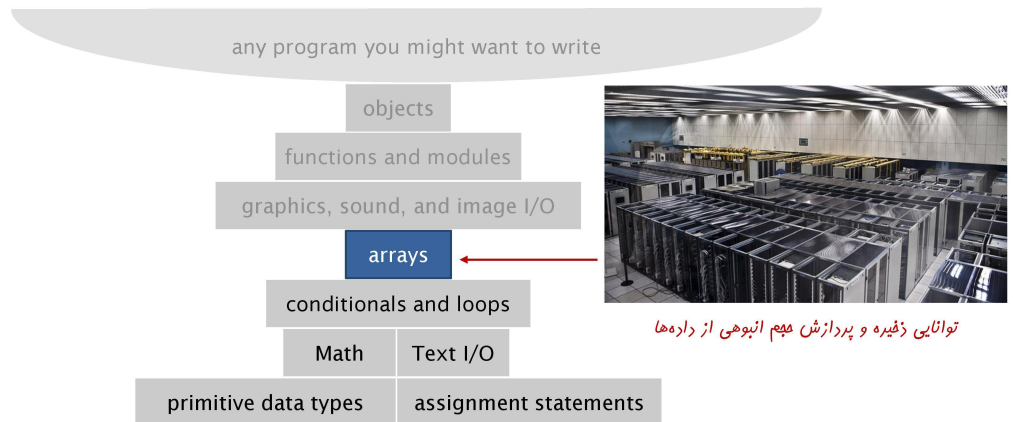
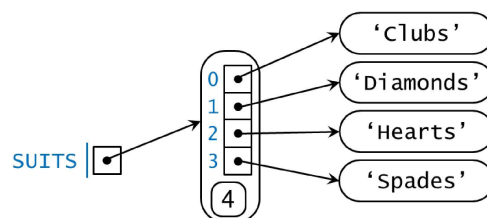


اجزای برنامه‌نویسی



آرایه‌ها

- ساختمان داده.
- روش ذخیره کردن داده‌ها در حافظه (به منظور دسترسی و پردازش آسان‌تر و کارآتر داده‌ها)
- آرایه. یک ساختمان داده به منظور ذخیره‌سازی یک دنباله از (ارجاع‌ها به) اشیا
- برای دسترسی به عناصر آرایه از شماره‌گذاری و اندیس‌گذاری استفاده می‌کنیم.



ساختمان داده آرایه

آرایه‌ها

□ آرایه. یک دنباله اندیس‌گذاری شده از (ارجاع به) اشیا.

index	value
0	2♥
1	6♠
2	A♦
3	A♥
...	
49	3♣
50	K♣
51	4♠



□ مثال‌ها.

- ۵۲ کارت بازی در یک دسته ورق.
- ۱۰ هزار دانشجوی کارشناسی در دانشگاه تبریز
- ۱ میلیون پیکسل در یک تصویر دیجیتال
- ۴ میلیارد نوکلئوتید در یک رشته DNA
- ۷۳ میلیارد پرس و جوی گوگل در یک سال
- ۸۶ میلیارد نورون در مغز
- ۵۰ تریلیون سلول در بدن انسان

پردازش تعداد زیادی مقدار از یک نوع

۱۰ مقدار، بدون استفاده از آرایه

```
# tedious and error-prone
a0 = 0.0
a1 = 0.0
a2 = 0.0
a3 = 0.0
a4 = 0.0
a5 = 0.0
a6 = 0.0
a7 = 0.0
a8 = 0.0
a9 = 0.0
...
a4 = 3.0
...
a8 = 8.0
...
x = a4 + a8
```

۱۰ مقدار، با استفاده از آرایه

```
# easy alternative
a = [0] * 10
...
a[4] = 3.0
...
a[8] = 8.0
...
x = a[4] + a[8]
```

