

# 315 programlama Dilleri

Yrd. Doç. Dr. Ahmet Arif AYDIN

# Ödev Sonuçları

# Geçen Haftanın Özeti

- İsimler (Names)
- Değişkenler (variables)
- Bağlanma (Bindings)
- Kapsam (Scopes)

## Data Types-1 (Veri Tipleri)

# Primitive Data Types

İlkel (basit)  
veri tipleri

- Başka veri tipleri tarafından tanımlanamazlar
- Her programlama dilinin kendine ait ilkel tipleri bulunmaktadır.
- Eski programlama dillerinde sadece sayısal veri tipleri kullanılmıştır.
- Günümüzdeki programlama dillerinde bulunan veri tipleri:
  - sayısal
  - mantıksal
  - kullanıcı tanımlı

# Primitive Data Types

İlkel (basit)  
veri tipleri

- 1- Numeric (Sayısal) Types
- 2- Boolean (mantıksal) Types
- 3- Character (karakter) Types

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)

integer

Floating-point

Complex

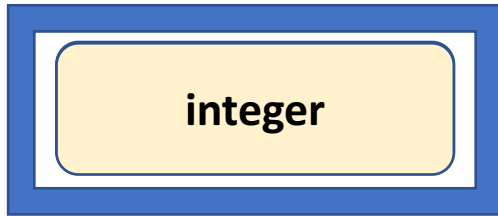
Decimal

Sayısal veri tiplerinin programlama da önemi büyüktür.

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)



Floating-point

Complex

Decimal

- Tamsayı
- Bilinen en genel sayısal veri tipidir.
- Farklı programlama dillerinde farklı uzunlukta integer tipleri bulunmaktadır.



# Primitive Data Types

İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)

integer

Floating-point

Complex

Decimal

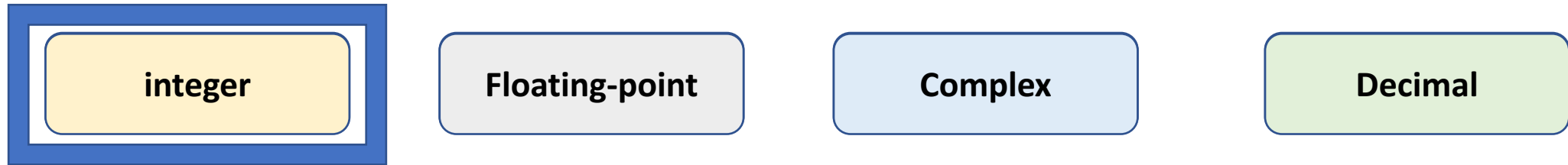
<u>byte</u>	1 byte	-128	127	From +127 to -128	byte b = 65;
<u>char</u>	2 byte	0	$2^{16}-1$	All Unicode characters	char c = 'A'; char c = 65;
<u>short</u>	2 byte	$-2^{15}$	$2^{15}-1$	+32,767 to -32,768	short s = 65;
<u>int</u>	4 byte	$-2^{31}$	$2^{31}-1$	+2,147,483,647 to -2,147,483,648	int i = 65;
<u>long</u>	8 byte	$-2^{63}$	$2^{63}-1$	From +9,223,372,036,854,775,807 -9,223,372,036,854,775,808	long l = 65L;

**JAVA da  
bulunan  
integer olarak  
kabul edilen  
veri tipleri**

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)



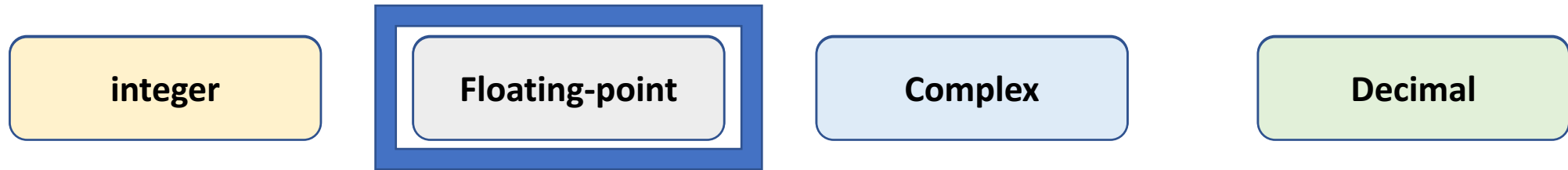
- *C*, *C++*, *C#* da işaretsiz (unsigned integer) tipi bulunmaktadır.
- Bilgisayarda tanımlanırken en soldaki birinci bit işaret biti olarak tanımlanır.



