

# آموزش پایتون از پایه : متغیر ها ، نوع اطلاعات و ساختار کنترل

## لینک مطلب :

<http://code.tutsplus.com/articles/python-from-scratch-variables-data-types-and-control-structure--net-20562>

دوباره به آموزش پایتون از پایه خوش آمدید ، جایی که از پایه یاد می گیرید ---! در درس قبل ، ما پایتون را نصب کردیم و آماده شدیم . امروز ، کمی جلوتر میریم ، و چیزهای مهم را می آموزیم . ما متغیر ها ، اپراتور ها را مرور می کنیم و بعد کار خود را با یادگیری ساختار های کنترلی برای مدیریت جاری کردن اطلاعات شما تمام می کنیم .

## متغیر ها (Variables)

متغیر ها اولین چیزی هستند که در هر زبانی باید یاد بگیرید . شما می توانید به آنها مثل کانتینر های دارای نام برای هر نوع از اطلاعات نگاه کنید . سینتکس (syntax) یا ترکیب شناسایی آنها به این صورت است :

name = value

نام را هر چیزی که میخواهید میتونید قرار دهید ، و مقدار آنها میتواند هر نوعی از اطلاعات یا همون دیتا باشد .

## نوع اطلاعات (Data Types)

اطلاعات نوع های مختلفی دارد ولی در زیر چهار تا از مهم ترین هاش رو میاریم :

### اعداد (Numbers)

اعداد میتونند یا اعداد صحیح (integers) باشند یا اعداد اعشاری (floating point numbers).

\* اعداد صحیح عدد کامل هستند

\* اعداد اعشاری اعشار دارند

### رشته ها (Strings)

رشته ها خط های متنی هستند که میتونن هر کاراکتری رو داشته باشند . آنها با علامت نقل قول ("" ) یا علامت نقل قول تکی (') شناسایی می شن .

```

1 | empty = ""
2 | escaped = "Can\'t"
3 | greeting = "Hello World"
4 | multiLine = "This is a long \n\
5 | string of text"

```

شما باید علامت های نقل قول دوتایی یا تکی رو که داخل خود رشته هستند را با بک اسلش رد کنید . وگر نه ، پایتون فکر میکنه که از اون ها برای اتمام رشته استفاده کردید . برای رفتن به خط بعد از کاراکتر زیر استفاده کنید :

\n

پایتون همچنین درج رشته رو توسط علامت درصد پشتیبانی میکنه مثل زیر:

```

1 | name = "John Doe"
2 | greeting = "My name is %s" % name

```

شما میتونید به ست های کاراکتر ها توسط اسلایس دسترسی داشته باشید ، که از علامت [ ] استفاده میکنه :

```

1 | "Hello"[2] #outputs "l"

```

## بولین ها (Booleans)

بولین ها یک از مقدار های **صحیح (True)** یا **غلط (False)** را نشان می دهند . نکته قابل توجه اینه که حرف اول باید بزرگ نوشته بشه . آنها اطلاعاتی را نشان می دهند که باید یک چیز یا چیز دیگری باشند . برای مثال :

isMale = True # Could be used in software with a database of users

isAlive = False # Could be used in a game , set when the character dies

بعد از علامت # در خط اول نوشته : در یک برنامه با دیتابیس کاربران میتونه استفاده بشه  
خط دومش : در یک بازی میتونه استفاده بشه ، وقتی شخصیت میمیره .

## لیست ها (Lists)

لیست ها برای گروه بندی داده های دیگر استفاده میشود . به آنها Arrays گفته میشود در تقریبا تمام زبان های دیگر . شما میتونید توسط علامت های [ ] یک لیست بسازید :

```

1 | emptyList = []
2 | numbersList = [1, 2, 3]
3 | stringsList = ["spam", "eggs"]
4 | mixedList = ["Hello", [1, 2, 3], False]

```

همون طور که در بالا می بینید ، لیست ها میتونن هر نوع داده ای را شامل بشن ، حتی لیست های دیگه یا کلا هیچی .

شما میتونید قسمت های مختلف لیست را توسط فهرست لیست (List Indexes) دسترسی داشته باشید . سینتکس یا ترکیبش همون مثل رشته هاست :

```
1 | numbersList[1] #outputs 2
2 | stringList[0] #outputs spam
3 | mixedList[1][2] #outputs 3
```

اگر در لیست شما یک لیست دیگر بود ، شما میتونید توسط چند تا فهرست لیست به آنها دسترسی پیدا کنید .

## کامنت ها یا نظر ها (Comments)

کامنت ها برای توصیف کد بکار میرن . در مواقعی که میخوانین دوباره به کدتون برگردین ، یا برای کار کردن روی پروژه با کسی دیگر .

```
1 | #This a comment on it's own line
2 | #You create them with the hash symbol
3 | var = "Hello" #They can be on the same line as code
```

## اپراتورها (Operators)

شما قبلا اپراتو ها رو دیدید . اون ها چیز هایی مثل  $+$  یا  $-$  هستند ، و به همون صورتی که در مدرسه ازشون استفاده می کردید در اینجا هم استفاده می کنید .

```
1 | 2 + 3 #Addition, returns 5
2 | 8 - 5 #Subtraction, returns 3
3 | 2 * 6 #Multiplication, returns 12
4 | 12 / 3 #Division, returns 4
5 | 7 % 3 #Modulo, returns the remainder from a division, 1 in this case.
6 | 3**2 #Raise to the power, returns 9
```

شما همچنین میتونید نتیجه یک عملیات را روی یک متغیر اعمال کنید با برگشت به همان متغیر با ترکیب اپراتور با علامت مساوی . برای مثال  $a += b$  نسخه کوتاه شده ی  $a = a + b$

```
1 | x = 2
2 | x += 4 #Adds 4 to x, it now equals 6
3 | x /= 2 #Divides x by 2, it now equals 3
```

## ساختار کنترلی (Control Structures)

وقتی که متغیر ها رو ساختید و دستکاریشون کردید ، ساختار های کنترلی به شما اجازه کنترل جاری شدن داده ها رو میدن . دو نوع از آنها که امروز می آموزیم شرطی ها (Conditionals) و حلقه ها (Loops) هستند .

## شرطی ها (Conditionals)

شرط ها به شما اجازه اجرای بلوک های کد مختلف رو بر اساس مقدار داده ها میدن .

```
1 a = 2
2 b = 3
3
4 if a < b:
5     print "Success"
```

## حلقه ها (Loops)

دو نوع از حلقه ها که ما در اینجا بحث می کنیم اینها هستند :

for loops

while loops

for loops ها با استفاده از لیست ها کار میکنند ، و while loops ها با استفاده از شرطی ها .

### While loops

```
1 a, b = 0, 5
2
3 while a < b:
4     print a
5     a += 1
```

### for loops

```
1 myList = [1, 2, 3, 4, 5]
2
3 for a in myList:
4     print a
```

## جمع بندی

این مطالب امروز بود ، اما ما تکنیک های زیادی رو پوشش دادیم . نگران نباشید برای اجرای همه چیز تا معنا پیدا کنند . من سعی میکنم و هر سوالی توی نظرات بود رو جواب میدم ، و امیدوارم در ادامه سری همراه ما باشید !

مترجم : علی مرادی

ایمیل : [adeadmarshal@gmail.com](mailto:adeadmarshal@gmail.com)