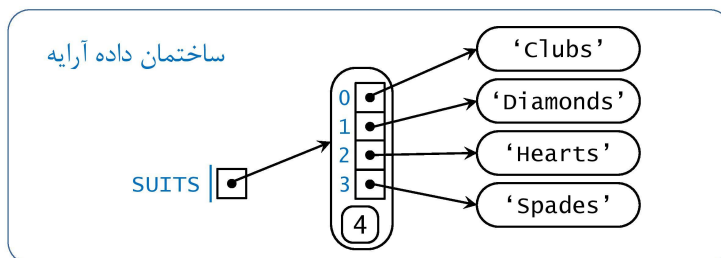
یک میلیون مقدار، با استفاده از آرایه

```
# handle huge amounts of data
a = [0] * 1000000
...
a[123456] = 3.0
...
a[987654] = 8.0
...
x = a[123456] + a[987654]
```

آرایه‌ها در پایتون

□ ساده‌ترین روش ایجاد آرایه‌ها در پایتون.

```
SUITS = ['Clubs', 'Diamonds', 'Hearts', 'Spades']
x = [0.30, 0.60, 0.10]
y = [0.50, 0.10, 0.40]
```



```
In [ ]: ▶ """
list
'index', 'count', 'insert', 'remove', 'pop', 'reverse'
, 'sort', 'extend', 'append', 'clear', 'copy',
"""
```

```
In [ ]: ▶ a = [5, 7, 12]
print(type(a)) # <class 'List'>
print(len(a)) # 3
```

```
In [ ]: ▶ print(a.index(7)) # 1
print(a[1]) # 7
a[1] = 8 # list is mutable
```

```
In [ ]: ▶ s = 'sara'
print(s[1])
# s[1]='d' # string is immutable
```

```
In [ ]: ▶ a = [1, 2]
b = [2, 1]
print(a == b) # False (list is ordered)
```

کار کردن با آرایه‌ها در پایتون

□ حلقه زدن بر روی عناصر یک آرایه.

ماسبه میانگین عناصر یک آرایه

```
total = 0.0

for i in range(len(a)):
    total += a[i]

average = total / len(a)
```

ماسبه میانگین عناصر یک آرایه

```
total = 0.0

for v in a:
    total += v

average = total / len(a)
```

```
In [ ]: ▶ friends = ['ali', 'sara', 'taha']
        for f in friends:
            print(f)
```

```
In [ ]: ▶ for i in range(3):
        print(friends[i])
```

```
In [ ]: ▶ L = [3,6,True , 'ali',2.7 , [5,8]]
```

کار کردن با آرایه‌ها در پایتون

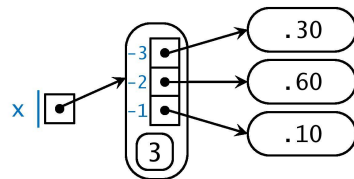
□ عملگر برش.

`a[i:j]`

□ ایجاد یک آرایه جدید شامل عناصر `a[i]` تا `a[j - 1]`.

■ مقدار پیش‌فرض `i` برابر است با 0

■ مقدار پیش‌فرض `j` برابر است با `len(a)`



□ چند مثال.

□ `a[1:]`: یک آرایه جدید شامل همه عناصر به جز اولی.

□ `a[:-1]`: یک آرایه جدید شامل همه عناصر به جز آخری.

□ `a[:]`: یک آرایه جدید شامل همه عناصر

```
In [ ]: ▶ a = [7, 5, 30, 2, 6, 25]
print(a[1:4])      # [5, 30, 2]
print(a[:3])      # [7, 5, 30]
print(a[3:])      # [2, 6, 25]
print(a[3:0])     # []
print(a[::-1])    # [25, 6, 2, 30, 5, 7]
```

```
In [ ]: ▶ print(a[0:7:2])    # [7, 30, 6]
print(a[6:0:-2])    # [25, 2, 5]
```

```
In [ ]: ▶ a[3:5]=[14, 15]
print(a)           # [7, 5, 30, 14, 15, 25]
```

```
In [ ]: ▶ a = [4, 7]
b = a*2
print(b)           # [4, 7, 4, 7]
```

```
In [ ]: ▶ a = [1, 2]
b = ['a', 'b', 'c']
c = a + b
print(c)           # [1, 2, 'a', 'b', 'c']
```

```
In [ ]: ▶ a = [7, 5, 30, 2, 6, 25]
print( 14 in a)    # False
print(14 not in a) # True
```

```
In [ ]: ▶ a = [3, [109, 27], 4, 15]
print(a[1])        # [109, 27]
print(a[1][1])    # 27
print(len(a))     # 4
```

```
In [ ]: ▶ #####
```

```
In [ ]: ▶ a = [7, 5, 30, 2, 6, 25]
m = -1
for i in a:
    if i > m:
        m = i
```

کار کردن با آرایه‌ها در پایتون

□ توابع پیش‌ساخته.

