, 21
`\ldots`, 23, 47
`\left`, 46, 47
`\leftmark`, 67, 69
`\lg`, 45
`\lim`, 45
`\liminf`, 45
`\limsup`, 45
`\line`, 75, 81
`\linebreak`, 19
`\linespread`, 94
`\linethickness`, 78, 79, 81
`\listoffigures`, 39
`\listoftables`, 39
`\ln`, 45
`\log`, 45
`\mainmatter`, 31
`\makebox`, 101
`\makeindex`, 66
`\maketitle`, 30
`\marginparpush`, 98
`\marginparsep`, 98
`\marginparwidth`, 98
`\mathbb`, 43
`\mathrm`, 51
`\max`, 45
`\mbox`, 21, 24, 101
`\min`, 45
`\multicolumn`, 36, 37
`\multirow`, 74, 78, 79
`\newcommand`, 88
`\newenvironment`, 89
`\newline`, 19
`\newpage`, 19
`\newsavebox`, 80
`\newtheorem`, 51

<code>\newtheoremstyle</code> , 52	<code>\scriptscriptstyle</code> , 51
<code>\noindent</code> , 95	<code>\scriptstyle</code> , 51
<code>\nolinebreak</code> , 19	<code>\sec</code> , 45
<code>\nonumber</code> , 50	<code>\section</code> , 29, 40
<code>\nopagebreak</code> , 19	<code>\sectionmark</code> , 67
<code>\not</code> , 55	<code>\selectlanguage</code> , 25
<code>\oddsidemargin</code> , 98	<code>\setlength</code> , 74, 94, 99
<code>\oval</code> , 79, 81	<code>\settodepth</code> , 99
<code>\overbrace</code> , 44	<code>\settoheight</code> , 99
<code>\overleftarrow</code> , 44	<code>\settowidth</code> , 99
<code>\overline</code> , 44	<code>\sin</code> , 45
<code>\overrightarrow</code> , 44	<code>\sinh</code> , 45
<code>\pagebreak</code> , 19	<code>\sloppy</code> , 20
<code>\pageref</code> , 31, 70	<code>\smallskip</code> , 97
<code>\pagestyle</code> , 10	<code>\sqrt</code> , 44
<code>\paperheight</code> , 98	<code>\stackrel</code> , 46
<code>\paperwidth</code> , 98	<code>\stretch</code> , 89, 96
<code>\par</code> , 93	<code>\subparagraph</code> , 29
<code>\paragraph</code> , 29	<code>\subsection</code> , 29
<code>\parbox</code> , 100, 101	<code>\subsectionmark</code> , 67
<code>\parindent</code> , 94	<code>\substack</code> , 46
<code>\parskip</code> , 94	<code>\subsubsection</code> , 29
<code>\part</code> , 29	<code>\sum</code> , 46
<code>\pmod</code> , 45	<code>\sup</code> , 45
<code>\Pr</code> , 45	<code>\tableofcontents</code> , 30
<code>\printindex</code> , 67	<code>\tan</code> , 45
<code>\prod</code> , 46	<code>\tanh</code> , 45
<code>\protect</code> , 40	<code>\TeX</code> , 21
<code>\providecommand</code> , 88	<code>\textcelsius</code> , 22
<code>\ProvidesPackage</code> , 90	<code>\texteuro</code> , 23
<code>\put</code> , 73–80	<code>\textheight</code> , 98
<code>\qbezier</code> , 73, 74, 81	<code>\textrm</code> , 51
<code>\qedhere</code> , 53	<code>\textstyle</code> , 51
<code>\qqquad</code> , 42, 47	<code>\textwidth</code> , 98
<code>\quad</code> , 42, 47	<code>\thicklines</code> , 76, 79
<code>\raisebox</code> , 101	<code>\thinlines</code> , 79
<code>\ref</code> , 31, 42, 70	<code>\thispagestyle</code> , 13
<code>\renewcommand</code> , 88	<code>\title</code> , 30
<code>\renewenvironment</code> , 89	<code>\today</code> , 21
<code>\right</code> , 46, 47	<code>\topmargin</code> , 98
<code>\right.</code> , 47, 48	<code>\totalheight</code> , 101, 102
<code>\rightmark</code> , 67, 69	<code>\underbrace</code> , 44
<code>\rule</code> , 89, 102	<code>\underline</code> , 32, 44
<code>\savebox</code> , 80	<code>\unitlength</code> , 74, 75