# Primitive Data Types

İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)

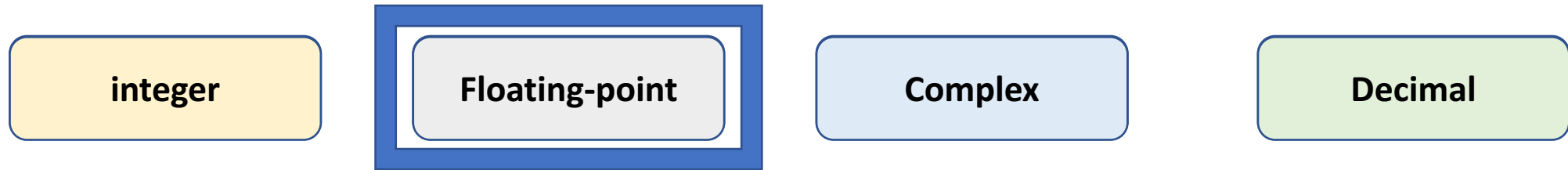


- Kayan-noktalı veri tipi gerçek sayıları modellemekte kullanılır.
- sayıların gerçeğe yakın değerlerini tanımlamak için kullanılır
- $\pi$  (Pi sayısı): 3.1415926535897932384626433832797... (100.000 digit)
- $e$  : 2.71828182845904523536028747135266

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)



- Programlama dilleri

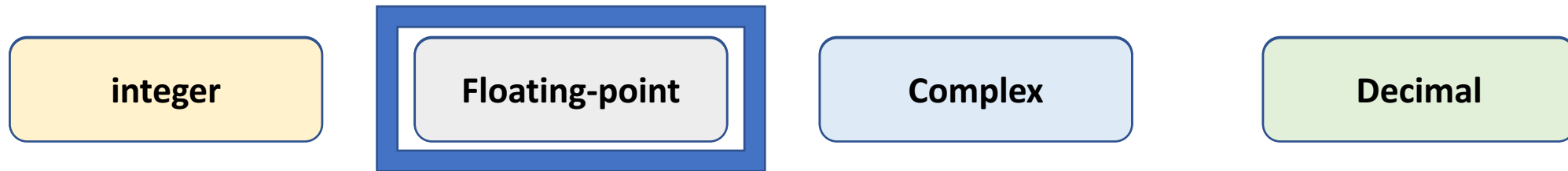
- Float (4 byte)
- Double (8 byte)

floating-point veri tiplerini içerir.

# Primitive Data Types

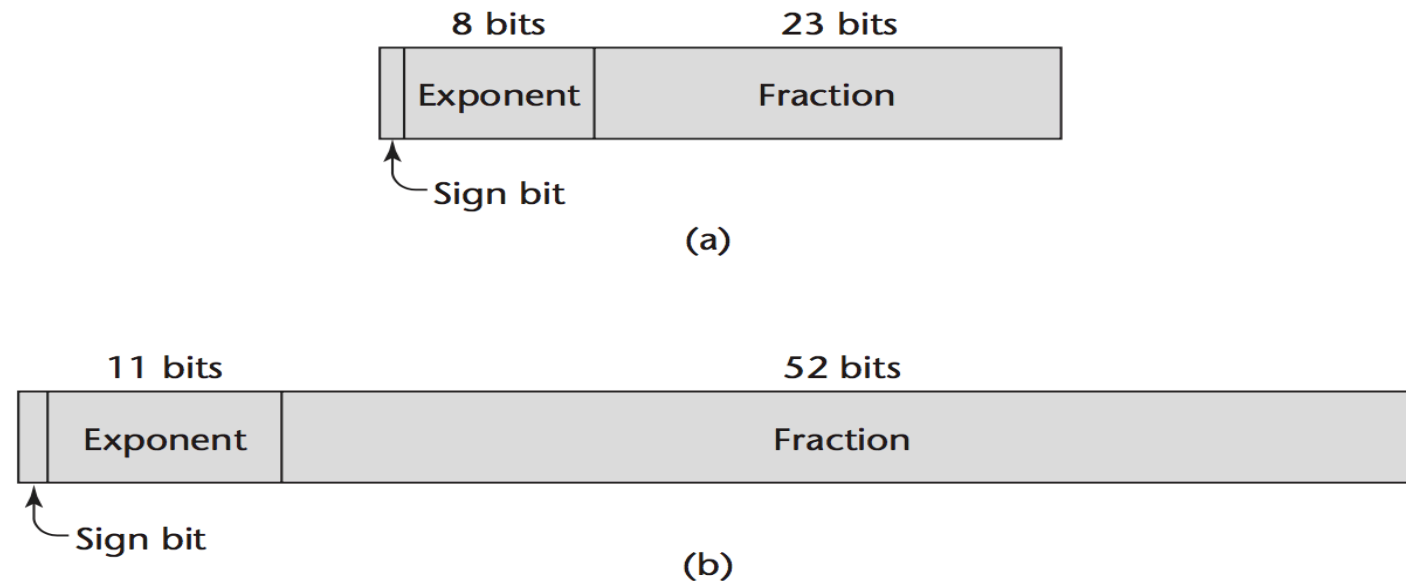
İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)



