

LaTeX Dersleri

Zafer ACAR

06/03/2022

İçindekiler

Sunuş	3
1 Temeller	4
1.1 TeX	4
1.2 LaTeX	4
1.3 Temel Farklılıklar	4
1.4 Neden LaTeX	5
1.5 Önemli Yapılar	6
1.5.1 Komutlar	6
1.5.2 Paketler	6
1.5.3 Ortamlar	7
1.5.4 Gruplar	7
1.5.5 Boşluklar	7
1.5.6 Yorum satırları	8
1.5.7 Özel amaçlı karakterler	8
1.6 Kurulum	9
1.6.1 GNU/Linux	9
1.6.2 Mac OS	10
1.6.3 Windows	10
1.6.4 LaTeX editörleri	10
1.6.5 Çevrimiçi editörler	10
2 İlk Adımlar	11
2.1 Tipik Bir Belge Yazımı	11
2.2 Belge Sınıfları ve Seçenekleri	12
2.3 Türkçe Dil Ayarları ve Çoklu Dil Kullanımı	13
2.4 Belgeye Başlık Oluşturma	15
2.5 Bölümlleme ve İçindekiler Tablosu	15
2.6 Büyük Projeler	18
3 Dizgi	22
3.1 Heceleme	22
4 Matematiksel İfadeler	23
4.1 Giriş	23
4.1.1 Genel	23

4.1.2	Matematik kipiyle metin kipi arasındaki farklar	25
4.1.3	Gruplandırma	25
4.1.4	Boşluklar	26
4.2	Parantezler, Gruplandırıcılar ve Oklar	27
4.3	Yunan Harfleri	30
4.4	Fonksiyonlar	31
4.5	Yığın Simgeleri	33
4.6	Matrisler	35
4.7	Yazı Biçem ve Boyutları	37
4.8	Denklemler Ortamları	39
4.9	Teorem ve Benzeri Ortamlar	44
4.9.1	Temel Teoremler	44
4.9.2	Sayaçlar	45
4.9.3	Kanıtlar	47
4.9.4	Teorem Stilleri	49
5	Özel Sayfalar	52
5.1	Kaynakça	52
5.1.1	Bütünleşik Kaynakça	52
5.1.2	Kaynakçanın ayrı dosyada hazırlanması	54
5.2	Dizin	60
5.2.1	Çoklu Dizin	61
5.2.2	Dizin İçindekiler tablosuna yazılması	63
6	Özelleştirmeler	64
	Kaynakça	65

Sunuş

Bu kitap, artık aktif olmayan <https://texdizgi.com/> adresinde yazmış olduğum yazıların derlenmesiyle ortaya çıktı. Aslında oradaki yazılar da LaTeX’le ilgili yazmaya başladığım ve bir türlü sonunun getiremediğim kitabın tamamlanmış bölümlerinin ürünüydü.

Kitabı doğal olarak LaTeX’le yazıyordum. [RMarkdown](#) ve [Quarto](#) ile tanıştıktan sonra kitabı yalnızca PDF olarak değil, etkileşime izin vermesi ve daha bir çok avantajından dolayı HTML olarak da dağıtmaya karar verdim.

Kitap etkileşimli hale [TeXLive.net](https://texlive.net) sayesinde geldi. Dolayısıyla okurun internet bağlantısı olduğunda HTML sürümünü kullanması önerilir.

```
\documentclass{article}
\begin{document}
  Merhaba \LaTeX!
\end{document}
```

Yukarıdaki “Derle” düğmesine bastığımızda LaTeX çıktısı ile “Çıktıyı Temizle” düğmesi görünecek.¹ Dilerseniz bu düğmeye basarak çıktıyı temizleyebilirsiniz. Ayrıca içeriği değiştirerek kendi denemelerinizi yapabilirsiniz. Bu sizin öğrenmenizi hızlandıracaktır.

Tüm bunların sayesinde okurun harici bir kaynağa ihtiyaç duymadan LaTeX’de epey ilerleyebileceğini düşünüyorum. Keyifli okumalar.

¹Bu düğmelerin PDF sürümde olmayacağını unutmayın.

1 Temeller

1.1 TeX

TEX, birçok matematiksel ve teknik ifade içeren belgeleri yüksek kaliteli çıktı ile üretmek için Stanford Üniversitesi'nden Profesör [Donald E.Knuth](#) tarafından, başlangıçta “[Bilgisayar Programlama Sanatı](#)” adlı kitap serisinin hazırlanması için tasarlanmış bir dizgi programıdır. Teknik kitaplar ve makaleler üreten birçok yazar ve yayıncı tarafından benimsenmiştir. TeX, Knuth tarafından ücretsiz olarak kullanıma sunulmuştur.

1.2 LaTeX

LaTeX (La(import)TeX), [Leslie Lamport](#) tarafından TeX üzerine kurulmuştur. Amacı TeX'i daha işlevli ve kolay hale getirmektir.

Günümüzde Frank Mittelbach liderliğinde bir grup programcı tarafından [geliştirilmektedir](#).

1.3 Temel Farklılıklar

LaTeX, genelde WYSIWYG (*Ne Görüyorsan Onu Alırsın*) editörleriyle karşılaştırılır. WYSIWYG, Microsoft Word, Libreoffice Writer gibi kelime işlemcilere ya da Adobe Indesign gibi programlara verilen genel bir isimdir. Hepsinin ortak özelliği, girdi ile çıktının aynı anda ve birlikte görünmesidir.

LaTeX'de girdi ve çıktı ekranı farklıdır ve çıktıyı görmek için girdinin derleme işleminden geçmesi gerekir. LaTeX'in zor gibi görünmesinin bir nedeni de komutlarla çalışmasıdır. Gerçekten de WYSIWYG programlarına alışkın bir kullanıcı için bu süreç başlangıçta biraz sıkıcı ve zor olabilir. Ancak belli bir aşamaya geldiğinizde komutlarla çalışmanın aslında işleri kolaylaştırdığını göreceksinizdir. Örneğin LaTeX'le binlerce sayfadan oluşan bir kitabın içindekiler tablosunu oluşturmak için sadece `\tableofcontents` komutunu girersiniz ve LaTeX bu komutu girdiğiniz yere içindekiler tablosunu hatasız bir şekilde yazdırır. Ayrıca örnekte görüldüğü gibi LaTeX komutları doğal dile¹ oldukça yakındır.

¹Tabii ki İngilizce.

Bir metnin genel görünümü ve okunabilirliği, metnin nasıl hizalandığından ve kesildiğinden büyük ölçüde etkilenir. LaTeX, tüm paragraf için hizalamayı ve kesmeleri optimize eden son derece gelişmiş TeX algoritmalarını kullanır. Kelime işlemciler ve diğer programlar, satır başına çalıştıkları için oldukça yetersiz kalırlar. Bu, diğer şeylerin yanı sıra düzensiz aralıklara ve birçok kısa çizgiye sebep olur. Sonuçları görmemiz için Microsoft Word 2008 (Mac), Adobe InDesign CS4 ve LaTeX’le dizilmiş bir metni [şuradan](#) inceleyebilirsiniz.

Sonuç, LaTeX’in diğer programların her ikisinden de üstün olduğunu açıkça gösterir: iki kat daha az tireleme kullanır ve yine de sözcük aralığındaki varyasyon, Word veya InDesign’dan belirgin şekilde daha azdır. LaTeX’te çok büyük sözcük aralığı içeren satırlar oluşmaz.

1.4 Neden LaTeX

LaTeX’i kullanmaya karar vermeniz için birkaç neden:

- Ücretsizdir ve tüm işletim sistemlerinde düşük donanım gereksinimleriyle çalışır.
- Basit bir metin editörüyle bile kaynak dosyanızı düzenleyebilirsiniz.
- Her sürüm bir önceki sürümü içerdiğinden şimdi hazırladığınız bir belgeyi on yıl sonra da kullanabilirsiniz, hiçbir zaman “Dosya bu sürümle uyumlu değil” biçiminde bir uyarıyla karşılaşmazsınız.
- Bir belgedeki biçim veya metni diğer bir belgeye kolaylıkla taşıyabilir ve düzenleyebilirsiniz. Bu, internetten bulabileceğiniz binlerce şablonu kullanabileceğiniz anlamına gelir.
- Belgenizi LaTeX biçimlendirir, siz sadece içeriğe odaklanırsınız.
- Dipnot koymak, atıf yapmak, kaynakça, dizin ve içindekiler tablosu oluşturmak işten bile sayılmaz.
- Matematiksel formüller kullanacaksanız, bu LaTeX’in en güçlü olduğu konudur. Formülleri belgenize adeta *inci dizer gibi* dizer.
- Çok yaygındır. Uluslararası bir çok yayınevi (örneğin Springer) ve dergi yazarlardan LaTeX dosyalarını talep eder.

1.5 Önemli Yapılar

1.5.1 Komutlar

LaTeX komutları bir geribölü (\) işaretiyle başlar ve ya sadece harflerden ya da bir tane harf olmayan karakterden oluşurlar. Komut yazıldıktan sonra ya boşluk, ya bir sayı ya da harf olmayan bir karakter gelebilir.

Çoğu komut, zorunlu değişken alır. Bu zorunlu değişken komut adından sonra çengelli parantezler içine yazılır. Zorunlu değişken alan komutlar, zorunlu olmayan (isteğe bağlı) değişkenler de alabilir, bunlar da komut adından sonra gelen köşeli parantezler içine yazılırlar. Eğer değişkenler birden fazlaysa aralarına virgül koyularak ayrılır.

```
\:  
\LaTeX  
\item[...]  
\emph{...}  
\subfloat[...][...]{...}  
\raisebox{...}[...][...]{...}  
\multicolumn{...}{...}{...}  
{\bfseries ...}
```

Fikir vermesi açısından yukarıda dokuz adet komut örneği verilmiştir. Birinci komut bir tane harf olmayan karakterden oluşan bir komuttur. İkincisi, değişkeni olmayan bir komuttur. Bazı harflerin büyük bazılarının da küçük olması komutların büyük-küçük harfe duyarlı olduğunu gösterir. Dokuzuncu komut ise bildirim şeklinde verilmiştir.

1.5.2 Paketler

LaTeX’de bazı özelliklerin (renkli yazmak, şekil eklemek vb.) kullanılabilmesi için kaynak dosyaya bazı paketlerin eklenmesi gerekir. Bu, `\usepackage` komutuyla yapılır. Bu komutun zorunlu değişkenine paket adı, zorunlu olmayan kısmına ise paket seçenekleri yazılır:

```
\usepackage[<seçenekler>]{<paket adı>}
```

Bu komutla paketin kaynak dosyaya eklenmesi TeX dağıtımıyla sisteminize kurulmuş olan paketin belgeye çağrılarak işe koşulması demektir.

1.5.3 Ortamlar

LaTeX’de ortamlar önemli bir yer tutar. Örneğin `document` bir ortamdır. Ortamları birden fazla ögeye uygulanan komutlar olarak düşünebiliriz.

Bir ortam `\begin` komutuyla başlayıp `\end` komutuyla biter. Her iki komutun zorunlu değişkeni ortamın adıdır:

```
\begin{<ortam adı>
...
\end{<ortam adı>}
```

1.5.4 Gruplar

Gruplar, ortam benzeri yapılardır. Grup `\begingroup` komutuyla başlar ve `\endgroup` komutuyla biter. Grubun içinde kullanılan bir bildirim sadece gruba uygulanır.

1.5.5 Boşluklar

LaTeX’de belgenizin metnini oluştururken ister klavyedeki Space, ister Tab tuşu ile boşluk bırakın, bu boşluklar LaTeX tarafından bir karakter boşluk olarak algılanır. Arka arkaya çok sayıda boşluk bırakılsa da LaTeX bunu tek bir boşluk olarak algılar.