□ طول آرایه، محاسبه مجموع، کمینه، بیشینه.

محاسبه میانگین عناصر یک آرایه

```
total = 0.0

for i in range(len(a)):
    total += a[i]

average = total / len(a)
```

```
average = float(sum(a)) / len(a)
```

تابع	توضیحات
len	تعداد عناصر آرایه
sum	مجموع عناصر آرایه
min	مقدار کمینه آرایه
max	مقدار بیشینه آرایه

```
In [ ]: ▶ print(m)           # 30

print(max(a))      # 30
print(min(a))      # 2
print(sum(a))      # 75
```

```
In [ ]: ▶ s = 0
for i in a:
    s += i
print(s)           # 75
```

```
In [ ]: ▶ a = [1, 2, 6, 5, 2]
print(a.count(2)) # 2
```

```
In [ ]: ▶ a = [1, 2, 6, 5, 2]
a.insert(2,13)
print(a)           # [1, 2, 13, 6, 5, 2]
```

```
In [ ]: ▶ a = [1, 2, 6, 5, 2]
a.remove(2)
print(a)           #[1, 6, 5, 2]
a.remove(2)
print(a)           #[1, 6, 5]
```

```
In [ ]: ▶ x = [10, 15, 12, 8]
a = x.pop()
print(x)           # [10, 15, 12]
print(a)           # 8
```

```
In [ ]: ▶ y = ['a', 'b', 'c']
        p = y.pop(1)
        print(p)      # b
        print(y)     # ['a', 'c']
```

```
In [ ]: ▶ a = [5, 9, 3]
        del a[1]
        print(a)     # [5, 3]
```

```
In [ ]: ▶ a = [0, 1, 2, 3, 4, 5, 6]
        del a[2:4]
        print(a)    # [0, 1, 4, 5, 6]
```

```
In [ ]: ▶ a = [1,2,3]
        a.reverse()
        print(a)    # [3, 2, 1]
```

```
In [ ]: ▶ a = [2,4,3,5,1]
        a.sort()
        print(a)    # [1, 2, 3, 4, 5]
```

```
In [ ]: ▶ x = [1, 2, 3]
        y = [4, 5]
        x.extend(y)
        print(x)    # [1, 2, 3, 4, 5]
        print(len(x)) # 5
        print(len(y)) # 2
```

```
In [ ]: ▶ x = [1, 2, 3]
        y = [4, 5]
        y.extend(x)
        print(y)    # [4, 5, 1, 2, 3]
        print(len(x)) # 3
        print(len(y)) # 5
```

```
In [ ]: ▶ x = [1, 2, 3]
        y = [4, 5]
        x.append(y)
        print(x)    # [1, 2, 3, [4, 5]]
        print(len(x)) # 4
        print(len(y)) # 2
```

```
In [ ]: ▶ a = [1,2,3]
        a.append(4)
        print(a)    # [1, 2, 3, 4]
```

```
In [ ]: ▶ a = [1,2,3]
a.clear()
print(a)      # []
print(len(a)) # 0
```

```
In [ ]: ▶ a = [1,2,3]
b = a.copy()
print(b)      # [1, 2, 3]
```

```
In [ ]: ▶ #####
```

```
In [ ]: ▶ a = []
for i in range(4):
    a.append(i)
print(a)      # [0, 1, 2, 3]
```

```
In [ ]: ▶ a = [i for i in range(4)]
print(a)      # [0, 1, 2, 3]
```

```
In [ ]: ▶ a = [i*2 for i in range(4)]
print(a)      # [0, 2, 4, 6]
```

```
In [ ]: ▶ # [9, 16, 25]
a = [i*i for i in range(3,6)]
print(a)
```

```
In [ ]: ▶ a = [1 , -2 , 5 , -56 , 8]
b = [abs(i) for i in a]
print(b)      # [1, 2, 5, 56, 8]
```

```
In [ ]: ▶ import math
a = [round(math.pi,i) for i in range(1,5)]
print(a)      # [3.1, 3.14, 3.142, 3.1416]
```

```
In [ ]: ▶ a = ['$ali', 'sara$']
b = [i.strip('$') for i in a]
print(b)      # ['ali', 'sara']
```

