

Programlama Dilleri (315)

Dr. Öğr. Üyesi Ahmet Arif AYDIN

Veri Tipleri -1

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

Basit Veri Tipleri (primitive data types)

- 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

Basit Veri Tipleri (primitive data types)

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

Basit Veri Tipleri (primitive data types)

1- Sayısal Veri Tipleri (Numeric Types)

integer

Floating-point

Complex

Decimal

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

Basit Veri Tipleri (primitive data types)

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 ve integer olarak kabul edilen veri tipleri

Basit Veri Tipleri (primitive data types)

1- Sayısal Veri Tipleri (Numeric Types)

integer

Floating-point

Complex

Decimal

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

0 = positive
1 = negative

İşaret
(1bit)

Değer



Basit Veri Tipleri (primitive data types)

1- Sayısal Veri Tipleri (Numeric Types)

integer

Floating-point

Complex

Decimal

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

Basit Veri Tipleri (primitive data types)

1- Sayısal Veri Tipleri (Numeric Types)

integer

Floating-point

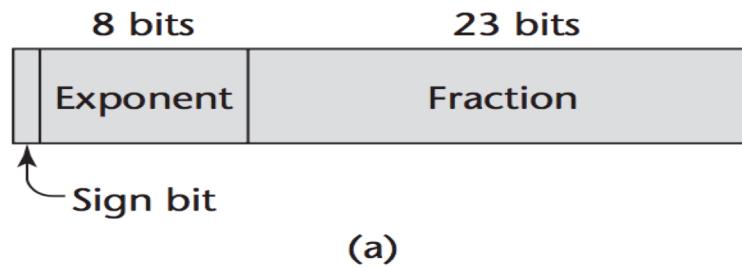
Complex

Decimal

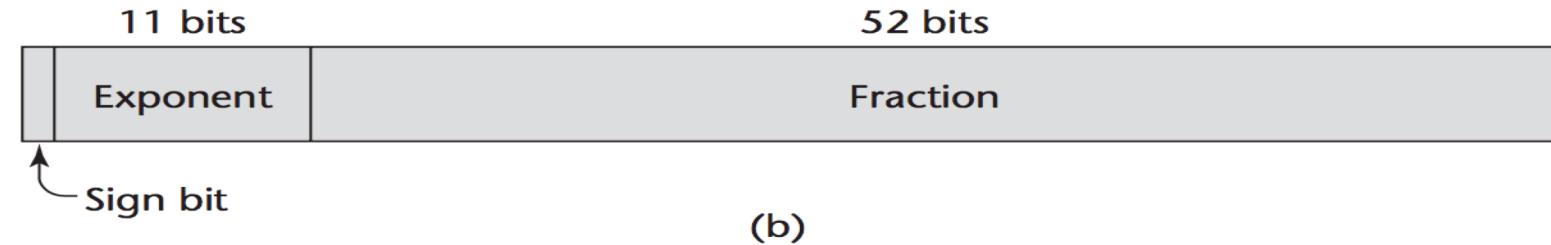
Programlama dilleri Float (4 byte) Double (8 byte) floating-point veri tiplerini içerir.

Figure 6.1

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



(a)



(b)

Basit Veri Tipleri (primitive data types)

1- Sayısal Veri Tipleri (Numeric Types)

integer

Floating-point

Complex

Decimal

Bazı programlama dilleri karmaşık sayıları desteklemektedir (Fortran, Python)

class complex([real[, imag]])

Python

$5 + 3j$

Elektrik muhendisliği
Fourier transform

Basit Veri Tipleri (primitive data types)

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#

Basit Veri Tipleri (primitive data types)

2- Boolean Types

FALSE
0

TRUE
1

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

Basit Veri Tipleri (primitive data types)

3- Character Types

Karakterler bilgisayarlarda sayısal değerler olarak tutulmaktadır. ASCII karakterler

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

Basit Veri Tipleri (primitive data types)