Bütün bir satırın boş bırakılması LaTeX tarafından paragraf başı olarak algılanır. Arka arkaya boş bırakılan çok sayıda boş satır LaTeX tarafından tek bir boş satır yani paragraf başı olarak algılanır.

```
\documentclass{article}
\begin{document}
  İster bir boşluk, isterseniz de çok          sayıda boşluk bırakın.
  İkisi de bir boşluk gibi işlem görür. Değişen bir şey yok.

  Ayrıca boş bir satır yeni paragraf demektir, burada olduğu gibi.
\end{document}
```

Komutlardan sonra gelen boşlukları LaTeX dikkate almaz. Komuttan sonra gerçekten bir boşluk bırakmak için, ya `{ }` ve ardından boşluk girilir ya da komut adından sonra özel bir boşluk komutu kullanılır.

```
\documentclass{article}
\begin{document}
```



```

\LaTeX boşluk yok.\\
\LaTeX{} boşluk var.\\
\LaTeX\ boşluk komutuyla boşluk.
\end{document}

```

1.5.6 Yorum satırları

Yazar, kaynak dosyada, baskıda gözükmeyecek, kendine hatırlatma yapmak amacıyla yorumlar eklemek isteyebilir. Bunun için % sembolü kullanılır. LaTeX, bu sembolü gördüğü anda, o satırın geri kalanındaki her şeyi işlemeyi bırakır ve bir sonraki satırın başına geçip devam eder.

```

\documentclass{article}
\begin{document}
  Yaz, % sonrası yok.
  devamı burada.
\end{document}

```

Eğer daha uzun yorumlar eklemek istenirse `\usepackage{verbatim}` komutuyla **verbatim** paketini ekledikten sonra yorum comment ortamında yazılır.

```

\documentclass{article}
\usepackage{verbatim}
\begin{document}
  Yorum eklemenin başka bir yolu da
  \begin{comment}
    Buraya uzun yorumlarınızı
    ekleyebilirsiniz. Burası
    baskıda gözükmeyecektir.
  \end{comment}
  budur.
\end{document}

```

1.5.7 Özel amaçlı karakterler

Aşağıdaki karakterlerin herbiri LaTeX’de özel bir amaç için kullanılır. Dolayısıyla bu karakterleri doğrudan kullanmak istenmeyen sonuçlara yol açabilir.

```

# $ % & { } ~ ^ _ \

```

Bu karakterleri çıktıda elde etmek isterseniz, sondaki hariç, başına bir geribölü koymanız gerekir. Sondaki için, yani bir geribölü sembolü elde etmek içinse `\textbackslash` komutunu kullanabilirsiniz. Eğer `\\` komutunu vererseniz yeni bir satır başlatmış olursunuz.

Örneğin (\$) karakteri matematik kipini açma ve kapatmaya yarar. (&) karakteri tablo ve benzeri yapılarda dikey hizalama yapmak için veya sütun ayracı olarak kullanılır. Çengelli parantezlerden ve yüzde sembolünden bahsettik. (#) karakteri yeni komutlar tanımlamakta kullanılır. Tilda (~) ise genişlemeyen bir boşluk yaratmak için kullanılır. (^) ve (_) karakterleri de matematikte üst ve alt indis yazmak için kullanılır. Her birinin kullanımlarından yeri geldiğinde tekrar bahsedeceğiz.

1.6 Kurulum

LaTeX'i kurmak için ilk olarak bir TeX dağıtımı edinmeniz gerekir. Dağıtımlar, dizgi sistemini ve LaTeX'de belge oluşturabilmek için gereken paketleri içerir.

İkinci ihtiyaç duyacağınız şey bir LaTeX editörüdür. Edindiğiniz TeX dağıtımları genelde bir LaTeX editörüyle birlikte gelir. Tabi editör kişisel bir tercihtir ve bir LaTeX editörü yerine basit bir metin editörü kullanabilirsiniz. Ancak farklı işletim sistemleri için birçok iyi LaTeX editörü vardır ve bunların kod vurgulama, otomatik tamamlama, otomatik belge oluşturma gibi LaTeX'e özgü işlevleri vardır. Dolayısıyla LaTeX'de yeniyseniz bir editör kullanmanızı tavsiye ederiz.

1.6.1 GNU/Linux

Linux sistemlere [MiKTeX](#) ya da [TeX Live](#) kurulabilir. MiKTeX'in indirme sayfasında Ubuntu, Mint, Debian, Fedora, CentOS ve openSUSE gibi Linux dağıtımlarında nasıl kurulacağı anlatılmıştır. TeX Live ise tüm popüler Linux dağıtımlarının depolarında mevcut olup, paket yöneticisi ya da komut satırı yardımıyla kurulabilir. Örneğin Ubuntu, Debian, Mint, Pardus gibi `.deb` uzantılı paketlerin kullanıldığı dağıtımlarda

```
sudo apt-get install texlive-base
```

komutuyla temel kurulum,

```
sudo apt-get install texlive-full
```

komutuyla da tam kurulum yapılır.

1.6.2 Mac OS

Mac OS kullanıcıları için iki seçenek mevcuttur: [MiKTeX](#) ya da [MacTeX](#). MiKTeX kurulumu için `.dmg` uzantılı, MacTeX içinse `.pkg` uzantılı dosya indirilir ve standart kurulum yapılır.

1.6.3 Windows

Windows için aşağıdaki dağıtımlardan birini kurabilirsiniz.

- [MiKTeX](#)
- [TeX Live](#)
- [proTeXt](#)

MiKTeX veya TeX Live dağıtımını kurarsanız sisteminize [TeXworks](#) editörü de kurulur. proTeXt dağıtımı MiKTeX tabanlı bir dağıtım olup, tüm paketleri içerir ve beraberinde [TeXstudio](#) editörüyle gelir.

1.6.4 LaTeX editörleri

Hangi editörü kullanacağınıza birkaç deneme yaptıktan sonra karar verebilirsiniz. [Burada](#) en çok beğenilen editörler listelenmiş.

Her LaTeX editöründe olan özelliklerin (otomatik kod tamamlama vb.) yanı sıra kullanıcı dostu arayüzü, yüzde yüze yakın Türkçe desteği, ücretsiz oluşu ve her üç sistemde de çalışabilmesinden dolayı [TeXstudio](#)'yu tavsiye ediyoruz. Karar sizin.

1.6.5 Çevrimiçi editörler

LaTeX'i hiçbir kurulum yapmadan çevrimiçi de kullanabilirsiniz. Aşağıda üç tanesi listelenmiştir.

- [Overleaf](#)
- [Papeeria](#)
- [LaTeX Base](#)

En popüler olanı Overleaf olup, sayfasında beğenebileceğiniz binlerce [şablon](#) ve LaTeX kullanımına yönelik [anlatımlar](#) bulunur.

2 İlk Adımlar

2.1 Tipik Bir Belge Yazımı

LaTeX'in varsayılan dosya uzantısı `.tex`'tir. Bu basit bir metin dosyası olup, LaTeX editörleriyle oluşturulup düzenlenebileceği gibi basit bir metin editörüyle de düzenlenebilir.

Bir belge hazırlamaya başlamak için verilecek ilk komut `\documentclass[...]{...}` olup, çengelli parantezler arasına oluşturmak istediğiniz belgenin sınıfı yazılır. Köşeli parantezlerin içine de isteğe bağlı bazı değişkenler yazılabilir. Eğer bu kısım boş bırakılırsa LaTeX varsayılan değerleri alacaktır. Bu komutun ardından sırasıyla `\begin{document}` ve `\end{document}` komutları verilerek belge ortamı oluşturulur. `\end{document}` komutuyla LaTeX'e belgenin bittiği söylenmiş olur ve LaTeX bu komuttan sonra girilenleri dikkate almaz.

`\documentclass` komutuyla `\begin{document}` komutu arasına *sahanlık* denir. Sahanlık, belgenin ayarlarının yapıldığı kısımdır ve bu kısım çıktıda görünmez. `\begin{document}` ile `\end{document}` arasına da *gövde* denir. İçerik burada oluşturulur.

Aşağıda asgari bir LaTeX kaynak dosyası gösterilmiştir. `\documentclass` komutunun değişkeni olan `article`, belgenin makale olacağını belirtir.

```
\documentclass{article}
\begin{document}
  İşte ilk belgem.
\end{document}
```

Bu noktadan sonra örnek kaynak dosyayı LaTeX editörünüzünde oluşturup önceden oluşturduğunuz bir dizine kaydedin. Kaydederken dosya adında boşluk ve Türkçe karakter kullanmayın. Örneğin kaynak dosyanız `belge1.tex` olsun.

İkinci aşama kaynak dosyanın derlenmesidir. Derleme işlemi için LaTeX editörlerinde genelde araç çubuğunda oklar bulunur. Oka tıklandığında dosya derlenir ve sonuç, çıktı ekranında görünür.

Eğer metin editörü kullanıyorsanız derlemeyi uçbirimde (terminal, konsol,...) yapmanız gerekir. Derleme için uçbirim kaynak dosyanın olduğu dizinde açılıp

```
pdflatex belge1
```

komutu verilmelidir.

Derleme işleminden sonra kaynak dosyanızın olduğu dizinde `belge1.tex` ve `belge1.pdf` dosyalarının yanında yine `belge1` ile başlayan farklı uzantılara sahip dosyalar olacaktır. Bu dosyaların ne olduklarına ilerleyen yazılarda değinilecektir ancak dileyen okur Oetiker et al. (2006)'e bakabilir.

2.2 Belge Sınıfları ve Seçenekleri

Başka sınıflar olmakla birlikte LaTeX'de varsayılan olarak kullanılan beş belge sınıfı vardır (Tablo 2.1).

Tablo 2.1: LaTeX'de Belge Sınıfları

Sınıf	Açıklama
<code>article</code>	Makale
<code>report</code>	Makaleden daha hacimli belgeler için kullanılır. Rapor, tez gibi
<code>book</code>	Kitap
<code>letter</code>	Mektup
<code>beamer</code>	Sunu

Bu beş sınıftan `article`, `report` ve `book` için kullanılabilecek seçenekler Tablo 2.2'de gösterilmiştir.

Tablo 2.2: LaTeX'de Belge Seçenekleri

Seçenek	Açıklama
<code>10pt</code> , <code>11pt</code> , <code>12pt</code>	Belge ana yazı büyüklüğü.
<code>a4paper</code> , <code>a5paper</code> , <code>letterpaper</code> ,...	Kağıt boyutu.
<code>fleqn</code>	Formülleri ortada yazmak yerine, sola bitişik yazar.
<code>leqno</code>	Formül numaralarını sağ yerine sol tarafa koyar.
<code>titlepage</code> , <code>notitlepage</code>	Belge başlığını attıktan sonra yeni bir sayfa açıp açmayacağını belirler.
<code>onecolumn</code> , <code>twocolumn</code>	Belgenin tek sütun veya çift sütun dizileceğini belirtir.
<code>twoside</code> , <code>oneside</code>	Belgenin kağıdın hep tek tarafına mı yoksa iki tarafına mı basılacağını belirtir.

Seenek	Aıklama
landscape	Belgeyi enine tutulmuř kağıda basılmak üzere hazırlar.
openright, openany	Belgede bölümleri hep sağı sayfalardan veya ilk gelen boş sayfadan başlatır.
draft, final	Belgeyi sırasıyla <i>taslak</i> ve <i>son</i> şeklinde hazırlar. draft seçilirse, sağı taraftan fırlamış olan satırlar kalın siyah bir çizgiyle işaretlenir.

Bu seeneklerin her birinin kullanılabilirliğı belge sınıfına göre farklılık gösterir. Tablo 2.3’de hangi seeneğın hangi sınıf için varsayılan olduğı ve kullanılabilir olup olmadığı gösterilmiştir.

Tablo 2.3: Seeneklerin belge sınıflarına göre kullanılabilirliğı (1: varsayılan 1/2: kullanılabilir 0: kullanılamaz)

Seenek	book	report	article
10pt	1	1	1
letterpaper	1	1	1
oneside	1/2	1	1
twoside	1	1/2	1/2
openany	1/2	1	0
openright	1	1/2	0
titlepage	1	1	1/2
final	1	1	1

Örneğın belgeye `\documentclass[a4paper,12pt]{article}` komutuyla başlarsak LaTeX’e kağıt boyutu A4, ana yazı büyüklüğü 12 punto olan bir makale yazacağımızı bildirmiş oluruz. Başka bir örnek `\documentclass[a5paper,11pt,twocolumn]{book}` olsun. Bu örnekte kağıt boyutu A5, ana yazı büyüklüğü 11 punto olan bir kitap yazacağımızı ve kitabın iki sütun olarak dizilmesini söyledik.

2.3 Türke Dil Ayarları ve Çoklu Dil Kullanımı

LaTeX’de Türke belgeler oluşturmak için öncelikle sahanlığa

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

komutlarının verilmesi gerekir.

T1 seçenekli **fontenc** paketi yazıtipi kodlamasıyla ilgili bir paket olup, hecelemenin doğru şekilde yapılmasını sağlar. Bir çok Avrupa dilinde de **T1** seçeneğiyle kullanılır. **turkish** seçenekli **babel** paketi de Chapter, Table, Contents,... gibi isimlerin Türkçeleşmesi (Bölüm, Tablo, İçindekiler,...) içindir.

i Not

Yakın zamana kadar ö, ş, ç,... gibi Türkçe karakterlerin kullanılabilmesi için sahanlığa `\usepackage[utf8]{inputenc}` ya da `\usepackage[latin5]{inputenc}` komutlarından birinin verilmesi gerekiyordu. Bu paket (**inputenc**) girdi kodlamasını yöneten bir pakettir. Son güncellemelerle birlikte bu paketin kullanılma zorunluluğu ortadan kalkmıştır.

