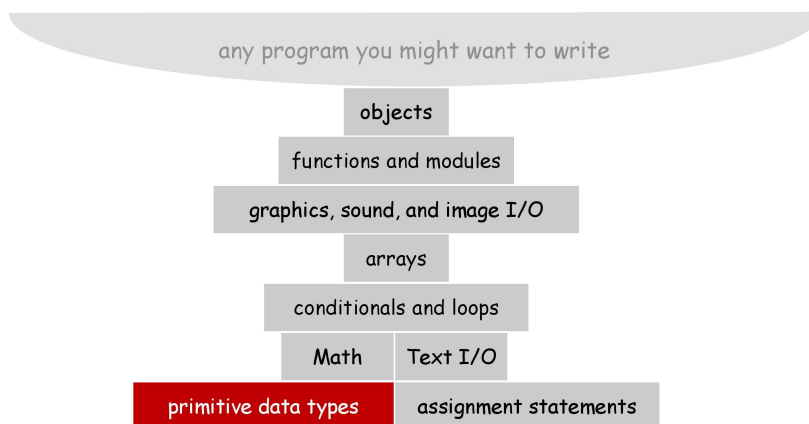


اجزای برنامه‌نویسی



انواع داده‌ای پیش‌ساخته

□ نوع داده‌ای. یک مجموعه از مقادیر به همراه عملیات تعریف شده بر روی آن مقادیر.

<i>type</i>	<i>set of values</i>	<i>common operators</i>	<i>sample literals</i>
int	integers	+ - * // % **	99 12 2147483647
float	floating point numbers	+ - * / **	3.14 2.5 6.022e23
bool	true-false values	and or not	True False
str	sequence of characters	+	'AB' 'Hello' '2.5'

برنامه = داده + الگوریتم

اعداد صحیح

□ نوع داده‌ای عدد صحیح. مفید برای بیان الگوریتم‌ها.

<i>values</i>				integers				
<i>typical literals</i>			1234	99	-99	0	1000000	
<i>operations</i>	sign	add	subtract	multiply	division	remainder	power	
<i>operators</i>	+ -	+	-	*	//	%	**	

اعداد اعشاری

□ نوع داده‌ای `float`. مفید در محاسبات علمی و کاربردهای تجاری.

<i>values</i>				real numbers			
<i>typical literals</i>			3.14159	6.022e23	-3.0	1.4142135623730951	
<i>operations</i>			add	subtract	multiply	division	exponentiation
<i>operators</i>			+	-	*	/	**

مقادیر بولی

□ نوع داده‌ای `bool` مفید برای کنترل منطق و روند اجرای برنامه.

<i>values</i>	true or false		
<i>literals</i>	True	False	
<i>operations</i>	and	or	not
<i>operators</i>	and	or	not

a	not a	a	b	a and b	a or b
False	True	False	False	False	False
True	False	False	True	False	True
		True	False	False	True
		True	True	True	True

```
In [ ]: ▶ """
Python Data Types:
        str , int , float , bool , complex ,list , tuple , set , dict , byte
"""
```

```
In [ ]: ▶ s = 'amin'
print(type(s))      # str
```

```
In [ ]: ▶ i = 2
print(type(i))      # int
```

```
In [ ]: ▶ f = 2.5
print(type(f))      # float
```

```
In [ ]: ▶ c = 2 + 3j      # 2 is the real part and 3 is imaginary
print(type(c))      # complex
```

```
In [ ]: ▶ b = True
print(type(b))      # bool

print(bool(5))      # True
print(bool(-2))     # True
print(bool('ali'))  # True

print(bool(0))      # False
print(bool(''))     # False
```

```
In [ ]: ▶ print(bool([]))      # False (empty list)
          print(bool({}))    # False (empty dictionary)
          print(bool(()))   # False (empty tuple)
```

```
In [ ]: ▶ print('--variable names---')
        """
        All identifiers must start with a letter or underscore (_),
        you can't use digits.
        Identifiers can contain letters, digits and underscores (_).
        Identifiers can't be a keyword.
        They can be of any length.
        """

        print('a2'.isidentifier())    # True
        print('2a'.isidentifier())    # False
        print('_myvar'.isidentifier()) # True
        print('my_var'.isidentifier()) # True
        print('my-var'.isidentifier()) # False
        print('my var'.isidentifier()) # False
        print('my$'.isidentifier())   # False
        print('my#'.isidentifier())   # False
```

```
In [ ]: ▶ # You cannot use reserved words as variable names
```

```
"""
False  class  return  is      finally
None   if     for     lambda  continue
True   def    from    while   nonlocal
and    del    global not     with
as     elif   try    or      yield
assert else   import pass
break  except in    raise
"""

from keyword import iskeyword
print( iskeyword('if'))  # True
```