3- Character Types

ISO-8859-1

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

	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>
8-	U+0080 PAD 128	U+0081 HOP 129	U+0082 BPH 130	U+0083 NBH 131	U+0084 IND 132	U+0085 NEL 133	U+0086 SSA 134	U+0087 ESA 135	U+0088 HTS 136	U+0089 HTJ 137	U+008A LTS 138	U+008B PLD 139	U+008C PLU 140	U+008D RI 141	U+008E SS2 142	U+008F SS3 143
9-	U+0090 DCS 144	U+0091 PU1 145	U+0092 PU2 146	U+0093 STS 147	U+0094 CCH 148	U+0095 MW 149	U+0096 SPA 150	U+0097 EPA 151	U+0098 SOS 152	U+0099 SGCI 153	U+009A SCI 154	U+009B CSI 155	U+009C ST 156	U+009D OSC 157	U+009E PM 158	U+009F APC 159
A-	U+00A0 NBSP 160	U+00A1 i	U+00A2 ¢	U+00A3 £	U+00A4 ¤	U+00A5 ¥	U+00A6 ؋	U+00A7 ،	U+00A8 ؍	U+00A9 ؎	U+00AA ؏	U+00AB ؐ	U+00AC ؑ	U+00AD ؒ	U+00AE ؔ	U+00AF ؕ
B-	U+00B0 o	U+00B1 ±	U+00B2 2	U+00B3 3	U+00B4 '	U+00B5 μ	U+00B6 ¶	U+00B7 .	U+00B8 ,	U+00B9 1	U+00BA ؓ	U+00BB ؔ	U+00BC ؖ	U+00BD ؖ	U+00BE ؖ	U+00BF ؖ
c-	U+00C0 À	U+00C1 Á	U+00C2 Â	U+00C3 Ã	U+00C4 Ä	U+00C5 Å	U+00C6 Æ	U+00C7 Ç	U+00C8 È	U+00C9 É	U+00CA Ê	U+00CB Ë	U+00CC Ì	U+00CD Í	U+00CE Î	U+00CF Ï
D-	U+00D0 Đ	U+00D1 Ñ	U+00D2 Ò	U+00D3 Ó	U+00D4 Ô	U+00D5 Õ	U+00D6 Ö	U+00D7 ×	U+00D8 Ø	U+00D9 Ù	U+00DA Ú	U+00DB Û	U+00DC Ü	U+00DD Ý	U+00DE Þ	U+00DF ß
E-	U+00E0 à	U+00E1 á	U+00E2 â	U+00E3 ã	U+00E4 ä	U+00E5 å	U+00E6 æ	U+00E7 ç	U+00E8 è	U+00E9 é	U+00EA ê	U+00EB ë	U+00EC ì	U+00ED í	U+00EE î	U+00EF ï
F-	U+00F0 đ	U+00F1 ñ	U+00F2 ò	U+00F3 ó	U+00F4 ô	U+00F5 õ	U+00F6 ö	U+00F7 ÷	U+00F8 ø	U+00F9 ù	U+00FA ú	U+00FB û	U+00FC ü	U+00FD ý	U+00FE þ	U+00FF ÿ

Basit Veri Tipleri (primitive data types)

3- Character Types

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

Karakter String Tipleri (Character String Types)

Birbirini takip eden karakterler dizilerinden oluşan tipdir.

Eğer string ilkel tip olarak tanımlanmamışsa (Fortran 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']`

Karakter String Tipleri (Character String Types)

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    char soru1[] = "İsmin nedir? ";
    string soru2 = "Yaşadığın ükle ?";
    char cevap1 [80];
    string cevap2;
    cout << soru1;
    cin >> cevap1;
    cout << soru2;
    cin >> cevap2;
    cout << "Merhaba, " << cevap1;
    cout << " yaşadığın ülke" << cevap2 << "!\n";
    return 0;
}
```

Karakter String Tipleri (Character String Types)