Aşağıda Türkçe asgari bir LaTeX kaynak dosyası örneği verilmiştir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\begin{document}
  İşte Türkçe ilk belgem.
\end{document}
```

Türkçe dışında ikinci bir dil kullanmak isterseniz, örneğin İngilizce, **babel** paketinin seçeneğini

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

şeklinde değiştirmeniz gerekir. Burada etkin olan dil Türkçedir. İngilizceyi etkin hale getirmek için `\selectlanguage{english}` komutu kullanılır. Tekrar Türkçeye geçmek için de benzer şekilde `\selectlanguage{turkish}` komutu kullanılır.

Bir kelime ya da cümle gibi kısa metinler kullanılacaksa `\foreignlanguage` komutu kullanılabilir:

```
\foreignlanguage{<dil>}{<metin>}
```

Uzun metinler içinse diğer bir seçenek `otherlanguage` ortamıdır.

```
\begin{otherlanguage}{<dil>}
...
```

```
\end{otherlanguage}
```

Bu ortamın isimleri değiştirmeyen, örneğin, dil seçeneği İngilizce olmasına rağmen belgeye bir tablo eklediğinizde “Table” yerine yine “Tablo” adını yazan yıldızlı sürümü de (`otherlanguage*`) vardır.

2.4 Belgeye Başlık Oluşturma

LaTeX’de belgeye başlık oluşturmak için `\title` komutu kullanılır. Yazar adı `\author` komutuyla girilir. Birden fazla yazar varsa yazar adları arasına `\and` komutu girilir.

İsteğe bağlı olarak tarih için `\date` komutu kullanılır. Eğer `\date` komutu kullanılmazsa LaTeX belgenizi derlediğiniz günün tarihini basar. Tarihin basılmasını istemiyorsanız, bu komutu tarih yazılmadan `\date{}` şeklinde kullanmanız gerekir.

Son olarak, başlığın belgenize yazılması için `\begin{document}` komutundan sonra başlığı oluşturmak istediğiniz yere `\maketitle` komutunu girersiniz. Belge başlığını attıktan sonra yeni bir sayfanın açılıp açılmayacağı belgenin sınıfına bağlı olarak belirlenir.

Ayrıca `\title`, `\author` ve `\date` komutları `\thanks` komutunu içerebilir. Bu komutun değişkeni bir e-posta adresi, iş adresi veya bir teşekkür metni olabilir.

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\title{Belge Başlığı}
\author{Yazar 1\thanks{A Üniversitesi} \and Yazar 2\thanks{B Üniversitesi}}
\date{XX.XX.XXXX}
\begin{document}
\maketitle
İçerik...
\end{document}
```

2.5 Bölümleme ve İçindekiler Tablosu

LaTeX’de belgenizi bölümlere ayırmak için 7 seviye bulunmaktadır.

Tablo 2.4: LaTeX’de Bölüm Seviyeleri

Komut	Seviye	Açıklama
<code>\part{...}</code>	-1(book ve report) 0 (article)	letter hariç
<code>\chapter{...}</code>	0	sadece book ve report
<code>\section{...}</code>	1	letter hariç
<code>\subsection{...}</code>	2	letter hariç
<code>\subsubsection{...}</code>	3	letter hariç
<code>\paragraph{...}</code>	4	letter hariç
<code>\subparagraph{...}</code>	5	letter hariç

Türkçe dil paketi ekli belgelerde `\part` komutu “Kısım”, `\chapter` komutu “Bölüm” olarak yazılır. Kısımlar I, II, III,... şeklinde bölümler ise 1, 2, 3,... şeklinde numaralandırılır. `\section` komutu **book** ve **report** sınıflarında `\chapter` komutunu takip ederek 1.1, 1.2,... diğer sınıflarda 1, 2, 3,... şeklinde numaralandırılır. `\subsection` komutu da `\section` komutunu takip ederek numaralandırılır.

Türkçe dil paketi ekli belgelerde `\part` komutu “Kısım”, `\chapter` komutu “Bölüm” olarak yazılır. Kısımlar I, II, III,... şeklinde bölümler ise 1, 2, 3,... şeklinde numaralandırılır. `\section` komutu **book** ve **report** sınıflarında `\chapter` komutunu takip ederek 1.1, 1.2,... diğer sınıflarda 1, 2, 3,... şeklinde numaralandırılır. `\subsection` komutu da `\section` komutunu takip ederek numaralandırılır.

İçindekiler tablosu için LaTeX’e `\tableofcontents` komutu verilir. Bu komutun yazıldığı yerde İçindekiler tablosu oluşturulur. İçindekiler tablosunun doğru dizilmesi için kaynak dosyanızı en az iki kere derlemeniz gerekir.

LaTeX’de **article** sınıfında 4 ve 5’inci seviye başlıklara, **book** ve **report** sınıflarında ise bunlara ek 3’üncü seviye başlıklara numara verilmez ve numara verilmeyen başlıklar İçindekiler tablosuna yazılmaz. Bu seviyelerdeki başlıklara numara verilmesini ve İçindekiler tablosuna yazılması için iki adet `\setcounter` komutu

```
\setcounter{secnumdepth}{<seviye>}
\setcounter{tocdepth}{<seviye>}
```

şeklinde kullanılır. Birinci komuttaki `<seviye>` değişkeninde kaçınıcı seviyeye kadar olan başlıkların numaralandırılacağını, ikinci komuttaki `<seviye>` değişkeninde de kaçınıcı seviyeye kadar olan başlıkların İçindekiler tablosuna yazılacağını sayıyla belirtirsiniz. Örneğin **book** ve **report** sınıflarında

```
\setcounter{secnumdepth}{3}
\setcounter{tocdepth}{3}
```

komutlarıyla `\subsubsection` komutuna kadar olan başlıklara hem numara verir hem de İçindekiler tablosuna yazdırırsınız. Komutların çalışması için ya sahanlıkta ya da `\tableofcontents` komutundan önce verilmelidir.

Uzun başlıkların İçindekiler tablosunda daha kısa yazılması istenirse bölüm komutlarının zorunlu olmayan değişkenine başlıkların kısa şekli yazılır:

```
\section[Kısa Başlık]{Uzuuuuuuuuuuuuuuuun Başlık}
```

Bölüm komutlarının birde yıldızlı sürümleri vardır:

```
\part*{...}  
\chapter*{...}  
\section*{...}  
\subsection*{...}  
\subsubsection*{...}  
\paragraph*{...}  
\subparagraph*{...}
```

Komutlar bu şekilde verildiğinde başlığa numara verilmez ve İçindekiler tablosuna yazılmaz.

İçindekiler tablosunu LaTeX otomatik oluştursa da elle eklemeler yapılabilir, hatta Kaynakça gibi özel sayfalarda bu eklemeler gereklidir. Bunun için `\addcontentsline` komutu kullanılır.

```
\addcontentsline{toc}{<giriş formatı>}{<giriş metni>}
```

Burada `toc`, bilginin yazılacağı İçindekiler tablosunun dosya uzantısıdır. Bütünleşik olarak oluşturduğunuz kaynakçanın İçindekiler tablosuna yazılması için `\begin{thebibliography}` komutunun peşine `book` ve `report` sınıflarında

```
\addcontentsline{toc}{chapter}{Kaynakça}
```

`article` sınıfında ise

```
\addcontentsline{toc}{section}{Kaynaklar}
```

komutunun verilmesi gerekir.

i Not

“Kaynakça” ya da “Kaynaklar” isimleri yerine farklı isimler kullanılabilir elbette. Ancak `thebibliography` ortamının oluşturulduğu yerlerde LaTeX bu isimleri yazdıracağından tutarlı olması açısından bu isimler önerilmiştir.

```
\documentclass[a4paper,12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\title{\LaTeX'de Bölümlendirme ve İçindekiler Tablosu Oluşturma}
\author{Zafer Acar}
\begin{document}
\maketitle
\tableofcontents
\section{Birinci Seviye Başlık}
İçerik...
\subsection{İkinci Seviye Başlık}
İçerik...
\subsubsection{Üçüncü Seviye Başlık}
İçerik...
\section[Kısa Başlık]{Uzuuuuuuuun Başlık}
İçerik...
\section*{Numarasız Başlık}
\addcontentsline{toc}{section}{Numarasız Başlık}
İçerik...
\end{document}
```

2.6 Büyük Projeler

LaTeX’de kitap yazmaya başlamak için belge sınıfı `book` seçilir. Bunun dışında kitapların *Baş*, *Gövde* ve *Son* kısımları olur. Bu kısımların nerede başlayıp nerede bittikleri aşağıdaki komutlarla LaTeX’e bildirilir:

- `\frontmatter` (baş) komutu `\begin{document}` komutundan hemen sonra verilir. Bu komut, baş taraftaki İçindekiler, Önsöz gibi kısımların sayfa numaralandırmasını Roma rakamıyla yapar. Ayrıca bu kısımda bölüm komutları (*) işareti olmadan verildiğinde (örneğin `\chapter{Önsöz}`) bunlara numara verilmez ancak İçindekiler tablosuna yazılırlar.
- `\mainmatter` (gövde) komutu kitabın ilk bölüm başlığından hemen önce verilmelidir. Buradan itibaren sayfa numaralandırmasını yeniden başlatıp Arap rakamlarına geçer.

- `\appendix` (ekler) komutu kitabınızın eklerindeki bölümleri harflerle numaralandırır (Ek A, Ek B, ...).
- `\backmatter` (son) komutu kitabınızda her şey bittikten sonra verilir fakat bilinen belge sınıflarında görünürde hiçbir etkisi yoktur.

```

\documentclass[a4paper,12pt]{book}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\title{Başlık}%Başlık
\author{Yazar Adı}%yazar
\date{Eylül 2022}%Tarih

\begin{document}
\frontmatter%sayfa numaraları I,II,...
\maketitle%başlığı oluştur
\tableofcontents%içindekiler tablosu

\chapter{Önsöz}
İçerik...

\mainmatter % sayfa numaraları 1,2,3,...

\part{Birinci Kısım}

\chapter{Birinci Bölüm}
\section{Alt bölüm}
İçerik...

\appendix %eklerin başlangıcı

\part{Ekler}

\chapter{Birinci Ek}

İçerik...

\backmatter

\end{document}

```

Kitap gibi büyük hacimli belgelerle çalışırken kaynak dosyanızı parçalara ayırmak gerekebilir. LaTeX bunun için size iki komutla yardımcı olur: `\input` ve `\include`. İkisi arasındaki fark

`\include` komutuyla eklediğiniz metin yeni bir sayfadan başlayarak dizilir.

Bu komutların zorunlu değişkeni eklemek istediğiniz dosyanın adıdır. Örneğin kaynak dosyanızla aynı dizinde yer alan `dosya1.tex` dosyasını eklemek için

```
\include{dosya1}
```

komutunu kullanırsınız. Eğer dosya uzantısı `.tex` değilse (örneğin `.txt` olsun) o zaman dosya adını uzantısıyla yazmanız gerekir:

```
\include{dosya1.txt}
```

Ayrıca, hangi dosyaların eklenebileceğini LaTeX’e bildiren bir komut vardır: `\includeonly`. Bu komut, sadece sahanlığa yazılabilir. Komutun zorunlu değişkeninde eklenebilecek dosyalar aralarına virgül koyularak (ve boşluk bırakılmadan) listelenir:

```
\includeonly{dosya1,dosya2,dosya3,...}
```

Böyle bir liste oluşturulduktan sonra bu listede olmayan bir dosya artık `\include` komutuyla kaynak dosyaya eklenemez.

`\input` komutu sahanlıkta da kullanılabilir. Örneğin, sahanlığınızı tek bir dosyaya yazıp, bu dosyayı bu komutla sahanlığa ekleyebilirsiniz.

Daha düzenli çalışmak adına kaynak dosyanızın olduğu dizini de düzenleyebilirsiniz.

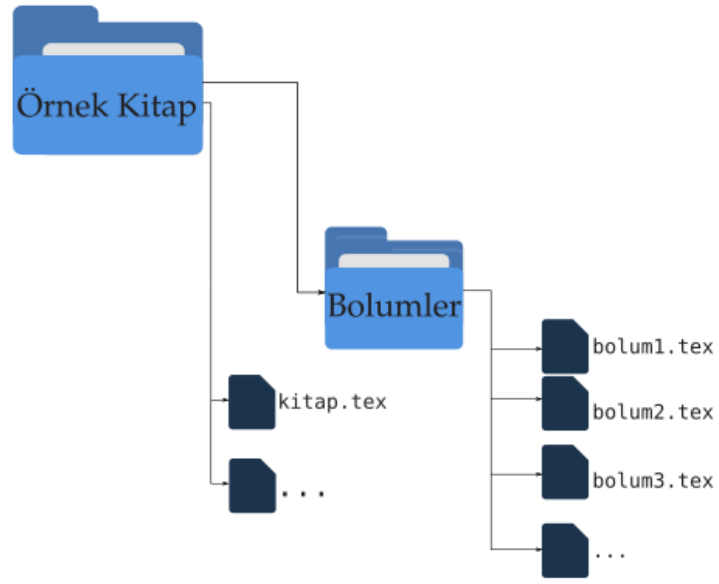
Bu şekilde bir düzenleme yaptığımızda `\input` ya da `\include` komutlarıyla dosya eklemek istediğinizde dosyanın bulunduğu dizini de göstermeniz gerekir.