**Figure 6.1**

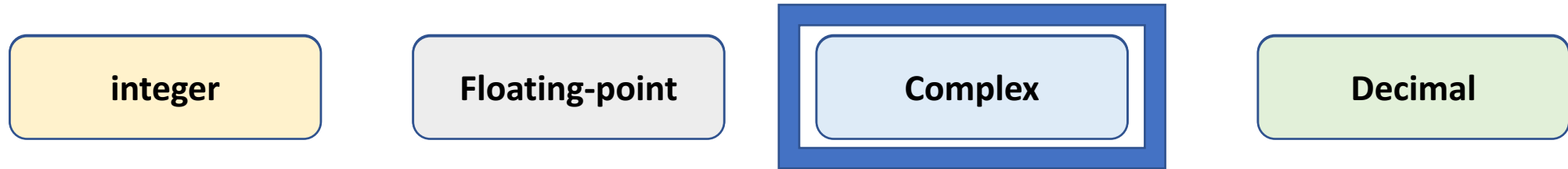
IEEE floating-point  
formats: (a) single  
precision, (b) double  
precision



# Primitive Data Types

İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)



- Bazı programlama dilleri karmaşık sayıları desteklemektedir (Fortran, Python)

```
class complex([real[, imag]])
```

Python

$5 + 3j$

Elektrik mühendisliği  
Fourier transform

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 1- Sayısal Veri Tipleri (Numeric Types)

integer

Floating-point

Complex

Decimal

- Sabit bir alanda sabit uzunlukta ondalıklı kısım bulundurmaktadır.
- Sabit uzunlukta olduğu için yapılacak işlemler güvenilirdir.
- Ticari uygulamalarda kullanılır : COBOL, C#

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 2- Boolean Types

- En basit veri tipidir.
- 1 bitlik yer kaplar fakat bellekte 1 byte'lık alanda tutulurlar

FALSE  
0

TRUE  
1



# Primitive Data Types

İlkel (basit)  
veri tipleri

## 2- Boolean Types

- Değer aralığında sadece 2 değer bulunur: 0 ve 1
- ALGOL 60, C++ da 0 ve 1 karşılaştırma sonucu olarak kullanılabilir  
fakat Java ve C# da kullanılmaz yerine TRUE kullanılması gerekir.

FALSE  
0

TRUE  
1

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 3- Character Types

Karakterler bilgisayarlarda sayısal değerler olarak tutulmaktadır.