```
char s1[]="programlama";  
char s2[40];
```

C++

strcpy (s2, s1)

s1'i s2'ye kopyala

s1 ="programlama";

s2 ="programlama";

strcat (s1," dilleri")

s1 in sonuna ekler

s1="programlama dilleri";

s2="programlama";

strlen (s1)

Stringin uzunlugunu ver

Karakter String Tipleri (Character String Types)

The screenshot shows a Java code editor interface. At the top, there's a navigation bar with icons for trinket, Java beta, Run, Share, and Remix. The main area has tabs for Main.java and a file icon. Below the tabs is a code editor with line numbers (1-12) and syntax highlighting. The code prints the string "Programlama Dilleri 315" and its 6th character ('a').

```
public class Main {
    public static void main(String[] args) {
        String str1 = "Programlama Dilleri 315";
        // String ifadesini karakter dizisine cevir
        char[] karakterDizisi = str1.toCharArray();
        System.out.println(karakterDizisi);
        System.out.println(karakterDizisi[5]);
    }
}
```

On the right side, there's a sidebar with the word "JAVA" in bold, the trinket logo, and text about programming languages (Programlama Dilleri 315). Below that is a link to a w3resource exercise page.

Powered by trinket
Programlama Dilleri 315
a

<https://www.w3resource.com/java-exercises/string/java-string-exercise-28.php>

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

Karakter String Tipleri (Character String Types)

Bazı programlama dillerinin içerisinde (built-in) örüntü eşleştirme (pattern matching) fonksiyonları bulunmaktadır (JavaScript, Ruby, Perl ve PHP)

Regular expresions ile örüntü bulma işlemi gerçekleştirilir.

C++, Java, Python, C#

Karakter String Tipleri (Character String Types)

Regular Expressions

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

Programlama dillerindeki
isim oluşturma kuralı

Karakter String Tipleri (Character String Types)

Programla dillerinde String uzunluğu üç biçimde tanımlanır

1. Static

- Python, Java, C++ standard string sınıfları

2. Sınırlı dinamik (limited dynamic)

- C ve C++'in izin verdiği maximum uzunluğa kadar string uzunluğu artabilir (1 byte)

3. Dinamik (sınırsız)

- Javascript

Karakter String Tiplerinin (Character String Types)

Figure 6.2

Compile-time descriptor
for static strings

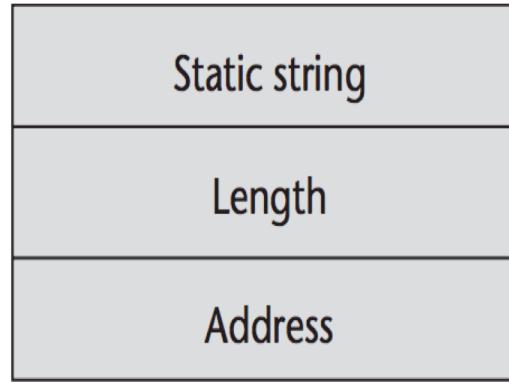
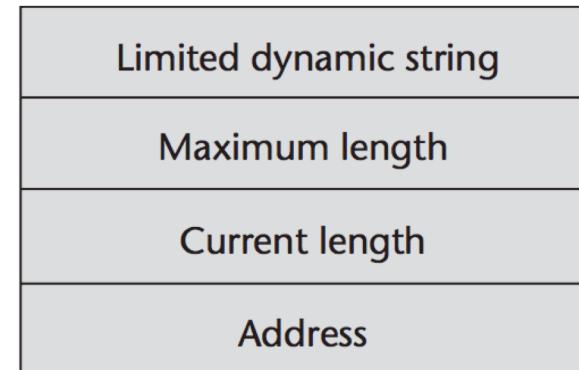


Figure 6.3

Run-time descriptor for
limited dynamic strings

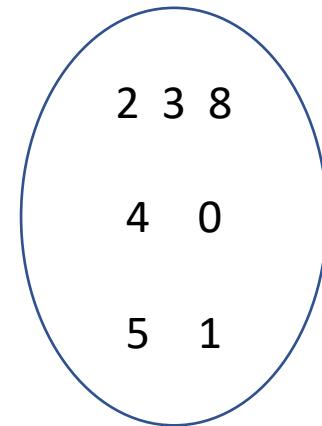


String tiplerinin derleme ve execution zaman dilimlerinde gösterimi

Kullanıcı Tanımlı Sıralı Tipler (User-Defined Ordinal Types)

1. Liste (Enumeration) Types
2. Alt Alan (Subrange) types

Yeni bir tip oluşturmayı sağlar



Değer kümesi

Belirlenmiş bir değer kümesi içinden seçim yapılır

1- Enumeration (liste) Types

- Gerçek hayatın 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.

Kullanıcı Tanımlı Sıralı Tipler (User-Defined Ordinal Types)

1- Enumeration (liste) Types

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