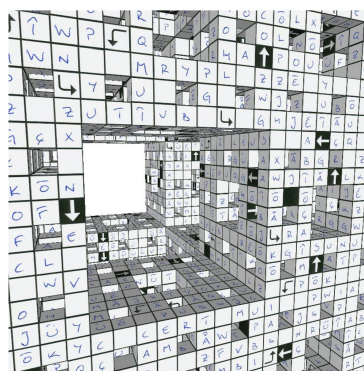
```
In [ ]: ▶ a = [11, 8, 14, 20 , 2]
b = [i for i in a if i < 10]
print(b)      # [8, 2]
```



```
In [ ]: ▶ a = [1, 2]
b = [1, 4, 5]
c = []
for i in a:
    for j in b:
        if i != j:
            c.append((i,j))
print(c)      # [(1, 4), (1, 5), (2, 1), (2, 4), (2, 5)]
```

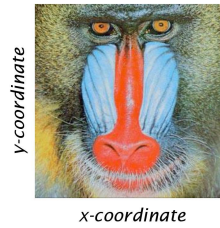
```
In [ ]: ▶ a = [2.6, float('NaN') , 4.8 , 6.9, float('NaN')]
b = []
import math
for i in a:
    if not math.isnan(i):
        b.append(i)
print(b)      # [2.6, 4.8, 6.9]
```

آرایه‌های چند بعدی



آرایه‌های دو بعدی

		grade						
		0	1	2	3	4	5	...
student ID	0	A	A	C	B	A	C	
	1	B	B	B	B	A	A	
	2	C	D	D	B	C	A	
	3	A	A	A	A	A	A	
	4	C	C	B	C	B	B	
	5	A	A	A	B	A	A	
...								



□ آرایه‌های دو بعدی.

- ماتریس‌ها در محاسبات ریاضی.
- یک جدول از نمرات به ازای هر دانشجو و تمرینات.
- یک جدول از داده‌ها به ازای هر آزمایش و نتایج آن.
- تراکنش‌های مربوط به مشتریان یک بانک.
- پیکسل‌ها در یک تصویر دیجیتال.
- داده‌های جغرافیایی.

آرایه‌های دو بعدی در پایتون: ایجاد و مقداردهی

- دستیابی به عناصر. $a[i][j]$ = عنصر واقع در سطر i و ستون j .
- اندیس گذاری مبتنی بر صفر. اندیس سطرها و ستون‌ها از صفر شروع می‌شود.

```
m = 10
n = 3
a = []
for i in range(m):
    row = [0] * n
    a += [row]
```

	$a[0][0]$	$a[0][1]$	$a[0][2]$
	$a[1][0]$	$a[1][1]$	$a[1][2]$
	$a[2][0]$	$a[2][1]$	$a[2][2]$
	$a[3][0]$	$a[3][1]$	$a[3][2]$
	$a[4][0]$	$a[4][1]$	$a[4][2]$
$a[5]$ →	$a[5][0]$	$a[5][1]$	$a[5][2]$
	$a[6][0]$	$a[6][1]$	$a[6][2]$
	$a[7][0]$	$a[7][1]$	$a[7][2]$
	$a[8][0]$	$a[8][1]$	$a[8][2]$
	$a[9][0]$	$a[9][1]$	$a[9][2]$