تبدیل نوع

- تبدیل نوع. تبدیل از یک نوع به نوع دیگر.
- تبدیل نوع صریح: با استفاده تبدیل نوع یا توابع

<i>function call</i>	<i>description</i>
<code>str(x)</code>	تبدیل شی X به یک رشته
<code>int(x)</code>	تبدیل رشته یا عدد اعشاری X به یک عدد صحیح
<code>float(x)</code>	تبدیل رشته یا عدد صحیح X به یک عدد اعشاری
<code>round(x)</code>	نزدیک‌ترین عدد صحیح به عدد X

- تبدیل نوع ضمنی (خودکار).

`x = 10 / 4.0` عدد صحیح ۱۰ به صورت فلوئید به یک عدد اعشاری تبدیل می‌شود

```
In [ ]: ▶ print('---Python Casting---')
```

```
i = 5
print(float(i))      # 5.0
```

```
In [ ]: ▶ s = '12'
print(int(s) + 1)    # 13
```

```
In [ ]: ▶ x = 1
c = complex(x)
print(c)             # (1+0j)
```

```
In [ ]: ▶ n = 12.5
print('%i' % n)      # 12
print('%f' % n)      # 12.500000
print('%e' % n)      # 1.250000e+01
```

```
In [ ]: ▶ a = 5
b = 1
print('Five plus one is {a + b}')    # Five plus one is {a + b}
print(f'Five plus one is {a + b}')  # Five plus one is 6
```

```
In [ ]: ▶ a = b = c = 5      # this statement assign 5 to c, b and a.
print(a, b, c)             # 5 5 5
```

```
In [ ]: ▶ x = 1
         y = 2
         y, x = x, y      # assign y value to x and x value to y
         print(x)        # 2
         print(y)        # 1
```

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

```
In [ ]: ▶ print('---string---')
         #Slicing string Syntax: s[start:end]

         s = "GolzariOskouei"

         print(s[0])     #G
         print(s[8])     #s
         print(s[-1])    #i
         print(s[-2])    #e
         print(s[0:4])   #Golz
         print(s[:4])    #Golz
         print(s[-9:-5]) #riOs
         print(s[4:])    #ariOskouei
         print(s[4:9])   #ariOs
         print(s[-5:])   #kouei
```

```
In [ ]: ▶ l = ["apples", "grapes", "oranges"]
         print(type(l)) # list
```

```
In [ ]: ▶ t = ("apple", "banana", "cherry")
         print(type(t)) # tuple
```

```
In [ ]: ▶ d = {'id': '123', 'name': 'amin'}
         print(type(d)) # dict
```

```
In [ ]: ▶ s = {'apple', 'banana', 'cherry'}
         print(type(s)) # set
```

```
In [ ]: ▶ #Receiving input from Console  
a = int(input('Enter a:'))  
b = int(input('Enter b:'))  
c = a + b  
print(c)
```

دانشگاه شهید مدنی آذربایجان
برنامه نویسی مقدماتی با پایتون
امین گلزاری اسکویی
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>)