```
enum renkler {red, blue, green, yellow, black};          C#
```

```
renkler renk1= blue, renk12= red;                      C++
```

2- Subrange (alt alan) Types

Bir birini takip eden sıralı tipin belirlenen bir 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

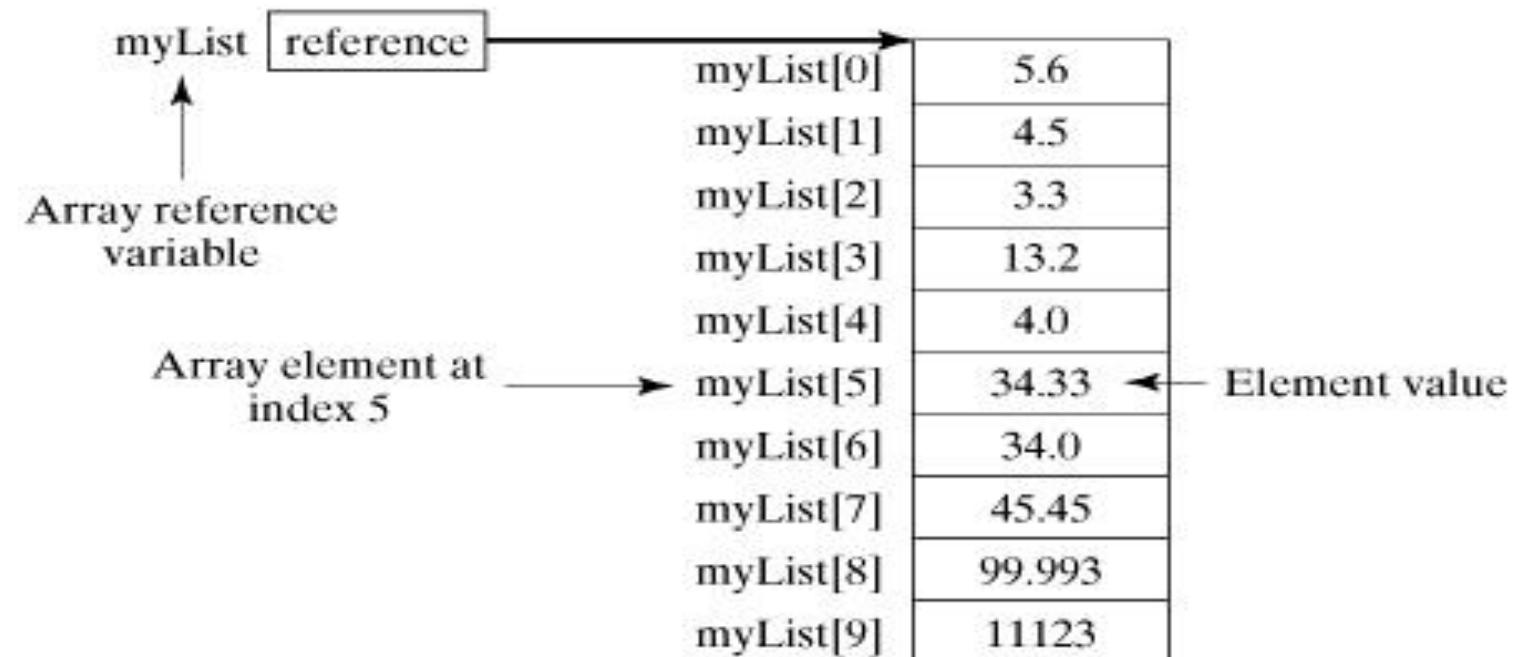
Diziler (Array Types)

Aynı tipden oluşan homojen verilerin oluşturduğu sıralı yapıdır.

Index yardımıyla dizinin elemanlarına erişim sağlanır.

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