Burada kaynak dosya `kitap.tex`’dir. Bu kaynak dosyaya `bolum1.tex` dosyasını eklemek istediğinizde komutu

```
\input{Bolumler/bolum1}
```

şeklinde verirsiniz. Bu sayede kaynak dosyanızın olduğu dizinde (Örnek Kitap) sadece `kitap` ile başlayan dosyalar olur. Diğer dosyalar alt dizinde (Bolumler) yer alır.

Dikkat edilirse “Örnek Kitap” dışında, “Bolumler” alt dizini ve tüm dosya adları Türkçe karakter ya da boşluk içermez.



Şekil 2.1: Kaynak dosyanın olduğu dizinin düzenlenmesi

3 Dizgi

3.1 Heceleme

Bazen tüm bu ayarlamalara rağmen LaTeX bazı kelimeleri doğru heceleyemeyebilir. Böyle durumlarda hecelemeyle elle yapmak gerekir. Yanlış hecelenen kelimenin bölünebileceği yerler `\-` komutuyla gösterilir:

```
He\ -ce\ -le\ -me
```

Bu sadece ilgili kelimenin tireyle ayrıldığı yerde doğru hecelenmesini sağlar. Aynı kelime belgenin başka bir yerinde yine yanlış hecelenebilir. Bunun yerine `\begin{document}` komutundan sonra `\hyphenation` komutuyla hece yerleri tire (-) işaretiyle gösterilmiş olan kelime listesi oluşturulursa belgenin tamamına bu kural uygulanmış olur. Örneğin

```
\hyphenation{He-ce-le-me FORTRAN}
```

komutuyla “Heceleme” kelimesinin nereden bölüneceği, “FORTRAN”, “Fortran” ya da “fortran” kelimelerinin bölünmeyeceği LaTeX’e söylenmiş olur.

4 Matematiksel İfadeler

Matematik formüllerini dizmek, kuşkusuz, LaTeX'in en güçlü olduğu konulardan biridir. Çok fazla matematiksel gösterimin varlığından dolayı da büyük bir konudur. Bu bölümde ileri bir matematik kitabını dizmek için gereken birçok şey anlatılacaktır ancak işin sınırları göz önüne alındığında başka kaynaklara da başvurmanız gerekebilir.

4.1 Giriş

Belgenizde yalnızca birkaç basit matematiksel formül kullanacaksanız herhangi bir pakete gerek olmadan yazabilirsiniz. Ancak çok sayıda karmaşık formül içeren bilimsel bir belge yazma niyetindeyseniz temel AMS paketlerini kullanmanız gerekir. Bu paketler `amsmath` , `amssymb` ve `amsfonts`'dir.

```
\usepackage{amsmath,amssymb,amsfonts}
```

Yukarıdaki komutu sahanlığa yazarak paketleri belgenize ekleyiniz. Bundan sonra bu paketleri eklediğinizi varsayarak devam edeceğiz ve bunların dışında bir pakete ihtiyaç duyarsak ayrıca belirteceğiz.

4.1.1 Genel

Belgenizin metnini oluştururken LaTeX'in metnin ne zaman matematiksel olduğunu bilmesi gerekir. Bunun nedeni, LaTeX'in matematiksel ifadeleri normal metinden farklı bir şekilde dizmesidir. Bu nedenle matematiksel ifadeler, normal metinden farklı olarak bazı ortamlarda girilirler.

Matematik özel ortamlar gerektirdiğinden, doğal olarak standart şekilde kullanabileceğiniz uygun ortam adları vardır. Bununla birlikte, diğer ortamların çoğundan farklı olarak, formülünüzü bildirmek için bazı kullanışlı kısaltmalar vardır. LaTeX'de bu ortamlar ya da kısaltmalar kullanılarak formüller iki türlü dizilir:

- Formüller satırın içinde, yani bildirildiği metnin gövdesi içine yazılır: $\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$. Görüldüğü gibi LaTeX, paragraf yapısını bozmamak için sembolleri olabildiğince sıkıştırır ve gerek görürse alttakileri yana kaydırır.

- Formüller ayrı bir satırda tek başlarına tüm detaylarıyla sergilenir:

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

Formülün satır içerisinde dizilmesi için ya `$. . . $` arasına arasına, ya `\(. . . \)` arasına ya da `\begin{math}` ile `\end{math}` arasına, yani `math` ortamında yazılması gerekir. Üçü de aynı sonucu verir.

Formülün sergilenmesi içinse ya `\[. . . \]` arasına ya `displaymath` ortamında ya da `equation` ortamında yazılması gerekir. `equation` ortamında yazılan formülü LaTeX otomatik numaralandırır. Numara verilmesini istemezseniz ortamı `equation*` şeklinde kullanmanız gerekir.

i Not

TeX'in eski sürümlerinde formüller, sergilenmeleri için `$$. . . $$` arasına yazılırdı. Bu kullanım hala geçerlidir ancak bazı sorunlara yol açabildiğinden (örneğin belge seçeneğine `fleqn` yazıldığında) kullanımı önerilmez.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}


$$x \in \mathbb{R} \text{ için } |x| < 1 \text{ ise}$$



$$-1 < x < 1$$


\end{equation*} olur.

\end{document}
```

Numara verilen bir formülü `\label` komutuyla etiketleyip, `\ref` ya da `\eqref` komutuyla formüle atıf yapılabilir. Atıf `\eqref` komutuyla yapılırsa formülün numarası parantez içinde yazılır.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{equation}
```

```

\label{eq:euler}
e^{i\pi}+1=0
\end{equation}
Euler'in \eqref{eq:euler} formülüne göre\dots

\end{document}

```

4.1.2 Matematik kipiyle metin kipi arasındaki farklar

Matematiksel ifadeleri girerken düz metin girişinden farklı olarak dikkat edilmesi gereken bazı noktalar vardır:

1. Boşlukların ve satır kesmelerinin çoğunun önemi yoktur, çünkü tüm boşluklar ya matematiksel ifadelerden mantıksal olarak türetilir, ya da özel komutlarla belirtilmesi gerekir.
2. Boş satırlara izin verilmez.
3. Aksanlı harfler kullanılmaz.
4. Her harf matematiksel bir değişken olarak kabul edilir ve italik dizilir. Eğer düz yazıyla ve normal aralıklarla bir metin girilecekse `\textrm{}` ya da `\text{}` komutları kullanılmalıdır. Bu komutlarla metin kipine geçiş yapılmış olur ve metin artık düz ve normal aralıklarla dizilir. İtalik ve normal aralıklarla metin girilecekse `\textit{}` komutu kullanılabilir. Ayrıca bu komutlarla aksanlı harfler de kullanılabilir.

Aşağıdaki örnek matematik kipi ile metin kipi arasındaki farkları gösterir.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}
 $x^2-1=0$  \text{ise  $x=\pm 1$ }.
\end{document}

```

4.1.3 Gruplandırma

Formülleri dizerken dikkat edilmesi gereken noktalardan biri komutların çoğunun kendisinden sonra ilk gelen karaktere etki etmesidir. Bu yüzden bir komutun çok sayıda karaktere etki etmesi istenirse bu karakterler iki çengelli parantez {...} arasına yazılarak gruplandırılmalıdır.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}
  $a^{x+y} \neq a^{x+y}$
\end{document}

```

4.1.4 Boşluklar

Bazen LaTeX formülleri dizerken olması gerektiği gibi boşluk bırakma konusunda yetersiz kalabilir. Bu durumda boşluklar elle oluşturulur. Boşluklar için kullanılabilecek komutlar tablodaki gibidir.

Negatif	\!
İnce	\,
Orta	\:
Kalın	\;
Sözcük arası	\
Bir quad	\quad
İki quad	\qquad

Örneğin \, komutunun bıraktığı ince boşluk bazı formüllerde çok kullanışlıdır.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\int_a^b f(x) \, dx \quad \sqrt{2} \, a
\quad \sqrt{\log x}
\]

\[
\int_a^b f(x) \, dx \quad
\sqrt{2} \, a \quad \sqrt{\, \, \log x}
\]

\end{document}

```

Negatif aralık bırakan `\!` komutu da fazla aralıklı ifadeleri birbirine yaklaştırmak için kullanılır.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
x^2/2 \quad a/\sin b
\]
\[
x^2\!/2 \quad a/\!\sin b
\]

\end{document}
```

4.2 Parantezler, Gruplandırıcılar ve Oklar

LaTeX’de her türlü parantez ve gruplandırıcı kullanılabilir. Yuvarlak ve köşeli parantezler klavyedeki yerlerinden, çengelli parantez ise `\{` ve `\}` komutları kullanılarak girilir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

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

\end{document}
```

Kullanılabilecek tüm gruplandırıcı işaretler Tablo 4.2’de gösterilmiştir.

Tablo 4.2: Gruplandırıcılar

<code>()</code>	\uparrow <code>\uparrow</code>	<code>[]</code> ya da <code>\lbrack</code>
<code>]]</code> ya da <code>\rbrack</code>	\downarrow <code>\downarrow</code>	<code>{ }</code> ya da <code>\lbrace</code>
<code>} }</code> ya da <code>\rbrace</code>	\updownarrow <code>\updownarrow</code>	<code>\langle \rangle</code> <code>\langle</code>
<code>\rangle</code>	$\ $ ya da <code>\vert</code>	<code>\lfloor</code>

Tablo 4.2: Gruplandırıcılar

\lfloor \rffloor	\lceil \lceil	\rfloor \rceil
$//$	\backslash \backslash	\Updownarrow \Updownarrow

Grup açıcı bir sembolün önüne `\left` komutu, grup kapatıcı bir sembolün önüne de `\right` komutu yazılırsa LaTeX onları en uygun boyda dizer. Her bir `\left` komutuna karşılık mutlaka bir `\right` komutu bulunmalıdır. Bunların doğru boyda dizilmesi için iki komutunda aynı satırda yer almasına dikkat edilmelidir. Sol/sağ tarafta gruplandırıcı bir işaret istenmiyorsa, görünmeyen `\left./\right.` komutu kullanılır.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\left(1+\frac{1}{n}\right)^n\quad
\left.\frac{x^3}{3}\right|_0^1
\]

\end{document}
```

Bazen gruplandırıcı sembolün boyunu elle ayarlamak gerekebilir. Bunun için, gruplandırıcı komutun önüne `\big`, `\Big`, `\bigg` veya `\Bigg` komutlarından biri verilir. `\bigl` (büyük sol) ve `\bigr` (büyük sağ) komutları da parantezleri biraz büyütür.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\big(\Big(\bigg(\Bigg(\quad
\big)\Big)\bigg)\Bigg)
\quad
\big|\Big|\bigg|\Bigg|
\]

\end{document}
```

`\end{document}`

Oklar için Tablo 4.3'deki komutlar kullanılır.

Tablo 4.3: Oklar

\leftarrow <code>\leftarrow</code> ya da <code>\gets</code>	\longleftarrow <code>\longleftarrow</code>	\uparrow <code>\uparrow</code>	\Leftarrow <code>\Leftarrow</code>
\Lleftarrow <code>\Lleftarrow</code>	\Uparrow <code>\Uparrow</code>	\rightarrow <code>\rightarrow</code> ya da <code>\to</code>	\longrightarrow <code>\longrightarrow</code>
\downarrow <code>\downarrow</code>	\Rightarrow <code>\Rightarrow</code>	\Rightarrow <code>\Rightarrow</code>	\Downarrow <code>\Downarrow</code>
\leftrightarrow <code>\leftrightarrow</code>	\longleftrightarrow <code>\longleftrightarrow</code>	\Uparrow <code>\Uparrow</code>	\Leftrightarrow <code>\Leftrightarrow</code>
\Leftrightarrow <code>\Leftrightarrow</code>	\Updownarrow <code>\Updownarrow</code>	\mapsto <code>\mapsto</code>	\longmapsto <code>\longmapsto</code>
\nearrow <code>\nearrow</code>	\hookrightarrow <code>\hookrightarrow</code>	\hookrightarrow <code>\hookrightarrow</code>	\searrow <code>\searrow</code>
\leftharpoonup <code>\leftharpoonup</code>	\rightharpoonup <code>\rightharpoonup</code>	\swarrow <code>\swarrow</code>	\leftharpoondown <code>\leftharpoondown</code>
\rightarrow <code>\rightarrow</code>	\nwarrow <code>\nwarrow</code>	\Rightarrow <code>\Rightarrow</code>	\leadsto <code>\leadsto</code>
\rightharpoondown <code>\rightharpoondown</code>		\rightleftharpoons <code>\rightleftharpoons</code>	

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\leftarrow
\downarrow
\big\downarrow
\Big\downarrow
\bigg\downarrow
\Bigg\downarrow
\rightleftharpoons

\end{document}