- ASCII karakterler

<http://www.bibase.com/ascii.htm>

# ASCII Tablosu

Dec	Hx	Char	Dec	Hx	HTML	Char	Dec	Hx	HTML	Char	Dec	Hx	HTML	Char
0	0	<b>NUL</b> (null)	32	20	&#32;	<b>Space</b>	64	40	&#64;	<b>@</b>	96	60	&#96;	<b>`</b>
1	1	<b>SOH</b> (Start of heading)	33	21	&#33;	<b>!</b>	65	41	&#65;	<b>A</b>	97	61	&#97;	<b>a</b>
2	2	<b>STX</b> (Start of text)	34	22	&#34;	<b>"</b>	66	42	&#66;	<b>B</b>	98	62	&#98;	<b>b</b>
3	3	<b>ETX</b> (End of text)	35	23	&#35;	<b>#</b>	67	43	&#67;	<b>C</b>	99	63	&#99;	<b>c</b>
4	4	<b>EOT</b> (End of transmission)	36	24	&#36;	<b>\$</b>	68	44	&#68;	<b>D</b>	100	64	&#100;	<b>d</b>
5	5	<b>ENQ</b> (Enquiry)	37	25	&#37;	<b>%</b>	69	45	&#69;	<b>E</b>	101	65	&#101;	<b>e</b>
6	6	<b>ACK</b> (Acknowledge)	38	26	&#38;	<b>&amp;</b>	70	46	&#70;	<b>F</b>	102	66	&#102;	<b>f</b>
7	7	<b>BEL</b> (Bell)	39	27	&#39;	<b>'</b>	71	47	&#71;	<b>G</b>	103	67	&#103;	<b>g</b>
8	8	<b>BS</b> (Backspace)	40	28	&#40;	<b>(</b>	72	48	&#72;	<b>H</b>	104	68	&#104;	<b>h</b>
9	9	<b>TAB</b> (Horizontal tab)	41	29	&#41;	<b>)</b>	73	49	&#73;	<b>I</b>	105	69	&#105;	<b>i</b>
10	A	<b>LF</b> (NL line fd, new line)	42	2A	&#42;	<b>*</b>	74	4A	&#74;	<b>J</b>	106	6A	&#106;	<b>j</b>
11	B	<b>VT</b> (Vertical tab)	43	2B	&#43;	<b>+</b>	75	4B	&#75;	<b>K</b>	107	6B	&#107;	<b>k</b>
12	C	<b>FF</b> (NP form fd, new page)	44	2C	&#44;	<b>,</b>	76	4C	&#76;	<b>L</b>	108	6C	&#108;	<b>l</b>
13	D	<b>CR</b> (Carriage return)	45	2D	&#45;	<b>-</b>	77	4D	&#77;	<b>M</b>	109	6D	&#109;	<b>m</b>
14	E	<b>SO</b> (Shift out)	46	2E	&#46;	<b>.</b>	78	4E	&#78;	<b>N</b>	110	6E	&#110;	<b>n</b>
15	F	<b>SI</b> (Shift in)	47	2F	&#47;	<b>/</b>	79	4F	&#79;	<b>O</b>	111	6F	&#111;	<b>o</b>
16	10	<b>DLE</b> (Data link escape)	48	30	&#48;	<b>0</b>	80	50	&#80;	<b>P</b>	112	70	&#112;	<b>p</b>
17	11	<b>DC1</b> (Device control 1)	49	31	&#49;	<b>1</b>	81	51	&#81;	<b>Q</b>	113	71	&#113;	<b>q</b>
18	12	<b>DC2</b> (Device control 2)	50	32	&#50;	<b>2</b>	82	52	&#82;	<b>R</b>	114	72	&#114;	<b>r</b>
19	13	<b>DC3</b> (Device control 3)	51	33	&#51;	<b>3</b>	83	53	&#83;	<b>S</b>	115	73	&#115;	<b>s</b>
20	14	<b>DC4</b> (Device control 4)	52	34	&#52;	<b>4</b>	84	54	&#84;	<b>T</b>	116	74	&#116;	<b>t</b>
21	15	<b>NAK</b> (Negative acknowledge)	53	35	&#53;	<b>5</b>	85	55	&#85;	<b>U</b>	117	75	&#117;	<b>u</b>
22	16	<b>SYN</b> (Synchronous idle)	54	36	&#54;	<b>6</b>	86	56	&#86;	<b>V</b>	118	76	&#118;	<b>v</b>
23	17	<b>ETB</b> (End of trans. block)	55	37	&#55;	<b>7</b>	87	57	&#87;	<b>W</b>	119	77	&#119;	<b>w</b>
24	18	<b>CAN</b> (Cancel)	56	38	&#56;	<b>8</b>	88	58	&#88;	<b>X</b>	120	78	&#120;	<b>x</b>
25	19	<b>EM</b> (End of medium)	57	39	&#57;	<b>9</b>	89	59	&#89;	<b>Y</b>	121	79	&#121;	<b>y</b>
26	1A	<b>SUB</b> (Substitute)	58	3A	&#58;	<b>:</b>	90	5A	&#90;	<b>Z</b>	122	7A	&#122;	<b>z</b>
27	1B	<b>ESC</b> (Escape)	59	3B	&#59;	<b>;</b>	91	5B	&#91;	<b>[</b>	123	7B	&#123;	<b>{</b>
28	1C	<b>FS</b> (File separator)	60	3C	&#60;	<b>&lt;</b>	92	5C	&#92;	<b>\</b>	124	7C	&#124;	<b> </b>
29	1D	<b>GS</b> (Group separator)	61	3D	&#61;	<b>=</b>	93	5D	&#93;	<b>]</b>	125	7D	&#125;	<b>}</b>
30	1E	<b>RS</b> (Record separator)	62	3E	&#62;	<b>&gt;</b>	94	5E	&#94;	<b>^</b>	126	7E	&#126;	<b>~</b>
31	1F	<b>US</b> (Unit separator)	63	3F	&#63;	<b>?</b>	95	5F	&#95;	<b>_</b>	127	7F	&#127;	<b>DEL</b>

