اعداد در پایتون

این بخش درباره اعداد در پایتون و نحوه استفاده از آن‌ها است. لیست زیر مطالبی است که در این بخش خواهیم دید:

۱. انواع اعداد در پایتون
۲. عملیات عددی پایه‌ای
۳. تفاوت تقسیم معمولی و تقسیم صحیح
۴. انتسابِ یک شی در پایتون

انواع اعداد

اعداد در پایتون «انواع» مختلفی دارند. در اینجا هدف کلی روی اعداد «صحیح» و «ممیز» شناور است. اعداد صحیح یک عدد مثبت یا منفی هستند، بدون هیچ بخش اعشاری و یا نماد علمی. اعداد ممیز شناور در پایتون به دو گونه هستند: یا همراه بخش اعشاری و یا با یک بخش نمایی: `4.2` یا `4E15` به طور معمول با این جنس اعداد سر و کار داریم. هر چند که انواع عددی دیگری نیز در دسترس هستند، ولی فعلا به همین اعداد اکتفا می‌کنیم. در جدول زیر چند مثال آمده است:

Examples	Number "Type"
1,2,-5,1000	Integers
1.2,-0.5,2e2,3E2	Floating-point numbers

```
In [4]: a = 1234567889007.50656546456456456456
        a
```

```
Out[4]: 1234567889007.5066
```

حال بیایید چند عملیات ریاضی را در مرور کنیم:

عملیات پایه‌ای

```
In [6]: # Addition
        2+1
```

```
Out[6]: 3
```

```
In [3]: # Subtraction
        2-1
```

```
Out[3]: 1
```

```
In [4]: # Multiplication
        2*2
```

```
Out[4]: 4
```

```
In [5]: # Division
        3/2
```

```
Out[5]: 1.5
```

```
In [8]: # Floor Division
        7//4.0
Out[8]: 1.0
```

چه اتفاقی افتاد؟ به نظر ۷ تقسیم بر ۴ ۱/۷۵ می‌شد نه ۱؟ علت این است که عملگر `//` تقسیم جزء صحیح است و باقیمانده تقسیم را به بزرگترین عدد صحیح کوچکتر از آن گرد می‌کند. بگذارید چند مثال زیر را هم ببینیم:

```
In [9]: type(7)
Out[9]: int

In [10]: type(7/4)
Out[10]: float

In [11]: type(7.0)
Out[11]: float

In [12]: type(7//4)
Out[12]: int

In [15]: type(7//4.0)
Out[15]: float
```

حالا اگر باقیمانده را خواستیم چکار کنیم؟

```
In [10]: # Modulo
         7%4.0
Out[10]: 3.0
```

عملگر `%` باقیمانده تقسیم را برمی‌گرداند.

ادامه عملیات ریاضی

```
In [12]: # Powers
         2**3
Out[12]: 8

In [8]: # Can also do roots this way
        4**0.5
Out[8]: 2.0

In [9]: # Order of Operations followed in Python
        2 + 10 * 10 + 3
Out[9]: 105
```

In [10]:
```
# Can use parentheses to specify orders
(2+10) * (10+3)
```

Out[10]: 156

Variable Assignments

انتساب یک متغیر

تا الان ما فهمیدیم که از پایتون چطوری به صورت یک ماشین حساب استفاده کنیم. حالا بگذارید متغیر ایجاد کنیم. ما از یک علامت تساوی برای انتساب استفاده می‌کنیم.

In [16]:
```
# Let's create an object called "a" and assign it the number 5
a = 5
```

Now if I call *a* in my Python script, Python will treat it as the number 5.

In [17]:
```
# Adding the objects
a+a
```

Out[17]: 10

What happens on reassignment? Will Python let us write it over?

In [13]:
```
# Reassignment
a = 10.0
```

In [14]:
```
# Check
a
```

Out[14]: 10.0

Yes! Python allows you to write over assigned variable names. We can also use the variables themselves when doing the reassignment. Here is an example of what I mean:

In [20]:
```
# Check
a
```

Out[20]: 10

In [15]:
```
# Use A to redefine A
a = a + a
```

In [16]:
```
# Check
a
```

Out[16]: 20.0

The names you use when creating these labels need to follow a few rules:

> 1. Names can not start with a number.
> 2. There can be no spaces in the name, use `_` instead.
> 3. Can't use any of these symbols : `'` `"` `,` `<` `>` `/` `?` `|` `\` `(` `)` `!` `@` `#` `$` `%` `^` `&` `*` `~` `-` `+`
> 4. It's considered best practice (`PEP8`) that names are `lowercase` or `CapWord` .
> 5. Avoid using the characters `'l'` (lowercase letter el), `'O'` (uppercase letter oh), or 'I' (uppercase letter eye) as single character variable names.
> 6. Avoid using words that have special meaning in Python like " `list` " and " `str` "; if you are in force, make difference by underscor, like: " `class_` " and *avoid to misspell word* like " `clss` "

Using variable names can be a very useful way to keep track of different variables in Python. For example:

In [17]:
```
# Use object names to keep better track of what's going on in your code!
my_income = 100

tax_rate = 0.1

my_taxes = my_income*tax_rate
```

In [18]:
```
# Show my taxes!
my_taxes
```

Out[18]:
```
10.0
```

So what have we learned? We learned some of the basics of numbers in Python. We also learned how to do arithmetic and use Python as a basic calculator. We then wrapped it up with learning about Variable Assignment in Python.

Up next we'll learn about Strings!