```
double[] myList = new double[10];
```



Diziler (Array Types)

Python

```
import array as arr
a = arr.array('i', [2, 4, 6, 8])
print("First element:", a[0])
print("Second element:", a[1])
print("Second last element:", a[-1])
```

Diziler (Array Types)

JavaScript, Python, ve Ruby farklı tipdeki elemanları liste içerisinde tutar.

```
liste1 = [ 'kedi',
            'string',
            'üniversite',
            0.1,
            45,
            6.7]
```

Python List Methods

append() - Add an element to the end of the list

extend() - Add all elements of a list to the another list

insert() - Insert an item at the defined index

remove() - Removes an item from the list

pop() - Removes and returns an element at the given index

clear() - Removes all items from the list

index() - Returns the index of the first matched item

count() - Returns the count of number of items passed as an argument

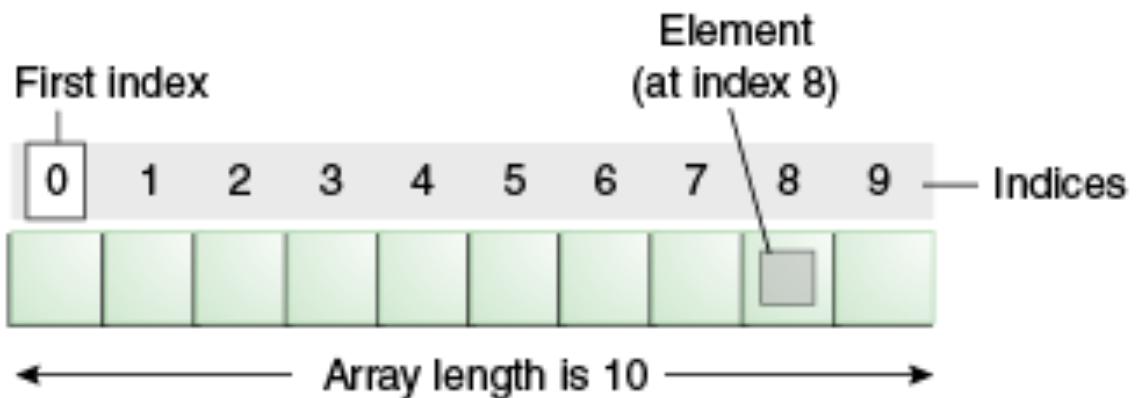
sort() - Sort items in a list in ascending order

reverse() - Reverse the order of items in the list

copy() - Returns a shallow copy of the list

Diziler (Array Types)

- Dizi içerisinde kayıtlı olan verilere erişim için index kullanılır.
- Syntax:
 - Diziadı (index)
 - []: Bir çok programlama dili
 - (): Fortran, Ada



Diziler (Array Types): index tipleri

- Dizi içerisinde saklanan değerlere index yardımıyla erişilir.
- C , Java gibi dillerde index **integer** dir
- Pascal da ise her türlü sıralı tip index olarak kullanılabilir.
 - Integer
 - Boolean
 - Char
 - enum

Diziler (Array Types)

- **Static array (efficiency):** index sınırları ve bellek yeri program çalışmadan önce belirlenir. Program çalışma süresince bellek yeri sabır kalır değişmez
- **Fixed stack-dynamic array(efficiency):** index belirlenir bellek bağlanması runtime da gerçekleşir. C ve Java'nın statik olmayan değişkenler
- **Stack-dynamic array:** index sınırları ve bellek bağlanması dinamik ve çalışma anında belirleniyor (C99) (flexible)
- **Fixed heap-dynamic array:** index sınırları ve bellek bağlanması dinamiktir. Bir defa belirlendikten sonra değişmez. (C , C++ , Java)
- **Heap-dynamic array:** index aralığı ve bellek bağlanması çalışma anında dinamik olarak belirlenir (Javascript, Python, Ruby)

Diziler (Array Types)

C# (heap dynamic)

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

Diziler (Array Types)

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)
```

```
ahmetarifaydins-MacBook-Pro:~ aaaydin$ irb
irb(main):001:0> dizi=Array.new(['a','b','c'])
=> ["a", "b", "c"]
irb(main):003:0> dizi[-2]
=> "b"
irb(main):004:0> dizi.insert(0,1)
=> [1, "a", "b", "c"]
irb(main):005:0> dizi.insert(-1,'d')
=> [1, "a", "b", "c", "d"]
irb(main):006:0> dizi.slice(1,2)
=> ["a", "b"]
irb(main):007:0> dizi.push('f')
=> [1, "a", "b", "c", "d", "f"]
irb(main):008:0> dizi.pop
=> "f"
irb(main):009:0> dizi
=> [1, "a", "b", "c", "d"]
irb(main):010:0> dizi.delete_at(2)
=> "b"
irb(main):011:0> dizi
=> [1, "a", "c", "d"]
```