```

Bunların dışında altlarına ya da üstlerine matematiksel ifadeler yazılabilen `\xleftarrow` ve `\xrightarrow` komutları vardır.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\(  

\leftarrow{a}  

\rightarrow[X]{a+b}  

\)
```

```

\end{document}
```

4.3 Yunan Harfleri

Yunan harfleri matematikte yaygın olarak kullanılır. Bu harfler ters eğik çizgiden sonra harfin adı yazılarak elde edilir. Eğer ilk harf küçük ise küçük, büyükse de büyük harf elde edilir. Bazı büyük Yunanca harfler Latin harfleri gibi görüldüğünden (örneğin, büyük harf Alpha ve Beta yalnızca sırasıyla “A” ve “B”dir) ayrıca tanımlanmamışlardır. Küçük harf epsilon, theta, kappa, phi, pi, rho ve sigma iki farklı sürümde sunulmaktadır. Alternatif sürümü, harf adının önüne “var” eklenerek oluşturulur.

Tablo 4.4: Yunan Harfleri

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

```

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

```

\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

 $\forall \epsilon > 0$  için

\end{document}

```

4.4 Fonksiyonlar

LaTeX’de fonksiyonlar aşağıdaki komutlarla dizilirler.

```

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

```

Matematik kipinde fonksiyonlar diğer değişkenler gibi italik değil düz yazılırlar ve boşluklar otomatik ayarlanır.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

 $\sin x$ ,  $\exp x$ ,  $\log x$ ,
 $\det A$ ,  $\min_{x \in A} f(x)$ 

\end{document}

```

Bunların dışında bir fonksiyon tanımlamak için `\DeclareMathOperator` komutu kullanılır. `\DeclareMathOperator{\obeb}{obeb}` komutundan sonra artık kullanabileceğiniz bir “obeb” fonksiyonu olur.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\DeclareMathOperator{\obeb}{obeb}
\begin{document}

```



```
 $\backslash\text{obeb}(12,16)=4$ 
```

```
 $\end{document}$ 
```

Bu komutun sınır değerleri sağ taraf yerine alta dizen yıldızlı sürümü vardır: `\DeclareMathOperator*`. Örneğin `\DeclareMathOperator*\Max\{Max}` komutunu sahanlıkta verdikten sonra belgede kullanırsanız şöyle bir çıktı alırsınız:

```
 $\documentclass{article}$  $\usepackage[T1]{fontenc}$  $\usepackage[turkish]{babel}$  $\usepackage{amsmath,amssymb,amsfonts}$  $\DeclareMathOperator*\Max\{Max\}$  $\begin{document}$  $\begin{equation*}$  $\quad \Max_{\{x\in A\}} f(x)$  $\end{equation*}$  $\end{document}$ 
```

Modülo fonksiyonu içinse `\mod` ya da `\pmod` komutları verilir. İkinci komut fonksiyonu parantez içinde yazar.

```
 $\documentclass{article}$  $\usepackage[T1]{fontenc}$  $\usepackage[turkish]{babel}$  $\usepackage{amsmath,amssymb,amsfonts}$  $\begin{document}$  $\$a\equiv b\pmod p$  ise  $\$p\mid a-b$ 'dir. $\end{document}$ 
```

Limit için `\lim` komutu aşağıdaki şekilde verilir.

```
 $\lim_{<değişken>}\to <değişken>$ 
```

Buradaki `\to` komutu \rightarrow üretir ve ∞ için `\infty` komutu verilir.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\lim_{x \rightarrow 0}
\frac{\sin x}{x} = 1 \quad \text{\\quad}
\lim_{n \rightarrow +\infty} f_n = \delta
\]

\end{document}

```

4.5 Yığın Simgeleri

Matematikte bazen bir ifadenin altına ya da üstüne başka ifadeler yazmak gerekebilir. Bunlar yığın simgeleri olarak adlandırılırlar.

LaTeX’de aşağıdaki

```

\overset{<birinci değişken>}{<ikinci değişken>}

```

komutu birinci değişkendeki sembolü, normal boyda yazılan ikincinin üzerine yazar. `\underset` komutu ise alta yazar.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\overset{R}{\sim}
\]
\[
\underset{R}{\sim}
\]
\end{document}

```

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

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\overline{x+y}
\]
\[
\underline{x+y}
\]

\end{document}
```

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

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\underbrace{1+2+\dots+n}_{\{ \}}=
\frac{n(n+1)}{2}
+(n+1)
\]

\end{document}
```

`\overleftarrow` komutu ifadenin üstüne sola, `\overrightarrow` ise sağa bir ok çizer. Bu komutlar vektörleri göstermek için kullanılabilir. Vektörler için `\vec` komutu da kullanılır. `\stackrel{\rel}{\overset}` komutu `\overset` gibi davranır.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
```

```

\begin{document}

\[
\overrightarrow{AB} \quad \vec{a}
\]

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

\end{document}

```

4.6 Matrisler

Temel matrisler `matrix` ortamında girilir. Bu ortamda elemanlar otomatik ortalanır ve sütunlar normal bir tablo gibi dizilir. Her sütun `&` karakteriyle ayrılır ve alt satıra geçmek için `\\` komutu verilir.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\begin{matrix}
a & b & c \\
d & e & f \\
g & h & i
\end{matrix}
\]

\end{document}

```

Çeşitli matrisler dizmek için `matrix` ortamının farklı sürümleri kullanılır: `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` ve `Vmatrix`. Bu ortamlar sırasıyla yuvarlak, köşeli, çengelli, dikey çubuklu ve çift dikey çubuklu matrisler oluşturur.

```

\documentclass{article}
\usepackage[T1]{fontenc}

```

```

\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\begin{pmatrix}
1 & 2 \\
3 & 4
\end{pmatrix}
\]

\begin{bmatrix}
1 & 2 \\
3 & 4
\end{bmatrix}
\]

\begin{bmatrix}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\]

\end{document}

```

Küçük bir matris yazmak için `smallmatrix` ortamı kullanılır. Bu matriste parantezler elle eklenmelidir. Ayrıca `mathtools` paketi `psmallmatrix`, `bsmallmatrix` vb. ortamlar sağlar.

Bazı durumlarda, hizalamayı elle yapmak ve sütunlar veya satırlar arasına çizgi çekmek istenebilir. Bu durumda `tabular` ortamının matematik sürümü olan `array` ortamını kullanılmalıdır.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\left(\begin{array}{r|r}
\end{array}\right)

```

```

-1&2\\hline
3&-4
\end{array}\right)
\]

\end{document}

```

4.7 Yazı Biçem ve Boyutları

Matematiksel ifadeleri dizerken bazen yazının biçimini ya da boyutunu değiştirmek isteyebilirsiniz.

LaTeX’de matematik kipindeki yazıların biçimleri aşağıdaki komutlar kullanılarak değiştirilir.

Tablo 4.5: Matematik Kipinde Yazı Biçemleri

Komut	Görünüm
<code>\mathnormal{ABC def 123}</code>	<i>ABCdef123</i>
<code>\mathrm{ABC def 123}</code>	ABCdef123
<code>\mathit{ABC def 123}</code>	<i>ABCdef123</i>
<code>\mathbf{ABC def 123}</code>	ABCdef123
<code>\mathtt{ABC def 123}</code>	ABCdef123
<code>\mathsf{ABC def 123}</code>	ABCdef123
<code>\mathfrak{ABC def 123}</code>	<mathfrak{abcdef123}< math=""></mathfrak{abcdef123}<>
<code>\mathbb{ABC}</code>	\mathbb{ABC}
<code>\mathcal{ABC}</code>	\mathcal{ABC}
<code>\mathscr{ABC}</code>	\mathscr{ABC}

Son satırdaki komutun kullanılabilmesi için

```
\usepackage{mathrsfs}
```

komutuyla **mathrsfs** paketi eklenmiş olmalıdır.

Bu komutlarla girilen ifadelerdeki boşluklar yine dikkate alınmaz ve yine aksanlı harfler girilemez.

Matematik kipindeki bir ifadenin hem kalın hem de italik yazılması için `\boldsymbol` komutu kullanılmalıdır.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{mathrsfs}
\begin{document}

\[
\mu, \mathbf{M} \quad \mathbf{\mu}, \mathbf{M} \quad
\boldsymbol{\mu}, \boldsymbol{M}
\]

\end{document}

```

Matematik kipindeki yazının boyutunu elle ayarlayabileceğiniz dört komut vardır: `\displaystyle`, `\textstyle`, `\scriptstyle` ve `\scriptscriptstyle`. `\textstyle` komutu ifadeyi normal metin boyutunda dizer, `\displaystyle` komutu ise ifadeyi ayrı satırda sergilenir gibi büyük dizer. `\scriptstyle` ve `\scriptscriptstyle` komutları da normal metin boyutundan küçük dizerler.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{mathrsfs}
\begin{document}

\[
\sum_{k=0}^n z^k \quad \textstyle\sum_{k=0}^n z^k \quad
\displaystyle\sum_{k=0}^n z^k \quad
\scriptstyle\sum_{k=0}^n z^k \quad
\scriptscriptstyle\sum_{k=0}^n z^k
\]

\end{document}

```

Kesirler dizilirken `\displaystyle\frac{...}{...}` ve `\textstyle\frac{...}{...}` komutları yerine onların kısaltmaları olan `\dfrac` ve `\tfrac` komutları kullanılabilir. Aynı şey `\binom` komutu için de geçerlidir.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{mathrsfs}
\begin{document}


$$\frac{1}{n} \log x \quad$$


$$\frac{1}{n} \log x \quad$$


$$\binom{n}{3} \quad$$


$$\dbinom{n}{3}$$



$$\left[ \begin{array}{l} \frac{1}{n} \log x \\ \frac{1}{n} \log x \\ \binom{n}{3} \\ \binom{n}{3} \end{array} \right]$$


\end{document}

```

4.8 Denklem Ortamları

Bir satıra sığmayacak kadar uzun bir formülü ya da birden çok satırdan oluşan bir denklemi veya denklem sistemlerini hizalayıp dizmek için LaTeX’de çeşitli ortamlar kullanılır.

`multline` ortamı bir formülü hizalanmamış bir kaç satıra ayırır.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{multline}
f=a+b+c \\
+i+j+k+l \\
+x+y+z
\end{multline}

\end{document}

```


Bu ortamda ilk satır sola, son satır sağa ve kalanlar ortalı hizalanır. Denklemin numarası da son satırın sağına yazılır. Dekleme numara verilmesi istenmiyorsa ortam `multline*` şeklinde kullanılmalıdır.

`split` ortamı denklemleri dikey hizalanmış birden çok satırda dizer.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{equation}
\begin{split}
a&= b+c-d\\
&= e-f\\
&= g
\end{split}
\end{equation}

\end{document}
```

Hizalama `&` karakteriyle yapılır (genelde `=` işaretinden hemen önce kullanılır). Ortam mutlaka formülün numaralandırılmasından sorumlu ya da numara vermeyen başka bir matematik ortamında kullanılması gerekir.

`gather` ortamı birden fazla formülü birlikte gruplandırır, ortalar ve her birini ayrı bir satırda numaralandırır. Yine `gather*` ortamı, aynı türden numaralandırılmamış formüller üretir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{gather}
a=b+c \\
V+F-S=2
\end{gather}

\end{document}
```

`align` ortamı, iki veya daha fazla satırdan oluşan bir denklemi her bir satırı hizalı ve numaralı şekilde dizmek için kullanılır. Hizalama aynı şekilde `&` karakteriyle yapılır. Ortam yıldızlı (`align*`) şekilde kullanılırsa hiçbir satır numaralandırılmaz. `align` ortamı aynı zamanda birden fazla özerk formül dizisini birleştirmek için de kullanışlıdır. Bu durumda, `&` karakteri konumuna bağlı olarak hizalama ve ayırıcı olmak üzere iki farklı işlev üstlenir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{align}
a& = b+c+d \\
e& = f \\
x-1& = y+z
\end{align}
\begin{align*}
a \&=b \& c\&=d \& e\&=f \\
u \&=v \& w\&=x \& y\&=z
\end{align*}
\end{document}
```

`alignat` ortamı `align` ortamına benzer fakat sütun sayısını belirten bir değişken alır (Bir satırda kullanılan `&` sayısının bir fazlasının yarısı sütun sayısını vermelidir} ve denklemler arasındaki yatay boşluğun kontrolünü sağlar. Eğer boşluk komutlarından biri kullanılmazsa denklem sistemleri arasında boşluk bırakılmaz (örnekte boşluk komutu olarak `\qquad` kullanılmıştır).