- `\usebox`, 80
- `\usepackage`, 10, 13, 23, 25, 26, 90
- `\vdots`, 47
- `\vec`, 44
- `\vector`, 76
- `\verb`, 35
- `\verbatim`, 35
- `\verbatiminput`, 69
- `\vspace`, 96
- `\widehat`, 44
- `\widetilde`, 44
- `\width`, 101, 102
- `\xymatrix`, 83
- konum parametresi, 38
- `\label`, 31, 42
- Lamport, Leslie, 2
- `\LARGE`, 92
- `\Large`, 92
- `\large`, 92
- `\LaTeX`, 21
- LaTeX3, 4
- `\LaTeXe`, 21
- latexsym, 12
- latin1, 26
- layout, 97
- `\ldots`, 23, 47
- `\left`, 46, 47
- `\leftmark`, 67, 69
- legal kağıt boyutu, 11
- letter kağıt boyutu, 11
- `\lg`, 45
- LGR, 27
- `\lim`, 45
- `\liminf`, 45
- `\limsup`, 45
- `\line`, 75, 81
- `\linebreak`, 19
- `\linespread`, 94
- `\linethickness`, 78, 79, 81
- `\listoffigures`, 39
- `\listoftables`, 39
- `\ln`, 45
- `\log`, 45
- longtable, 37
- lscommand, 87
- macukr, 26
- `\mainmatter`, 31
- `\makebox`, 101
- makeidx, 12, 66
- makeidx paketi, 66
- `\makeindex`, 66
- makeindex programı, 66
- `\maketitle`, 30
- `\marginparpush`, 98
- `\marginparsep`, 98
- `\marginparwidth`, 98
- marjları, 97
- marvosym, 23
- matematik, 41
 - aksanlar, 44
 - fonksiyonlar, 45
 - gruplandırıcı, 47
- matematik yazıtipi puntosu, 50
- matematikte boşluk, 47
- math, 41
- `\mathbb`, 43
- `\mathbf`, 92
- `\mathcal`, 92
- matematik
 - eksi, 22
- `\mathit`, 92
- `\mathnormal`, 92
- `\mathrm`, 51, 92
- mathrsfs, 61
- `\mathsf`, 92
- `\mathtt`, 92
- `\max`, 45
- `\mbox`, 21, 24, 101
- `\min`, 45
 - minimal sınıf, 10
- minipage, 100, 101
- Mittelbach, Frank, 2
- Modülo fonksiyonu, 45
- `\multicolumn`, 36, 37
- `\multipt`, 74, 78, 79

- `\newcommand`, 88
- `\newenvironment`, 89
- `\newline`, 19
- `\newpage`, 19
- `\newsavebox`, 80
- `\newtheorem`, 51
- `\newtheoremstyle`, 52
- `\noindent`, 95
 - nokta, 23
 - noktasız ı ve j, 24
- `\nolinebreak`, 19
- `\nonumber`, 50
- `\nopagebreak`, 19
- `\normalsize`, 92
- `\not`, 55

- `\oddsidemargin`, 98
 - œ, 24
 - ok işareti, 44
 - ondalık hizalama, 37
 - opsiyonlar, 9
 - orta-tire, 22
 - ortamlar
 - abstract, 35
 - array, 48, 49
 - center, 33
 - comment, 6
 - description, 33
 - displaymath, 42
 - enumerate, 33
 - eqnarray, 49
 - equation, 42
 - figure, 38, 39
 - flushleft, 33
 - flushright, 33
 - lscommand, 87
 - math, 41
 - minipage, 100, 101
 - parbox, 101
 - picture, 73, 74, 77, 78
 - proof, 53
 - pspicture, 73
 - quotation, 34
 - quote, 34
 - subarray, 46
 - table, 38, 39
 - tabular, 36, 100
 - thebibliography, 64
 - verbatim, 35, 69
 - verse, 34
- OT1, 26
- `\oval`, 79, 81
- `\overbrace`, 44
 - overflow hbox, 19
- `\overleftarrow`, 44
- `\overline`, 44
- `\overrightarrow`, 44

- özel karakterler, 5

- paketler
 - amssys, 53
 - amsmath, 43, 61
 - amsmath, 42, 45–48, 50, 51, 53
 - amssymb, 43, 54
 - amsthm, 52, 53
 - babel, 20, 25
 - bm, 53
 - calc, 99
 - doc, 12
 - dvips, 64, 70
 - eepic, 73, 77
 - epic, 73
 - eursym, 23
 - exscale, 12, 47
 - fancyhdr, 67–69
 - fontenc, 12, 26
 - geometry, 69
 - graphics, 63, 64
 - graphicx, 63
 - hyphenat, 69
 - ifthen, 12
 - indentfirst, 95
 - inputenc, 12, 26
 - latexsym, 12
 - layout, 97
 - longtable, 37
 - makeidx, 12, 66