↑
↓
←
→
↑
↓
←
→

آرایه‌های دو بعدی در پایتون: دسترسی به عناصر

- دستیابی به عناصر. $a[i][j]$ = عنصر واقع در سطر i و ستون j .
- اندیس گذاری مبتنی بر صفر. اندیس سطرها و ستون‌ها از صفر شروع می‌شود.

```
for i in range(m):
    for j in range(n):
        print(a[i][j], end=' ')
    print()
```

```
for row in a:
    for v in row:
        print(v, end=' ')
    print()
```

	a[0][0]	a[0][1]	a[0][2]
	a[1][0]	a[1][1]	a[1][2]
	a[2][0]	a[2][1]	a[2][2]
	a[3][0]	a[3][1]	a[3][2]
	a[4][0]	a[4][1]	a[4][2]
a[5] →	a[5][0]	a[5][1]	a[5][2]
	a[6][0]	a[6][1]	a[6][2]
	a[7][0]	a[7][1]	a[7][2]
	a[8][0]	a[8][1]	a[8][2]
	a[9][0]	a[9][1]	a[9][2]

یک آرایه ۱۰×۳

مقداردهی آرایه دو بعدی

- مقداردهی آرایه دو بعدی با لیست کردن مقادیر.

```
a = [[ .92, .02, .02, .02, .02 ],
      [ .02, .02, .32, .32, .32 ],
      [ .02, .02, .02, .92, .02 ],
      [ .92, .02, .02, .02, .02 ],
      [ .47, .02, .47, .02, .02 ]]
```

	.92	.02	.02	.02	.02
سطر ۱ →	.02	.02	.32	.32	.32
	.02	.02	.02	.92	.02
	.92	.02	.02	.02	.02
	.47	.02	.47	.02	.02

ستون ۳

```
In [ ]: ▶ print('---- matrix ----')
```

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

print(len(m)) # 3
```

```
In [ ]: ▶ print(m[0]) # [1, 2, 3]
```

```
In [ ]: ▶ for i in m:  
        print(i)
```

```
In [ ]: ▶ for i in m:  
        print(i[0],end=' ') # 1 4 7
```

```
In [ ]: ▶ for i in range(0,3) :  
        print(m[i][i],end=' ') # 1 5 9
```

```
In [ ]: ▶ for i in range(0,3):  
        print(m[i][2-i],end=' ') # 3 5 7
```

```
In [ ]: ▶ a = []  
        a.extend([sum(i) for i in m])  
        print(a) # [6, 15, 24]
```

```
In [ ]: ▶ b = []  
        for col in range(3):  
            b.append(sum(i[col] for i in m))  
        print(b) # [12, 15, 18]
```

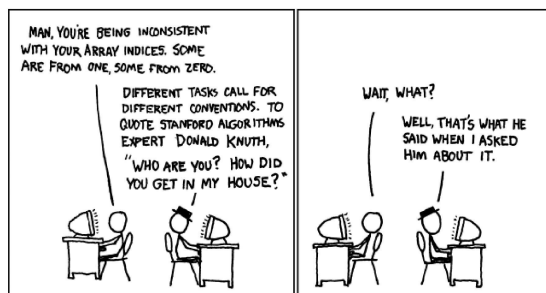
```
In [ ]: ▶ x = 2  
        y = x  
        y += 1  
        print(x) # 2  
        print(y) # 3
```

```
In [ ]: ▶ x = []  
        y = x  
        y.append(5)  
        print(x) # [5]  
        print(y) # [5]
```

خلاصه

□ آرایه‌ها.

- یک روش سازمان یافته به منظور ذخیره‌سازی حجم انبوهی از داده‌ها.
- استفاده از آرایه تقریباً به اندازه استفاده از انواع اولیه، ساده است.
- با داشتن اندیس یک عنصر، می‌توانیم به صورت مستقیم به آن عنصر دسترسی داشته باشیم.



□ گام بعدی.

- خواندن مقدار زیادی داده از فایل و
- ذخیره‌سازی آنها در یک آرایه.

دانشگاه شهید مدنی آذربایجان

برنامه نویسی مقدماتی با پایتون

امین گلزاری اسکوئی

۱۴۰۰-۱۴۰۱

[Codes and Projects \(click here\) \(https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021\)](https://github.com/Amin-Golzari-Oskouei/Python-Programming-Course-Basic-2021), [slides and videos \(click here\)](#) (<https://drive.google.com/drive/folders/1ZsQjBJJ4UAAp9zrGxm3c4qrhmvGBUYHw>)