

پایتون

شیرین بقولی زاده

کاربرد زبان پایتون

✓ ایجاد برنامه‌های تحت وب

✓ ایجاد نرم‌افزارهای کاربردی

✓ برقراری ارتباط با پایگاه‌های داده

✓ کار بر روی داده‌های حجیم و انجام عملیات پیچیده

✓ پسوند فایل‌ها `.py` است

متغیرها

Example

Variables in Python:

```
x = 5  
y = "Hello, World!"
```

✓ متغیرها به محض مقداردهی اولیه ایجاد می‌شوند.

✓ دستور جداگانه‌ای در پایتون برای تعریف متغیر وجود ندارد.

Example

```
x = "John"  
# is the same as  
x = 'John'
```

✓ برای مقداردهی به صورت رشته از علامت ' یا " استفاده می‌شود

اسامی متغیرها

Example

This will create two variables:

```
a = 4
A = "Sally"
#A will not overwrite a
```

✓ حساس به حروف بزرگ و کوچک

✓ شروع حتما با یک حرف انگلیسی یا علامت _
• نام متغیر نمی‌تواند با عدد شروع شود.

✓ شامل فقط حروف انگلیسی، اعداد و علامت _

Example

Legal variable names:

```
myvar = "John"
my_var = "John"
_my_var = "John"
myVar = "John"
MYVAR = "John"
myvar2 = "John"
```

Example

Illegal variable names:

```
2myvar = "John"
my-var = "John"
my var = "John"
```

اسامی چند کلمه‌ای

```
my_variable_name = "John"
```

```
myVariableName = "John"
```

```
MyVariableName = "John"
```

print

Example

```
x = "Python is "  
y = "awesome"  
z = x + y  
print(z)
```

Example

```
x = "awesome"  
print("Python is " + x)
```

✓ امکان استفاده از عملگر جمع بین رشته‌ها یا متغیرها از این جنس

Example

```
x = 5  
y = 10  
print(x + y)
```

✓ امکان استفاده از عملگر جمع بین اعداد یا متغیرها از این جنس

print

Example

```
x = 5  
y = "John"  
print(x + y)
```

✓ خطا در استفاده از عملگر جمع مابین اعداد و رشته‌ها

برخی انواع داده (Data Type)

✓int: اعداد صحیح مثبت و منفی و بدون محدودیت در طول

Integers:

```
x = 1
y = 35656222554887711
z = -3255522
```

Floats:

```
x = 1.10
y = 1.0
z = -35.59
```

Floats:

```
x = 35e3
y = 12E4
z = -87.7e100
```

✓float: اعداد اعشاری مثبت و منفی
• استفاده از حرف e یا E برای نمایش توان ده

برخی انواع داده (Data Type)

Complex:

```
x = 3+5j  
y = 5j  
z = -5j
```

✓complex: اعداد مختلط

✓bool: True و False

مقادیر Boolean

Example

```
print(10 > 9)  
print(10 == 9)  
print(10 < 9)
```

Output:

```
True  
False  
False
```

✓ کاربرد در ارزیابی یک عبارت

✓ شامل دو مقدار True و False

✓ شروع دو کلمه با حروف بزرگ

✓ تمام رشته‌ها، به جز رشته تهی، معادل True هستند

✓ تمام اعداد، به جز عدد صفر، معادل True هستند

اطلاع از نوع داده

✓ استفاده از type()

Example

```
x = 5  
y = "John"  
print(type(x))  
print(type(y))
```

تغییر نوع متغیر

✓ از طریق مقداردهی

Example

```
x = 4          # x is of type int
x = "Sally"    # x is now of type str
print(x)
```

Example

```
x = str(3)     # x will be '3'
y = int(3)     # y will be 3
z = float(3)   # z will be 3.0
```