- marvosym, 23
- mathrsfs, 61
- pdflatex, 70
- pstricks, 73, 77
- showidx, 67
- syntonly, 12, 15
- textcomp, 22, 23
- ucs, 26
- verbatim, 6, 69
- \pagebreak, 19
- \pageref, 31, 70
- \pagestyle, 10
- paket, 87
- paketler, 9
- paketleri, 7
- \paperheight, 98
- \paperwidth, 98
- \par, 93
- paragraf, 17
- \paragraph, 29
- parametre, 5
- \parbox, 100, 101
- parbox, 101
- \parindent, 94
- \parskip, 94
- \part, 29
- payanda, 102
- PDF, 70
- pdflatex, 70
- picture, 73, 74, 77, 78
- \pmod, 45
- POSTSCRIPT, 3, 9, 40, 63, 64, 71–73
- Encapsulated, 63
- \Pr, 45
- \printindex, 67
- proc (bildiri) sınıfı, 10
- \prod, 46
- proof, 53
- \protect, 40
- \providecommand, 88
- \ProvidesPackage, 90
- pspicture, 73
- pstricks, 73, 77
- punto, 91
- \put, 73–80
- \qbezier, 73, 74, 81
- \qedhere, 53
- \qqquad, 42, 47
- \quad, 42, 47
- quotation, 34
- quote, 34
- \raisebox, 101
- \ref, 31, 42, 70
- \renewcommand, 88
- \renewenvironment, 89
- renkli harfler, 9
- report (rapor) sınıfı, 10
- \right, 46, 47
- \right., 47, 48
- \rightmark, 67, 69
- \rule, 89, 102
- sade, 10
- sahanlık, 7
- satır aralığı, 94
- \savebox, 80
- sayfa biçimi, 10
- boş, 10
- sade, 10
- tepelik, 10
- sayfa boyutu, 97
- sayfa düzeni, 97
- \scriptscriptstyle, 51
- \scriptsize, 92
- \scriptstyle, 51
- \sec, 45
- \section, 29, 40
- \sectionmark, 67
- \selectlanguage, 25
- serifsiz, 91
- \setlength, 74, 94, 99
- \settodepth, 99
- \settoheight, 99
- \settowidth, 99
- showidx, 67
- sigma toplama işareti, 46
- \sin, 45

- single sided, 11
- \sinh, 45
- slides (asetat) smifi, 10
- \sloppy, 20
- \small, 92
- \smallskip, 97
- \sqrt, 44
- \stackrel, 46
- \stretch, 89, 96
- subarray, 46
- \subparagraph, 29
- \subsection, 29
- \subsectionmark, 67
- \substack, 46
- \subsubsection, 29
- \sum, 46
- \sup, 45
- syntonly, 12, 15

- T1, 27
- T2A, 27
- T2B, 27
- T2C, 27
- türev, 44
- tırnak işaretleri, 21
- table, 38, 39
- \tableofcontents, 30
- tabular, 36, 100
- \tan, 45
- \tanh, 45
- tek sütun, 11
- tepelik, 10
- texttttepelik, 10
- \TeX, 21
- \textbf, 91
- \textcelsius, 22
- textcomp, 22, 23
- \texteuro, 23
- \textheight, 98
- \textit, 91
- \textmd, 91
- \textnormal, 91
- \textrm, 51, 91
- \textsc, 91
- \textsf, 91
- \textsl, 91
- \textstyle, 51
- \texttt, 91
- \textup, 91
- \textwidth, 98
- thebibliography, 64
- \thicklines, 76, 79
- \thinlines, 79
- \thispagestyle, 13
- tilda, 22
- tilda (~), 28
- \tiny, 92
- tire, 22
- title, 11
- \title, 30
- \today, 21
- \topmargin, 98
- \totalheight, 101, 102

- ucs, 26
- uluslararası, 25
- umlaut, 24
- \underbrace, 44
- underfull hbox, 20
- \underline, 32, 44
- \unitlength, 74, 75
- URL, 22
- \usebox, 80
- \usepackage, 10, 13, 23, 25, 26, 90
- utf8x, 26
- uzun denklemler, 49
- uzun-tire, 22

- üç nokta, 23, 47
- üs, 44, 46

- \vdots, 47
- \vec, 44
- \vector, 76
- vektörler, 44
- \verb, 35
- verbatim, 6, 69
- \verbatim, 35
- verbatim, 35, 69

- \verbatiminput, 69
- verse, 34
- virgül, 23
- \vspace, 96
- \widehat, 44
- \widetilde, 44
- \width, 101, 102
- www, 22
- WYSIWYG, 2, 3
- X2, 27
- Xpdf, 71
- \xymatrix, 83
- yüzer-gezer nesnelere, 38
- yapı, 6
- yatay
 - üç nokta, 47
 - çengel, 44
 - çizgi, 44
 - boşluk, 95
- yazıtipi, 91
 - \footnotesize, 92
 - \Huge, 92
 - \huge, 92
 - \LARGE, 92
 - \Large, 92
 - \large, 92
 - \mathbf, 92
 - \mathcal, 92
 - \mathit, 92
 - \mathnormal, 92
 - \mathrm, 92
 - \mathsf, 92
 - \mathtt, 92
 - \normalsize, 92
 - \scriptsize, 92
 - \small, 92
 - \textbf, 91
 - \textit, 91
 - \textmd, 91
 - \textnormal, 91
 - \textrm, 91
 - \textsc, 91
 - \textsf, 91
 - \textsl, 91
 - \texttt, 91
 - \textup, 91
 - \tiny, 92
- yazıtipi kodlamaları
 - LGR, 27
 - OT1, 26
 - T1, 27
 - T2A, 27
 - T2B, 27
 - T2C, 27
 - X2, 27
- yorum satırları, 6
- Yunanca harfler, 43
- zorunlu olmayan (opsiyonel) parametreler, 5