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{alignat*}{3}
a\&=b\qquad \& c\&=d\qquad \& e\&=f \\
u\&=v\qquad \& w\&=x\qquad \& y\&=z
\end{alignat*}

\end{document}
```

`flalign` ortamı `align` ortamına benzer ancak ilk denklem sistemini sola ve son denklem sistemini sağa yaslar.

`aligned` ortamı yine `align` ortamına benzer fakat başka bir matematik ortamında kullanılması gerekir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
\left.
\begin{aligned}
a&= b+1 \\
a&= 2b
\end{aligned}
\right\}
\quad
\text{$a=2$ ve $b=1$.}
\]

\end{document}
```

`cases` ortamı parçalı fonksiyonları dizmek için kullanışlıdır. Ortamın içine yazılan denklemlerin solunda uygun boyda bir çengelli parantez açar. Sütunlar sola yaslıdır. Ortamın başka bir matematik ortamında kullanılması gerekir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\[
n!=
\begin{cases}
1 & \text{$n=0$ ise} \\
n(n-1)! & \text{$n\ge 1$ ise}
\end{cases}
\]

\end{document}
```

```
\end{document}
```

Her satıra numara veren bir ortamda bazı satırların numarasız olması istenirse bu satırların sonuna `\notag` ya da `\nonumber` komutları verilir. `\tag` komutuyla ise keyfi bir numara ya da işaret yazılabilir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{align}
x&=y\\
z&=y+1 \notag\\
w&=3 \tag{*}
\end{align}

\end{document}
```

Numaralı formüllere etiket yine `\label` komutuyla koyulur ve `\eqref` komutuyla atıf yapılır.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{align}
a &= b+c \\
c &= d \label{eq:cd}\\
e &= f+g
\end{align}
Yukarıdaki \eqref{eq:cd}
formülü\dots

\end{document}
```

Aynı ortamda yer alan formüllerin 1, 2,... yerine 1.a, 1.b,... biçiminde numaralandırılması için ortamın `subequations` ortamının içine yazılması gerekir.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\begin{document}

\begin{subequations}
\label{eq:sub}
\begin{align}
a &= b+c \\
c &= d \label{eq:subb} \\
e &= f+g
\end{align}
\end{subequations}

Formül~\eqref{eq:sub} ve
altformül~\eqref{eq:subb}\dots

\end{document}

```

4.9 Teorem ve Benzeri Ortamlar

Gerçek bir matematik kitabı karıştırdıysanız “Teorem 2.1”, “Sonuç 2.1.1” gibi ifadelerle başlayan paragraflara rastlamış olmalısınız. Bunlar metnin geri kalanından ayrılmış ve yanında sıralı sayılarla etiketli paragraflardır. Bu, matematikte teoremler için yaygın olarak kullanılır, ancak her şey için kullanılabilir.

LaTeX, herhangi bir teorem benzeri bildirimi kolayca tanımlamanıza izin veren bir komut sunar: `\newtheorem`.

4.9.1 Temel Teoremler

Öncelikle sahanlığa

```
\usepackage{amsthm}
```

komutuyla `amsthm` paketini ekleyiniz. En basit kullanım

```
\newtheorem{<ad>}{<başlık>}
```

komutunun sahanlığa verilmesidir. İlk değişken olan `<ad>`, referans olarak kullanacağınız addır, ikinci değişken `<başlık>` ise LaTeX'in her kullandığınızda yazdıracağı çıktıdır. `<ad>` değişkeni aksanlı bir harf içermemelidir. Örneğin

```
\newtheorem{tanım}{Tanım}
```

komutunu sahanlığa verdiğinizde `tanım` ortamını LaTeX'e tanıtmış olursunuz. Ayrıca kullanılan teoreme (bu örnekte Tanım) özel bir ad vermek ya da not düşmek isteyebilirsiniz. Bu, ortam komutundan sonra köşeli parantezler içinde belirtilebilir:

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{amsthm}
\newtheorem{tanım}{Tanım}
\begin{document}

\begin{tanım}
İşte yeni bir tanım.
\end{tanım}

\begin{tanım}[Gauss]
Gauss'un tanımı.
\end{tanım}

\end{document}
```

4.9.2 Sayaçlar

Sayaçlar, belge sınıfına göre varsayılan değerleri kullanır. Örneğin `book` sınıfında bir teorem kullanıldığında “Teorem 2.3” (kitabın 2’inci bölümünde yer alan 3’üncü teorem), `article` sınıfında bir teorem kullanıldığında “Teorem 3” (makaledeki 3’üncü teorem) benzeri çıktılar alınır. Varsayılan ayarları değiştirmek için sayacın takip etmesi istenilen bölüm seviyesi (chapter, section gibi) belirtilebilir:

```
\newtheorem{<ad>}{<başlık>}[<sayaç>]
```

Örneğin `article` sınıfında sahanlığa

```
\newtheorem{teorem}{Teorem}[section]
```

komutunu verdiğinizde teoreminiz `\section` başlık seviyesinin numarasına göre numara alır.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{amsthm}
\newtheorem{teorem}{Teorem}[section]
\begin{document}

\section{Teoremler}

\begin{teorem}
İşte bölüm numarasını takip eden teorem.
\end{teorem}

\end{document}
```

Varsayılan olarak, her teorem kendi sayacını kullanır. Bununla birlikte, benzer teoremlerin (örneğin Teoremler, Lemmalar ve Sonuçlar) bir sayacı paylaşması yaygındır. Bu durumda, sonraki teoremleri şöyle tanımlayın:

```
\newtheorem{<ad>}[<sayaç>]{<başlık>}
```

Burada `<sayaç>` kullanılacak olan sayacın adıdır. Genelde ana teorem adı olur. Örneğin sahanlıkta

```
\newtheorem{lemma}[teorem]{Lemma}
```

tanımlamasını yaparsanız (Bu komutu verebilmek için önceden `teorem` ortamı tanımlanmış olmalıdır) artık Lemma'lar Teorem'lerle aynı sayacı kullanacaktır.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{amsthm}
```

```

\newtheorem{teorem}{Teorem}[section]
\newtheorem{lemma}[teorem]{Lemma}
\begin{document}

\section{Teoremler}

\begin{teorem}
İşte bölüm numarasını takip eden teorem.
\end{teorem}

\begin{lemma}
İşte teoremle aynı sayacı paylaşan lemma.
\end{lemma}

\end{document}

```

`\newtheorem` komutu en fazla bir tane isteğe bağlı değişken içerebilir. Ayrıca komut `\newtheorem*` şekilde kullanılırsa numara verilmeyen bir teorem tanımlamış olur.

4.9.3 Kanıtlar

Bir teoremin kanıtı `proof` ortamında yazılır. Genel kullanım şöyledir:

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{amsthm}
\newtheorem{teorem}{Teorem}[section]
\begin{document}

\section{Teoremler}

\begin{teorem}
İşte teoremim.
\end{teorem}

\begin{proof}
İşte kanıtım.
\end{proof}

```



```
\end{document}
```

Bu ortamı kullandığınızda en sona kanıtın bittiği anlamında bir kare (QED adıyla da bilinir) ekler ve Türkçe dil paketi ekli belgelerde “Kanıt” adını yazar. Bu ad, `\proofname` komutunda saklı olup istenirse değiştirilebilir:

```
\renewcommand{\proofname}{İspat}
```

Bu komuttan sonra kullanılan tüm `proof` ortamlarında artık “Kanıt” yerine “İspat” yazar.

Ayrıca kanıtı tek seferliğine elle adlandırmak isterseniz, kendi adınızı köşeli parantezler içine yazabilirsiniz:

Kanıtın sonunu bildiren \square işareti bazen son satırda yalnız kalırsa, `\qedhere` komutuyla onu doğru yere oturtabilirsiniz:

Özel bir QED sembolü kullanmak isterseniz `\qedsymbol` komutunu yeniden tanımlayabilirsiniz.

Eğer sembolü gizlemek isterseniz `\renewcommand` komutunun son değişkenini boş bırakmanız yeterli olur.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{amsthm}
\newtheorem{teorem}{Teorem}[section]
\begin{document}

\section{Teoremler}

\begin{teorem}
  İşte teoremim.
\end{teorem}

\begin{proof}
  İşte kanıtım.
\end{proof}

\begin{proof}[İspat]
  İşte diğer ispatım.
\end{proof}
```

```

\begin{proof}
  Sadeleştirme yapılırsa  $[E=mc^2 \quad \text{qedhere}]$ 
\end{proof}

\renewcommand{\qedsymbol}{\blacksquare}
\begin{proof}
  Siyah kare.
\end{proof}

\end{document}

```

4.9.4 Teorem Stilleri

Teorem stilleri `\theoremstyle` komutuyla değiştirilir. Bu komut, `\newtheorem` komutu kullanarak tanımlanan ortamların çıktısını değiştirme olanağını verir.

```
\theoremstyle{<stil adı>}
```

Buradaki `<stil adı>` kullanmak istediğiniz stildir. Bu komuttan sonra tanımlanmış tüm teoremler bu stili kullanacaktır. LaTeX’de önceden tanımlanmış stiller aşağıdakilerdir:

Tablo 4.6: Teorem Stilleri

Stil Adı	Açıklama	Görünüm
plain	Teoremler, lemmalar, önermeler vb. için kullanılır (varsayılan)	Başlık düz ve kalın, gövde metni vurgulu
definition	Tanımlar ve örnekler için kullanılır	Başlık düz ve kalın, gövde metni düz
remark	Açıklamalar ve notlar için kullanılır	Başlık vurgulu, gövde metni düz

Örneğin sahanlığa

```

\theoremstyle{remark}
\newtheorem{notum}{Not}

```

komutlarını verip `notum` ortamını kullandığınızda başlık vurgulu, gövde metni düz olacaktır.

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{amsthm}
\theoremstyle{remark}
\newtheorem{notum}{Not}
\begin{document}

\section{Stiller}

\begin{notum}
  Buraya not aldım.
\end{notum}

\end{document}

```

4.9.4.1 Özel Stiller

Kendi stilinizi tanımlamak için `\newtheoremstyle` komutunu kullanabilirsiniz:

```

\newtheoremstyle{<stil adı>}% kullanılacak stilin adı
{<üst boşluk>}% teoremin üstünde bırakılacak boşluk. Örn:3pt.
{<alt boşluk>}% teoremin altında bırakılacak boşluk. Örn:3pt.
{<gövde yazı tipi>}% teorem gövdesinde kullanılacak yazı tipi.
  %Örn:\normalfont, \itshape...
{<girinti>}% Paragraf girintisi ölçüsü. Örn:0pt
{<başlık yazı tipi>}% teorem başlık yazı tipi.
  %Örn:\sffamily,\bfseries
{<noktalama>}% başlıktan sonraki noktalama.
  %Noktalama istenmezse boşluk bırakılabilir. Örn:\;
{<boşluk>}% başlıktan sonraki boşluk. Örn:0.25em
{<manuel başlık>}% Elle başlık belirtilir.

```

Boş bırakılan herhangi bir değişken olursa varsayılan değerler alınır. Son satırdaki `<manuel başlık>` değişkeni `\thmname`, `\thmnumber` ve `\thmnote` komutlarıyla biçimlendirilir. Birinci komut başlığı, ikinci komut numarayı, üçüncüsü ise notu biçimlendirmek içindir.

Not değişkeni her zaman isteğe bağlıdır, ancak başlık oluşturulurken `\thmnote` komutuyla belirtilmezse varsayılan olarak görünmeyecektir.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{amsmath,amssymb,amsfonts}
\usepackage{amsthm}
\newtheoremstyle{benimstilim}
{5pt}%
{5pt}%
{\normalfont}%
{} %
{\bfseries\sffamily}%
{\;}%
{0.25em}%
{\thmname{#1} \thmnumber{#2}\thmnote{---#3}.}%
\theoremstyle{benimstilim}%
\newtheorem{sonuc}{Sonuç}%

\begin{document}

\section{Özel Stiller}

\begin{sonuc}[Özel]
  Sonucu görüyor olmalısınız.
\end{sonuc}

\end{document}
```

5 Özel Sayfalar

Bu bölümde bir kitaptaki özel sayfalar olan kaynakça ve dizinden bahsedeceğiz.

5.1 Kaynakça

5.1.1 Bütünleşik Kaynakça

LaTeX’de kaynakça oluşturma’nın bir yolu, kaynakçayı kaynak dosyanızın (`.tex` uzantılı ana dosyanız) içindeki bir ortamda hazırlamaktır. Kullanacağınız ortam `thebibliography` ortamıdır.