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 3- Character Types

Bilgisayarların dünyanın dört bir yanına bağlanması ile ASCII karakterleri ihtiyacı karşılanamamıştır ve ISO 8859-1 tablosu ortaya çıkmıştır.(256 karakter)

# ISO 8859-1 tablosu

[http://www.brescianet.com/appunti/vari/  
tabellaascii/iso-8859-  
1.gif](http://www.brescianet.com/appunti/vari/tabellaascii/iso-8859-1.gif)

ISO-8859-1																
	—0	—1	—2	—3	—4	—5	—6	—7	—8	—9	—A	—B	—C	—D	—E	—F
0—	NUL 0000 0	SOH 0001 1	STX 0002 2	ETX 0003 3	EOT 0004 4	ENQ 0005 5	ACK 0006 6	BEL 0007 7	BS 0008 8	HT 0009 9	LF 000A 10	VT 000B 11	FF 000C 12	CR 000D 13	SO 000E 14	SI 000F 15
1—	DLE 0010 16	DC1 0011 17	DC2 0012 18	DC3 0013 19	DC4 0014 20	NAK 0015 21	SYN 0016 22	ETB 0017 23	CAN 0018 24	EM 0019 25	SUB 001A 26	ESC 001B 27	FS 001C 28	GS 001D 29	RS 001E 30	US 001F 31
2—	SP 0020 32	! 0021 33	" 0022 34	# 0023 35	\$ 0024 36	% 0025 37	& 0026 38	' 0027 39	( 0028 40	) 0029 41	* 002A 42	+ 002B 43	, 002C 44	— 002D 45	. 002E 46	/ 002F 47
3—	0 0030 48	1 0031 49	2 0032 50	3 0033 51	4 0034 52	5 0035 53	6 0036 54	7 0037 55	8 0038 56	9 0039 57	: 003A 58	; 003B 59	< 003C 60	= 003D 61	> 003E 62	? 003F 63
4—	@ 0040 64	A 0041 65	B 0042 66	C 0043 67	D 0044 68	E 0045 69	F 0046 70	G 0047 71	H 0048 72	I 0049 73	J 004A 74	K 004B 75	L 004C 76	M 004D 77	N 004E 78	O 004F 79
5—	P 0050 80	Q 0051 81	R 0052 82	S 0053 83	T 0054 84	U 0055 85	V 0056 86	W 0057 87	X 0058 88	Y 0059 89	Z 005A 90	[ 005B 91	\ 005C 92	] 005D 93	^ 005E 94	_ 005F 95
6—	` 0060 96	a 0061 97	b 0062 98	c 0063 99	d 0064 100	e 0065 101	f 0066 102	g 0067 103	h 0068 104	i 0069 105	j 006A 106	k 006B 107	l 006C 108	m 006D 109	n 006E 110	o 006F 111
7—	p 0070 112	q 0071 113	r 0072 114	s 0073 115	t 0074 116	u 0075 117	v 0076 118	w 0077 119	x 0078 120	y 0079 121	z 007A 122	{ 007B 123	 007C 124	}	~ 007E 126	DEL 007F 127
8—	PAD 0080 128	HOP 0081 129	BPH 0082 130	NBH 0083 131	IND 0084 132	NEL 0085 133	SSA 0086 134	ESA 0087 135	HTS 0088 136	HTJ 0089 137	VTIS 008A 138	PLD 008B 139	PLU 008C 140	RI 008D 141	SS2 008E 142	SS3 008F 143
9—	DCS 0090 144	FU1 0091 145	FU2 0092 146	STS 0093 147	CCH 0094 148	MW 0095 149	SPA 0096 150	EPA 0097 151	SOS 0098 152	SGCI 0099 153	SCI 009A 154	CSI 009B 155	ST 009C 156	OSC 009D 157	FM 009E 158	APC 009F 159
A—	NBSP 00A0 160	ı 00A1 161	ć 00A2 162	£ 00A3 163	¤ 00A4 164	¥ 00A5 165	! 00A6 166	Š 00A7 167	ˆ 00A8 168	© 00A9 169	ª 00AA 170	« 00AB 171	¬ 00AC 172	ŠHY 00AD 173	® 00AE 174	— 00AF 175