✓ با استفاده از دستورات casting

عملگرها

✓ عملگرهای ریاضی

✓ عملگرهای انتساب

✓ عملگرهای مقایسه‌ای

✓ عملگرهای منطقی

✓ عملگرهای بیتی

عملگرهای ریاضی

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

عملگرهای انتساب

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3

عملگرهای مقایسه‌ای

Operator	Name	Example
==	Equal	<code>x == y</code>
!=	Not equal	<code>x != y</code>
>	Greater than	<code>x > y</code>
<	Less than	<code>x < y</code>
>=	Greater than or equal to	<code>x >= y</code>
<=	Less than or equal to	<code>x <= y</code>

عملگرهای منطقی

Operator	Description	Example
and	Returns True if both statements are true	<code>x < 5 and x < 10</code>
or	Returns True if one of the statements is true	<code>x < 5 or x < 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x < 5 and x < 10)</code>

توضیحات

- ✓ برای توضیحات (comment) از علامت # ابتدای خط استفاده می‌شود.
- ✓ در نتیجه آن خط در اجرا نادیده گرفته می‌شود.

Example

Comments in Python:

```
#This is a comment.  
print("Hello, World!")
```

توضیحات

Example

```
print("Hello, World!") #This is a comment
```

✓ توضیحات می‌توانند در انتهای یک خط کد قرار گیرند.

Example

```
""  
This is a comment  
written in  
more than just one line  
""  
  
print("Hello, World!")
```

✓ توضیحات چندخطی در پایتون وجود ندارد.

- به عنوان جایگزین می‌توان از رشته‌های چندخطی استفاده کرد.
- رشته‌های چندخطی که به مقداری منتسب نمی‌شوند نادیده گرفته می‌شوند.

دریافت ورودی

```
username = input("Enter username:")  
print("Username is: " + username)
```

✓ با استفاده از تابع `input()`

✓ `input("message")`

✓ منتظر ماندن برنامه برای دریافت ورودی از کاربر

✓ دریافت به صورت رشته

```
# Program to check input  
# type in Python  
  
num = input ("Enter number :")  
print(num)  
name1 = input("Enter name : ")  
print(name1)  
  
# Printing type of input value  
print ("type of number", type(num))  
print ("type of name", type(name1))
```

Output :

```
Enter number :123  
123  
Enter name : geeksforgeeks  
geeksforgeeks  
type of number <class 'str'>  
type of name <class 'str'>  
>>> |
```

دریافت ورودی

```
username = input("Enter username:")  
print("Username is: " + username)
```

✓ با استفاده از تابع `input()`

✓ `input("message")`

✓ منتظر ماندن برنامه برای دریافت ورودی از کاربر

✓ دریافت به صورت رشته

```
# Program to check input  
# type in Python  
  
num = input ("Enter number :")  
print(num)  
name1 = input("Enter name : ")  
print(name1)  
  
# Printing type of input value  
print ("type of number", type(num))  
print ("type of name", type(name1))
```

Output :

```
Enter number :123  
123  
Enter name : geeksforgeeks  
geeksforgeeks  
type of number <class 'str'>  
type of name <class 'str'>  
>>> |
```

دندانه‌گذاری (Indentation)

✓ برای بلاک‌بندی تعدادی خط کد از فاصله‌گذاری (space) ابتدای آن‌ها استفاده می‌شود.

Example

```
if 5 > 2:  
    print("Five is greater than two!")
```

Syntax Error:

```
if 5 > 2:  
print("Five is greater than two!")
```

دندانه گذاری

✓ برای بلاک بندی، تعداد فاصله ها در ابتدای خط به دلخواه برنامه نویس است.

Example

```
if 5 > 2:  
    print("Five is greater than two!")  
if 5 > 2:  
    print("Five is greater than two!")
```

دندانه گذاری

تمام کدهای داخل یک بلاک باید دارای تعداد فاصله یکسان باشند ✓

Example

Syntax Error:

```
if 5 > 2:  
    print("Five is greater than two!")  
        print("Five is greater than two!")
```