```
\begin{thebibliography}{<sayı>}

\end{thebibliography}
```

Ortam komutundaki `<sayı>` değişkeni, kaynağın etiketi veya etiket girilmediği takdirde verilen sıra numarasının kaç karakter uzunluğunda olacağını belirtir. Örneğin ortam

```
\begin{thebibliography}{9}

\end{thebibliography}
```

şeklinde oluşturulursa etiket veya etiket girilmediği takdirde verilen sıra numarası için bir karakter uzunluğunda yer ayrılması gerektiği ve toplamda bu ortama en fazla dokuz adet kaynak girileceği belirtilmiş olur. Eğer dokuzdan fazla kaynak girilecekse, 99 kaynağa kadar izin veren “45” gibi iki basamaklı bir sayı girilebilir.

Ortama her kaynak `\bibitem` komutuyla eklenir ve komuttan sonra kaynağı tanımlayıcı bilgiler girilir. Bu bilgiler girilirken biçim elle oluşturulur.

```
\bibitem[<etiket>]{<anahtar>}
```

Komutun zorunlu değişkeni olan `<anahtar>`, ileride kaynağa atıf yapmak için kullanacağınız bir tanımlayıcıdır ve her kaynak için benzersiz olmalıdır. Genelde akılda kolay kalması için yazarın soyadı ve yayın yılı olacak şekilde düzenlenir.

Zorunlu olmayan `<etiket>` değişkeni girilmediği takdirde kaynağın önüne köşeli parantezler içinde kaynağın sıra numarası yazdırılır.

Kaynağın sıra numarasının köşeli parantezler içinde yazılması istenmezse aşağıdaki komutlarla değişiklik yapılabilir.

```
\makeatletter
\renewcommand{\@biblabel}[1]{\textbf{#1.}}
\makeatother
```

Bu komut verilirse sıra numaraları parantezsiz, kalın ve ardında nokta olacak şekilde yazılır.

Ortam genelde `\end{document}` komutundan hemen önce oluşturulur ve ortamın oluşturulduğu yerde LaTeX, `book` ve `report` sınıflarında eğer Türkçe dil paketi eklenmişse “Kaynakça”, `article` sınıfında ise “Kaynaklar” ismini ve ardından kaynakları yazdırır.

Kaynaklardan herhangi birine atıf `\cite` komutuyla yapılır.

```
\cite[<seçenekler>]{<anahtar>}
```

Komutun zorunlu değişkeni olan `<anahtar>`, atıf yapılmak istenen kaynağın `\bibitem` komutundaki zorunlu değişkenidir. İsteğe bağlı `<seçenekler>` değişkeninde ise sayfa numarası, bölüm numarası gibi fazladan vurgulanmak istenen bilgiler girilebilir.

Atıf yapılan yerde kaynağın etiketi ya da etiket girilmediği takdirde sıra numarası köşeli parantez içinde yazdırılır. Eğer fazladan yapılan vurgu varsa, bu, etiket ya da sıra numarasının devamında virgülden sonra yazdırılır.

Aynı yerde birden fazla kaynağa atıf yapılacaksa atıf yapılacak kaynakların anahtarları aralarına virgül koyularak `\cite{<anahtar1>,<anahtar2>,<anahtar3>}` şeklinde yazılır.

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\title{\LaTeX'de Kaynakça Yönetimi 1: Bütünleşik Kaynakça}
\author{Zafer Acar}

\begin{document}
\maketitle
```

WYSIWYG editörleri yerine, `\TeX/\LaTeX{}` `\cite{lamport94}` dizgi sistemini kullanmaya başlayın. Görüldüğü gibi kaynakça oluşturmak ve atıf yapmak oldukça kolaydır.

Ali Nesin, `\cite[s.~47]{nesin07}`'de pokerin matematiğini anlatıyor.

İki kaynağa birden atıf `\cite{lamport94,nesin07}` şeklinde yapılır.

```
\begin{thebibliography}{9}
\bibitem{lamport94}
Leslie Lamport,
\textit{\LaTeX: a document preparation system}, Addison Wesley,
Massachusetts, 2nd edition, 1994.

\bibitem[N]{nesin07} Ali Nesin, \textit{Matematik ve Oyun},
Nesin Yayıncılık, 2007.
\end{thebibliography}
\end{document}
```

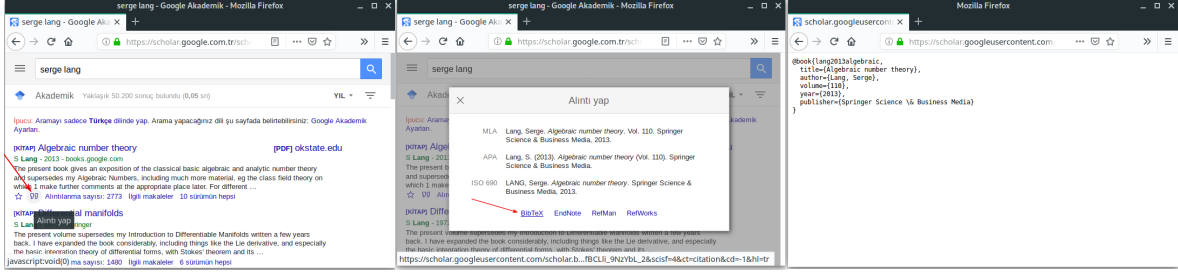
5.1.2 Kaynakçanın ayrı dosyada hazırlanması

Kaynakçayı ayrı bir dosyada hazırlayıp TeX dağıtımlarıyla hazır olarak gelen BiBTeX programıyla yazdırabiliriz.

Bu yöntemde kaynakça, uzantısı `.bib` olan ayrı bir dosyada hazırlanır. Bu dosya basit bir metin dosyası olup, metin editörü ya da LaTeX editörü kullanılarak oluşturulabilir, düzenlenebilir. Ayrıca [Mendeley](#) ya da [Jabref](#) gibi akademik referans düzenleme programlarından da yararlanılabilir.

Bu yöntemin önemli avantajları vardır:

1. Biçimlendirme otomatik yapılır. Eğer çalışmanızı yayımlayacak dergi ya da yayınevi kaynakçayı farklı bir formatta isterse her kaynağı tek tek elle biçimlendirmek zorunda kalmazsınız. Basit bir komut işinizi görür.
2. Dosyayı bir kere oluşturur ve sonra başka çalışmalarda kullanabilirsiniz.
3. [Google Akademik](#), [Google Kitaplar](#) ve [DergiPark](#) gibi platformlardan kullandığınız kaynakların BiBTeX kodunu çekebilirsiniz (bkz. Şekil @ref(fig:fig-google)).
4. Yukarıda da bahsettiğimiz gibi Mendeley ve Jabref gibi akademik atıf düzenleme programlarını kullanarak kaynakların BiBTeX kodunu oluşturabilir, düzenleyebilirsiniz.



(a) Alıntı yap

(b) BibTeX

(c) Kodu kopyala

Şekil 5.1: Google Akademik alıntı yapma

5.1.2.1 Dosyanın hazırlanması

Aşağıda .bib uzantılı bir dosya örneği gösterilmiştir.

```
@book{lang13,
  title={Algebraic number theory},
  author={Lang, Serge},
  volume={110},
  year={2013},
  publisher={Springer Science & Business Media},
}

@article{lamport78,
  title={Time, clocks, and the ordering of events in a
    distributed system},
  author={Lamport, Leslie},
  journal={Communications of the ACM},
  volume={21},
  number={7},
  pages={558--565},
  year={1978},
  publisher={ACM},
}

@manual{Oetiker06,
  author = {Oetiker, Tobias and Partl, Hubert and Hyna, Irene
    and Schlegl, Elisabeth},
  title = {İnce bir {\LaTeXe} Elkitabı veya, 116 dakikada
    {\LaTeXe}},
  note = {Türkçesi: Bekir Karaoğlu},
  url = {http://ftp.ntua.gr/mirror/ctan/info/lshort/turkish/
    lshort-tr.pdf},
}
```



```
    year    = {2006},  
}
```

Bu dosyada Serge Lang’a ait bir kitap (`@book`), Leslie Lamport’a ait bir makale (`@article`) ve LaTeX için bir teknik kılavuz (`@manual`) vardır.

Her kaynağın ilk olarak `@` işaretiyle türü belirtilir. Yukarıdakilere ek olarak rapor için `@report`, tez için `@thesis`, çevrimiçi kaynaklar için `@online` kullanılır. Bunların dışındaki birçok türe LaTeX editörlerinin menü çubuğuklarında bulunan “Kaynakça (Bibliography)” menüsünden ulaşılabilir.

İlk girdi (`lang13`, `lamport78`, `Oetiker06`) kaynağa atıf yapmak için kullanılan anahtardır. Sonrasında gelenler de tahmin edilebileceği gibi başlık (`title`), yazar (`author`), yayıncı (`publisher`), yıl (`year`), dergi (`journal`), cilt (`volume`)... gibi kaynağı tanımlayan bilgilerdir. Bu tanımlamaların her biri eşittir işaretinden sonra iki çengelli parantez arasında yapılır (çift tırnak da kullanılabilir) ve her tanımlama (sonuncusu olsa dahi) virgülle ayrılır.

Yazar adı ya

```
author={Adı Soyadı}
```

ya da

```
author={Soyadı, Adı}
```

şeklinde girilmelidir ve birden fazla yazar varsa yazarlar yukarıdaki yazımdan dolayı virgülle değil **and** ile ayrılmalıdır. Yazarları ayırmak için virgül kullanırsanız yüksek ihtimalle LaTeX, yazarların adları ve soyadlarını karıştıracaktır.

Bir diğer önemli nokta özel kelimeleri yazmak için kullanılan komutları ve aksanlı harfleri iki çengelli parantez içinde yazmaktır. Örneğin “â” için `{\^a}` yazılmalıdır. Genel olarak sorun yaşanan karakterleri iki çengelli parantez içine yazmak gerekir.

Her tür için zorunlu olarak belirtilmesi gereken bilgiler ve isteğe bağlı bilgiler vardır. Bunların ne olduklarını tahmin etmek zor değildir. Bu konuda editörden de yararlanabilirsiniz. Örneğin, `.bib` uzantılı dosyayı açıp editörde “Kaynakça → Tez” yolunu izlerseniz aşağıdaki listeyi yazdıracaktır.

```
@thesis{ID,  
  author = {author},  
  title  = {title},  
  type   = {type},  
  institution = {institution},
```

```

date = {date},
OPTsubtitle = {subtitle},
OPTtitleaddon = {titleaddon},
OPTlanguage = {language},
OPTnote = {note},
OPTlocation = {location},
OPTmonth = {month},
OPTisbn = {isbn},
OPTchapter = {chapter},
OPTpages = {pages},
OPTpagetotal = {pagetotal},
OPTaddendum = {addendum},
OPTpubstate = {pubstate},
OPTdoi = {doi},
OPTeprint = {eprint},
OPTeprintclass = {eprintclass},
OPTeprinttype = {eprinttype},
OPTurl = {url},
OPTurldate = {urldate},
}

```

Görüldüğü gibi ilk altı satır zorunlu, OPT ile başlayanlar isteğe bağlıdır. İsteğe bağlı olanlardan belirtmek istediklerinizin başında bulunan OPT'yi silip tanımlamayı yapabilirsiniz.

Editörden yararlanmanın diğer bir yolu “Kaynakça → Kaynakça Kaydı Ekle...” yolunu izlemektir. Bu yolu izlediğinizde aşağıdaki pencere açılır (örnek TeXstudio editörüne aittir).