Diziler (Array Types)

Bazı programlama dilleri dizilere ilk değer atamaya izin verir.

```
int list [] = {4, 5, 7, 83};      C, C++, Java, and C#
```

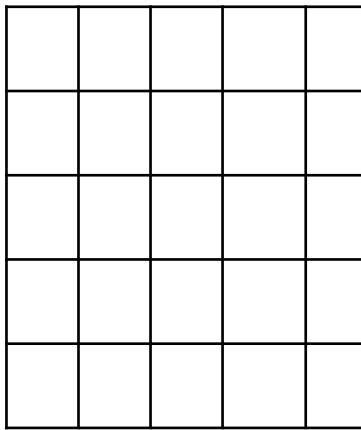
```
char isim[] = "programlama";      C ve C++
```

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

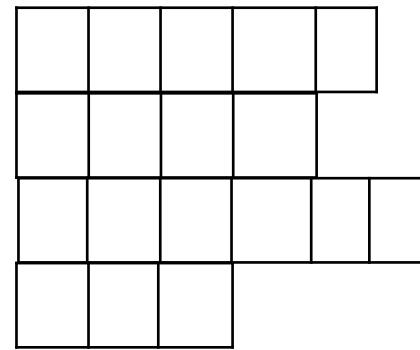
```
String[] names = ["Bob", "Jake", "Darcie"];      Java
```

Diziler (Array Types)

5 x 5



Düzenli (rectangular) dizi



Düzensiz (jagged) dizi

C, C++, java , python
destekler

Diziler (Array Types)

Programlama dilleri diziler üzerinde farklı işlemlere izin verirler.

Python

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

```
mylist [:]
```

```
mylist [:5]
```

```
mylist [2:5]
```

```
ahmetarifaydins-MacBook-Pro:~ aaaydin$ python
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on
darwin
Type "help", "copyright", "credits" or "license" for more
information.
>>> mylist=[1,2,3,4,5]
>>> mylist
[1, 2, 3, 4, 5]
>>> mylist[::]
[1, 2, 3, 4, 5]
>>> mylist[:3]
[1, 2, 3]
>>> mylist[:5]
[1, 2, 3, 4, 5]
>>> mylist[2:5]
[3, 4, 5]
>>>
```