B—	° 00B0 176	± 00B1 177	² 00B2 178	³ 00B3 179	´ 00B4 180	µ 00B5 181	¶ 00B6 182	· 00B7 183	¸ 00B8 184	¹ 00B9 185	º 00BA 186	» 00BB 187	¼ 00BC 188	½ 00BD 189	¾ 00BE 190	¿ 00BF 191
C—	À 00C0 192	Á 00C1 193	Â 00C2 194	Ã 00C3 195	Ä 00C4 196	Å 00C5 197	Æ 00C6 198	Ç 00C7 199	È 00C8 200	É 00C9 201	Ê 00CA 202	Ë 00CB 203	Ì 00CC 204	Í 00CD 205	Î 00CE 206	Ï 00CF 207
D—	Ð 00D0 208	Ñ 00D1 209	Ò 00D2 210	Ó 00D3 211	Ô 00D4 212	Õ 00D5 213	Ö 00D6 214	× 00D7 215	Ø 00D8 216	Ù 00D9 217	Ú 00DA 218	Û 00DB 219	Ü 00DC 220	Ý 00DD 221	Þ 00DE 222	ß 00DF 223
E—	à 00E0 224	á 00E1 225	â 00E2 226	ã 00E3 227	ä 00E4 228	å 00E5 229	æ 00E6 230	ç 00E7 231	è 00E8 232	é 00E9 233	ê 00EA 234	ë 00EB 235	ì 00EC 236	í 00ED 237	î 00EE 238	ï 00EF 239
F—	ø 00F0 240	ñ 00F1 241	ò 00F2 242	ó 00F3 243	ô 00F4 244	õ 00F5 245	ö 00F6 246	÷ 00F7 247	ø 00F8 248	ù 00F9 249	ú 00FA 250	û 00FB 251	ü 00FC 252	ý 00FD 253	þ 00FE 254	ÿ 00FF 255
	—0	—1	—2	—3	—4	—5	—6	—7	—8	—9	—A	—B	—C	—D	—E	—F

# Primitive Data Types

İlkel (basit)  
veri tipleri

## 3- Character Types

- 1991 yılında UCS-2 standardı oluşturulmuştur ( 16-bit karakter seti)
- 2000 yılında ISO 4-byte lık karakter kodunu geliştirmiştir
  - UCS-4, UTF-32 (ISO/IEC 10646 Standard)

# Character String Types

Karakter string tipi

birbirini takip eden karakterlerden (karakter dizilerinden) oluşur.

Eğer string ilkel tip olarak tanımlanmamışsa (Fortan 77, Basic)

string verisi birbirini takip eden karakter dizisi olarak kaydedilir (C ve C++).

`char ifade [] = "program";`        `ifade = ['p', 'r', 'o', 'g', 'r', 'a', 'm']`

# Character String Types

**C++**    `char s1[]="programlama";`  
          `char s2[40];`

`strcpy (s2, s1)`

s1'i s2'ye kopyala

`strlen (s1)`

Stringin uzunlugunu ver

`strcat (s1," dilleri")`

s1 in sonuna ekler

`strcmp (s1, s2)`

İki stringi karşılaştırır

`s1 ="programlama";`

`s2 ="programlama";`

`s1="programlama dilleri";`

`s2="programlama";`



# Character String Types

## JAVA

- String class bulunmaktadır
- charAt()
- substring (int beginIndex, int endIndex)
- replace (char oldChar, char newChar)
- trim ()
- boolean contains (CharSequence s)

# Character String Types

Bazı programlama dillerinin içerisinde (built-in)

örüntü eşleştirme (pattern matching) fonksiyonları bulunmaktadır

Perl, JavaScript, Ruby, ve PHP

Regular expressions ile örüntü bulma işlemi gerçekleştirilir.