Yeni BibLaTeX Kaydı

Dosyaya Ekle:	Kayıt Türü:	Alanlar:																																																												
<New File> mycollection.bib kaynakca.bib kaynakca.bib <dosya>.bib	Dergi İçin Yayın Kitap Çok Ciltli Kitap Kendi Başlığı Olan Bir Kitap Bölümü Kitap içinde Kitap Bir Kitaptaki Ek Materyal Kitapçık Tek Ciltli Derleme Çok Ciltli Derleme Derleme İçin Yayın Bir Derlemedeki Ek Materyal Teknik Kılavuz Diğer Çevrimiçi Kaynak Patent Sürelî Bir Yayının Tüm Sayısı Sürelî Bir Yayındaki Ek Materyal Konferans Kitabı Çok Ciltli Konferans Kaydı Konferans Kitabı İçin Yayın Başvuru Çok Ciltli Başvuru Kaydı Bir Başvurudaki Yayın Rapor Tez Yayınlanmamış	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">ID</th> <th style="width: 45%;">Alan</th> <th style="width: 50%;">Değer</th> </tr> </thead> <tbody> <tr><td>1</td><td>ID</td><td></td></tr> <tr><td>2</td><td>author</td><td></td></tr> <tr><td>3</td><td>title</td><td></td></tr> <tr><td>4</td><td>date</td><td></td></tr> <tr><td>5</td><td>editor</td><td></td></tr> <tr><td>6</td><td>editora</td><td></td></tr> <tr><td>7</td><td>editorb</td><td></td></tr> <tr><td>8</td><td>editorc</td><td></td></tr> <tr><td>9</td><td>translator</td><td></td></tr> <tr><td>10</td><td>annotator</td><td></td></tr> <tr><td>11</td><td>commentator</td><td></td></tr> <tr><td>12</td><td>introduction</td><td></td></tr> <tr><td>13</td><td>foreword</td><td></td></tr> <tr><td>14</td><td>afterword</td><td></td></tr> <tr><td>15</td><td>subtitle</td><td></td></tr> <tr><td>16</td><td>titleaddon</td><td></td></tr> <tr><td>17</td><td>maintitle</td><td></td></tr> <tr><td>18</td><td>mainsubtitle</td><td></td></tr> <tr><td>19</td><td>maintitleaddon</td><td></td></tr> </tbody> </table>	ID	Alan	Değer	1	ID		2	author		3	title		4	date		5	editor		6	editora		7	editorb		8	editorc		9	translator		10	annotator		11	commentator		12	introduction		13	foreword		14	afterword		15	subtitle		16	titleaddon		17	maintitle		18	mainsubtitle		19	maintitleaddon	
ID	Alan	Değer																																																												
1	ID																																																													
2	author																																																													
3	title																																																													
4	date																																																													
5	editor																																																													
6	editora																																																													
7	editorb																																																													
8	editorc																																																													
9	translator																																																													
10	annotator																																																													
11	commentator																																																													
12	introduction																																																													
13	foreword																																																													
14	afterword																																																													
15	subtitle																																																													
16	titleaddon																																																													
17	maintitle																																																													
18	mainsubtitle																																																													
19	maintitleaddon																																																													

☐ Boş, İsteğe Bağlı Alanlar Ekle

Pencerenin solunda kaydı eklemek istediğiniz dosyayı ve ortada kayıt türünü belirtir, sağda da kaynağın bilgilerini girersiniz. Zorunlu bilgiler en üstte yer alan kalın yazılmış olanlardır.

5.1.2.2 Kaynakçanın yazdırılması

Kaynakçayı yazdırmak için BiBTeX'i kullanacağız. BiBTeX'in LaTeX'le standart olarak geldiğini ifade etmiştik. Dolayısıyla bu programı kullanmak için ek bir şey yapmanız gerekmez.

Oluşturulan .bib uzantılı dosya `\bibliography` komutuyla içeri aktarılır, `\bibliographystyle` komutuyla da kullanılacak biçim belirtilir.

```
\bibliographystyle{<biçim>}
\bibliography{<dosya>}
```

Burada yer alan <dosya> uzantısının belirtilmesine gerek yoktur. Dosyanın `kaynakca.bib` olduğunu varsayarak, komut `\bibliography{kaynakca}` şeklinde verilir. Kullanılabilecek biçimler `abbrv`, `acm`, `alpha`, `apalike`, `ieeetr`, `plain`, `siam` ve `unsrt`'dir. Biçimlerin nasıl çıktığı verdiklerini görmek için [şuraya](#) bakabilirsiniz.

Atıf, bütünleşik kaynakçada olduğu gibi `\cite` komutuyla yapılır fakat bütünleşik kaynakçadan farklı olarak atıf yapılmayan kaynaklar yazdırılmaz. Bazı kaynakların bu kuraldan ayrı tutulması istenirse `\nocite` komutu, değişkenine kaynağın anahtarı yazılarak `\bibliography` komutundan önce verilmelidir.

```
\nocite{<anahtar>}
```

Eğer tüm kaynakların bu kuraldan ayrı tutulması isteniyorsa komut `\nocite{*}` şeklinde verilmelidir.

Kaynakçanın belgeye yazılması için kaynak dosyanın derlenip, BiBTeX programının çalıştırılması ve ardından dosyanın en az iki kere daha derlenmesi gerekir. BiBTeX programı, editörde “Araçlar → Kaynakça” yoluyla çalıştırılır (klavye kısa yolu F8). Aynı şey, uçbirimde sırasıyla

```
pdflatex kaynakdosya
bibtex kaynakdosya
pdflatex kaynakdosya
pdflatex kaynakdosya
```

komutları çalıştırılarak yapılabilir.

Aşağıda kaynak dosya örneği verilmiştir. Bu dosyayı derleyebilmeniz için içeriği yukarıda verilen `kaynakca.bib` dosyasının bu dosyayla aynı dizinde olması gerektiğini unutmayınız.

```
\documentclass[10pt,a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[turkish]{babel}
\usepackage{dtk-logos} % \BibTeX komutu için...
\title{Kaynakça Yönetimi 2: \BibTeX}
\author{Zafer Acar}
\begin{document}
\maketitle
Lang'ın kitabı \cite{lang13}, Lamport'un makalesi \cite{lamport78}
ve \LaTeX{} için Türkçe kaynak \cite{Oetiker06} \dots

\bibliographystyle{siam}
\bibliography{kaynakca}
\end{document}
```

5.2 Dizin

Bilimsel bir yapıtta bulunması gereken *dizin* ya da diğer adıyla *indeks*, bir yapıtın kiři, konu, yer adı vb. bakımından içindekileri yer numarasıyla belirten ve yapıtın arkasında yer alan alfabetik listedir.

LaTeX’de dizin oluşturabilmek için sahanlığa

```
\usepackage{makeidx}
\makeindex
```

komutları girilir. Birinci komut dizin için gerekli olan **makeidx** paketini çağırır, ikinci komut ise dizinleme komutlarını etkinleştirir.

Dizinde gösterilmek istenen madde, `\index` komutunun değışkeni olarak girilir:

```
\index{<madde>}
```

Dizin maddesi girme örnekleri aşağıda gösterilmiştir.

Örnek	Dizin Maddesi	Açıklama
<code>\index{kuvvet}</code>	kuvvet, 7	Düz madde
<code>\index{magnetik!kuvvet}</code>	magnetik, 10	‘kuvvet’in alt maddesi
<code>\index{magnetik see{kuvvet}}</code>	magnetik, <i>bkz.</i> kuvvet	Atıflı madde
<code>\index{Türev@\textsl{Türev}}</code>	<i>Türev</i> , 2	Biçimlendirilmiş madde
<code>\index{Limit@\textbf{Limit}}</code>	Limit , 7	Biçimlendirilmiş madde
<code>\index{Aç1 textbf}</code>	Aç1, 3	Biçimlendirilmiş sayfa numarası
<code>\index{İvme textit}</code>	İvme, 5	Biçimlendirilmiş sayfa numarası
<code>\index{ecole@\'ecole}</code>	école, 4	Aksanlı madde giriři

LaTeX, kaynak dosyanızı derlediğinizde bu dizin maddelerini sayfa numaralarıyla birlikte, kaynak dosyayla adı aynı fakat uzantısı `.idx` olan bir dosyaya kaydeder (bu dosyaya *ham dosya* denir). Bu dosyanın `makeindex` programından geçirilmesi gerekir. Bu editörde “Araçlar → Dizin” yoluyla yapılır. Aynı şey uçbirimde,

`makeindex` kaynakdosya

komutu girilerek yapılabilir. Dosya tekrar derlendiğinde sıralanmış dizin belgeye yazılır. Bunun için dizinin yazılması istenen yere `\printindex` komutu verilir. Bu genelde, belgenin en sonunda `\end{document}` komutundan hemen öncedir. Komutun girildiği yere LaTeX, Türkçe dil paketi ekli belgelerde “Dizin” başlığını oluşturur ve belgede `\index` komutuyla eklenmiş maddeleri sırayla dizer.

Program, ham dosyayı işleyip dizin maddelerini abece sırasına göre dizer ve `.ind` uzantılı bir dosyaya aktarır. Ancak, Türkçe aksanlı harflerle başlayan kelimeler doğru yerde yazılmazlar. Bu harflerin doğru yere yazılması için `.ind` uzantılı dosyanın metin editörüyle açılarak elle düzenlenmesi gerekir. Ardından kaynak dosya derlenir. Elle düzeltmeden sonra tekrar `makeindex` programı çalıştırılırsa `.ind` uzantılı dosya tekrar oluşturulacağı için elle yapılan düzeltmeler bozulur. O yüzden düzeltme en son yapılmalıdır.

i Not

Aksanlı harflerle başlayan kelimelerin doğru yerde yazılmaları için aksanlı madde girme komutundan faydalanılabilir. Örneğin “çekiç” kelimesinin peşine `\index{czekiç@çekiç}` komutu verilirse bu kelime doğru yerde dizilecektir. Burada yapılan sorun yaratan “ç” harfi yerine “cz” yazılmasıdır.

5.2.1 Çoklu Dizin

Birden fazla dizin oluşturmak isterseniz (örneğin biri *normal dizin* diğeri de *simgeler dizini*) **index** paketini kullanabilirsiniz. Her dizin paket eklendikten ve `\makeindex` komutu sahanlıkta verildikten sonra `\newindex` komutuyla tanıtılır.

```
\usepackage{index}
\makeindex
\newindex{normal}{ndx}{nnd}{Normal Dizin}
\newindex{simge}{sdx}{snd}{Simgeler Dizini}
```

Komutun dört değişkeni vardır. Bunlar sırasıyla, dizin adı (örnekte `normal` ve `simge`), oluşturulacak ham dosyanın uzantısı (örnekte `.ndx` ve `.sdx`), `makeindex` tarafından ham dosyanın işlenmesiyle oluşturulan dosyanın uzantısı (örnekte `.nnd` ve `.snd`) ve son olarak `\printindex` komutuyla yazdırılacak başlıktır (örnekte “Normal Dizin” ve “Simgeler Dizini”). Buradaki uzantılar varsayılan `.idx` ve `.ind` uzantılardan farklı olmalıdır.

Ardından bir kelime ya da simgeyi dizine eklemek için, eklemek istenilen dizine göre `\index` komutu köşeli parantezler içinde dizin adı belirtilerek kullanılır.

```
\index[normal]{kuvvet}
\index[simg]{F@$\vec{F}$}
```

Birinci komut, “kuvvet” kelimesini normal dizine, ikinci komut \vec{F} simgesini simgeler dizinine ekler.

Belge derlendikten sonra iki tane `\makeindex` komutu uçbirimde,

```
makeindex kaynakdosya.ndx -o kaynakdosya.nnd
makeindex kaynakdosya.sdx -o kaynakdosya.snd
```

şeklinde verilir. Belgenizde dizinlerin yazılması istenen yere de

```
\printindex[normal]
\printindex[simg]
```

komutları verilir. Ardından belge tekrar derlenerek dizinler yazdırılır.

Çoklu dizin için diğer bir seçenek **multind** paketini kullanmaktır. Görece index paketine göre daha pratiktir. Sahanlığa

```
\usepackage{multind}
\makeindex{normal}
\makeindex{simg}
```

komutlarıyla `normal` ve `simg` adında iki dizin tanımlanır. Yine dizine yazılması istenen maddeler `\index` komutundan önce çengelli parantezler içinde dizin adı belirtilerek girilir.

```
\index{normal}{kuvvet}
\index{simg}{F@$\vec{F}$}
```

Bu defa `makeindex` programı uçbirimde

```
makeindex normal
makeindex simg
```

komutlarıyla çalıştırılır. Yine `\printindex` komutları dizinlerin eklenmesi istenen yere

```
\printindex{normal}{Normal Dizin}
\printindex{simg}{Simgeler Dizini}
```

şeklinde verilir.

5.2.2 Dizinin İçindekiler tablosuna yazılması

Dizini İçindekiler tablosuna yazmak için `\printindex` komutunun peşine `book` ve `report` sınıflarında `\addcontentsline` komutu,

```
\addcontentsline{toc}{chapter}{\indexname}
```

şeklinde, `article` sınıfında ise

```
\addcontentsline{toc}{section}{\indexname}
```

şeklinde verilmelidir. Çoklu dizin oluşturulmuşsa, `book` ve `report` sınıflarında

```
\printindex{normal}{Normal Dizin}  
\addcontentsline{toc}{chapter}{Normal Dizin}  
\printindex{simge}{Simgeler Dizini}  
\addcontentsline{toc}{chapter}{Simgeler Dizini}
```

şeklinde, `article` sınıfında ise komutlardaki `chapter` yazan yere `section` yazarak verilme-
lidir.

6 Özelleştirmeler

Kaynakça

Oetiker, Tobias, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. 2006. *İnce bir LaTeX2e Elkitabı veya, 116 dakikada LaTeX2e*.