

# 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

✓متغیرها به محض مقداردهی اولیه ایجاد می‌شوند.

✓دستور جداگانه‌ای در پایتون برای تعریف متغیر وجود ندارد.

## Example

Variables in Python:

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

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

## Example

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

## 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 نمی تواند 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)

Integers:

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

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

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 ✓

## 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

# توضیحات

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

## 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.

# 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!")
```