Diziler (Array Types)

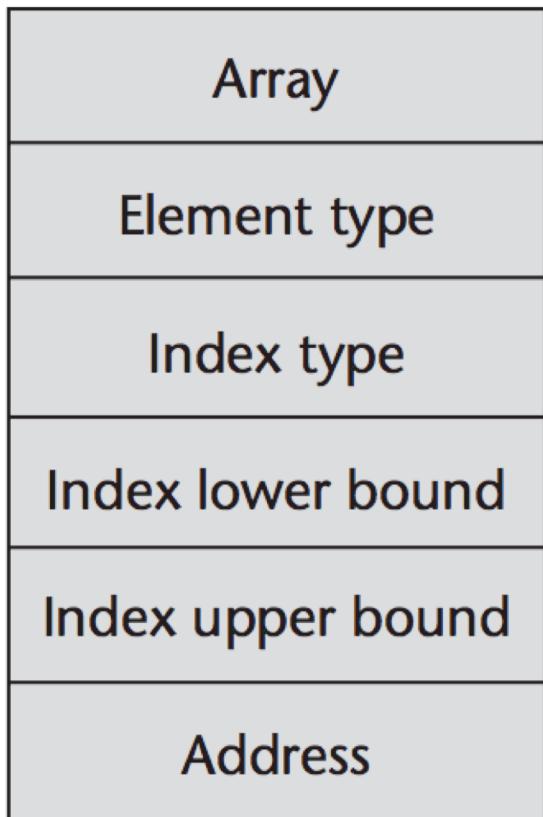
Çok boyutlu diziler matrislerin tanımlanmasında kullanılır.

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

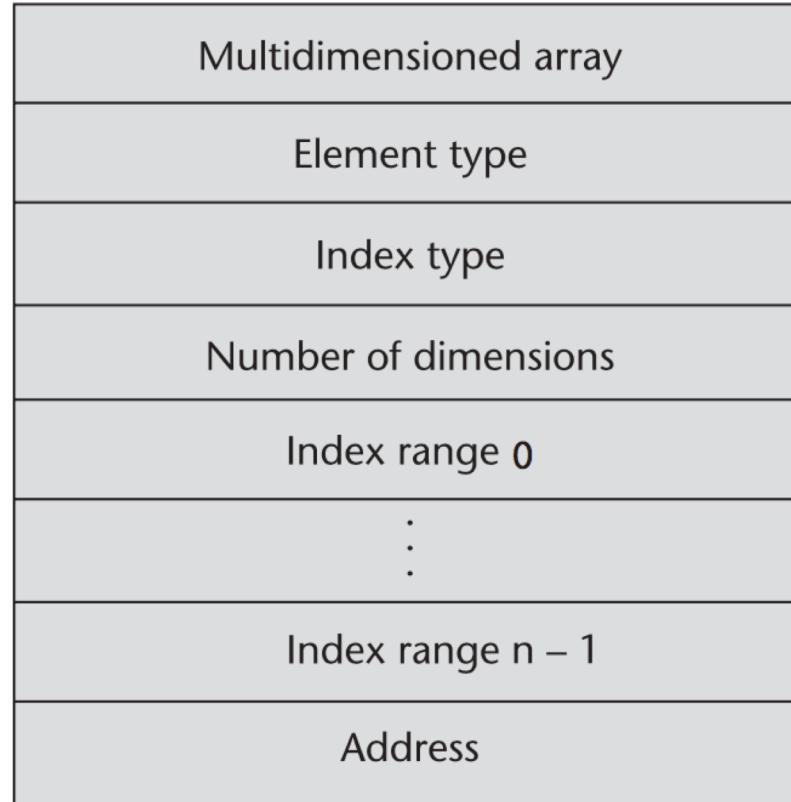
python

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

Diziler (Array Types)



Derleme zamanındaki
Tek boyutlu dizi tanımı



Derleme zamanındaki
çok boyutlu dizi tanımı