C++, Java, Python, C#

# Character String Types

Regular Expressions

`/[A-Za-z][A-Za-z\d]+/`

Programlama dillerindeki isim  
oluşturma kuralı

# Character String Types

String uzunluğu üç biçimde tanımlanır

## 1. static

- Python, Java, C++ standard string sınıfları

## 2. Sınırlı dinamik (limited dynamic)

- C ve C++ ın izin verdiği maximum uzunluğa kadar string uzunlugu artabilir (1 byte)

## 3. Dinamik (sınırsız)

- Javascript

# Character String Types

**Figure 6.2**

Compile-time descriptor  
for static strings

Static string
Length
Address

**Figure 6.3**

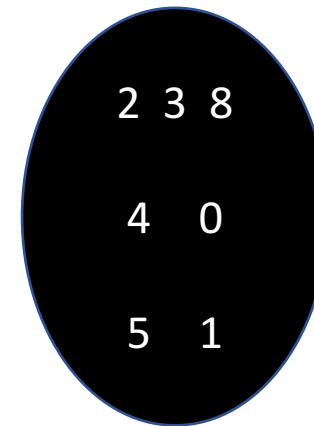
Run-time descriptor for  
limited dynamic strings

Limited dynamic string
Maximum length
Current length
Address

# User-Defined Ordinal Types

Kullanıcı Tanımlı Sıralı Tipler

1. Enumeration (liste) Types
2. Subrange (alt alan) types



Değer kümesi

Belirlenmiş bir değer kümesi içinden seçim yapılır

Yeni bir tip oluşturmayı sağlar

# User-Defined Ordinal Types

Kullanıcı Tanımlı Sıralı Tipler

## 1- Enumeration (liste) Types

- Gerçek hayattaki tipleri bir listeye dönüştürüp her bir değere index (0, 1,2,3 .. 6) değerleriyle erişim sağlanır.
- Belirlenen tip parantez içindeki değerlerle sınırlıdır.
- Bir döngü yardımıyla değerlere erişilebilir.

# User-Defined Ordinal Types

Kullanıcı Tanımlı Sıralı Tipler

## 1- Enumeration (liste) Types

```
enum günler {Mon, Tue, Wed, Thu, Fri, Sat, Sun};
```

C#

```
enum renkler {red, blue, green, yellow, black};  
renkler renk1= blue, renk12= red;
```

C++



# User-Defined Ordinal Types

Kullanıcı Tanımlı Sıralı Tipler

2- Subrange (alt alan) Types

Bir birini takip eden sıralı tipin belirlenen alt kümesidir.

**A..Z , 1..20 , 50..100 , a..z**

# User-Defined Ordinal Types

Kullanıcı Tanımlı Sıralı Tipler

## 2- Subrange (alt alan) Types

Bir birini takip eden sıralı tipin belirlenen alt kümesidir.

**A..Z , 1..20 , 50..100 , a..z**

```
type Days is (Mon, Tue, Wed, Thu, Fri, Sat, Sun);  
subtype Weekdays is Days range Mon..Fri;  
subtype Index is Integer range 1..100;
```

ADA

# Array Types (Diziler)

Aynı tipden oluşan homojen verilerin oluşturduğu sıralı yapıdır.

Index yardımıyla dizinin elemanlarına erişim sağlanır.

*C, C++, Java, Ada, ve C#* gibi diller de  
diziye kaydedilecek elemanların tipi aynı olmak zorundadır.

JavaScript, Python, ve Ruby  
farklı tipdeki elemanları aynı dizi içinde barındırır.

# Array Types (Diziler)

- İndex yardımıyla dizi elemanlarına erişim sağlanır
- Syntax:
  - Diziadi ( index değeri) -> istenilen eleman
  - []: Bir çok programlama dili
  - () : Fortran, Ada

# Array Types (Diziler)

## Dizi index tipleri

- Integer (C , Java )
- Pascal : her türlü sıralı tip
  - Integer
  - Boolean
  - Char
  - enum

# Array Types (Diziler)

- Static array (efficiency)
  - index sınırları ve bellek yeri runtime dan önce hesaplanır.
- Fixed stack-dynamic array(efficiency)
  - index belirlenir bellek bağlanması runtime da gerçekleşir.
  - C ve Java nın statik olmayan değişkenleri
- Stack-dynamic array
  - Index sınırları ve bellek bağlanması dinamikdir.
- Fixed heap-dynamic array
  - Index sınırları ve bellek bağlanması dinamikdir. Bir defa belirlendikten sonra değişmez.
  - C , C++ , Java
- Heap-dynamic array
  - Index aralığı ve bellek bağlanması dinamikdir (Javascript,Python, Ruby)

# Array Types (Diziler)

C# (heap dynamic)

```
List<String> stringList = new List<String>();  
stringList.Add("yeni");
```

# Array Types (Diziler)

## Ruby

```
dizi= Array.new(['a', 'b', 'c'])
```

```
dizi[-2]
```

```
dizi.insert(0, 1)
```

```
dizi.insert(-1, 'd')
```

```
dizi.slice(1, 2)
```

```
dizi.push('f')
```

```
dizi.pop
```

```
dizi.delete_at(2)
```



# Array Types (Diziler)

Bazı programlama dilleri dizilere ilk değer atamaya izin verir.

```
int list [] = {4, 5, 7, 83};
```

*C, C++, Java, and C#*

# Array Types (Diziler)

Bazı programlama dilleri dizilere ilk değer atamaya izin verir.

```
int list [] = {4, 5, 7, 83};
```

*C, C++, Java, and C#*

```
char isim[] = "programlama";
```

```
char *names [] = {"Bob", "Jake", "Darcie"};
```

*C ve C++*

# Array Types (Diziler)

Bazı programlama dilleri dizilere ilk değer atamaya izin verir.

```
int list [] = {4, 5, 7, 83};
```

*C, C++, Java, and C#*

```
char isim[] = "programlama";
```

```
char *names [] = {"Bob", "Jake", "Darcie"};
```

*C ve C++*

```
String[] names = ["Bob", "Jake", "Darcie"];
```

**Java**

# Array Types (Diziler)

Programlama dilleri diziler üzerinde farklı işlemlere izin verirler.

Python

```
myList =[1,2,3,4,5,6]
```

```
myList [:]
```

```
myList [:5]
```

```
myList [2:5]
```

**slice**

# Array Types (Diziler)

Çok boyutlu diziler matrislerin tanımlanmasında kullanılır.

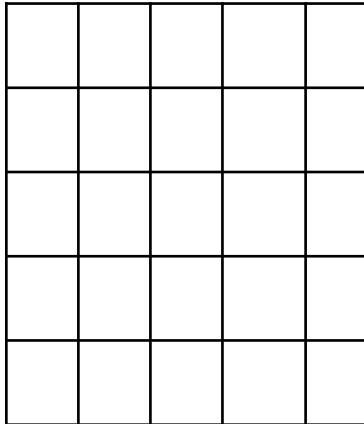
```
matris= [[1, 2, 3],[4, 5, 6],[7, 8, 9]]
```

python

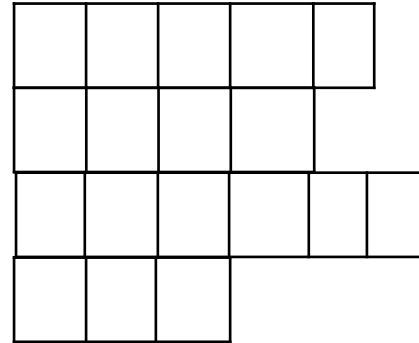
```
matris [i] [j]
```

# Array Types (Diziler)

5 x 5



**Düzenli (rectengular) dizi**



**Düzensiz (jagged) dizi**

*C, C++, java , python destekler*

# Array Types (Diziler)

Tek boyutlu dizi

Array
Element type
Index type
Index lower bound
Index upper bound
Address

Derleme zamanındaki  
Tek boyutlu dizi tanımı

Multidimensioned array
Element type
Index type
Number of dimensions
Index range 0
⋮
Index range $n - 1$
Address

Derleme zamanındaki  
çok boyutlu dizi tanımı