

Python102

Python for Data Science Bootcamp

(3.1) Basics of Python Part 1

AIAT Academy

Python Basics' Outline (Part 1)

- Interactive Interpreter
- Comments
- Variable and Types
- Numbers and Booleans



Interactive Interpreter

```
terminal$ python3
```

```
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015,  
11:00:19)
```

```
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin  
Type "help", "copyright", "credits" or "license" for  
more information.
```

```
>>>
```

Interactive Interpreter

```
terminal$ python3
```

```
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
```

```
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin Type "help",  
"copyright", "credits" or "license" for more information.
```

```
>>>
```



We can write a line of Python code here!

Variables

Variables

```
x = 2
```

```
x * 7
```

```
>> 14
```

```
x = "Hello, "
```

```
x + "Python!"
```

```
>> "Hello, Python!"
```

Variables

```
x = 2
```

```
x * 7
```

Where is a semicolon!

```
>> 14
```

```
x = "Hello, "
```

```
x + "Python!"
```

Where is data type?

```
>> "Hello, Python!"
```

Variables in C/JAVA and Python

`int x = 0; // In C or JAVA`

`x = 0 # In Python!!`

Variable's naming

- Names are case sensitive and cannot start with a number. They can contain letters, numbers, and underscores.

bob Bob _bob _2_bob_ bob_2 BoB

- There are some reserved words:
and, assert, break, class, continue, def, del, elif,
else, except, exec, finally, for, from, global, if,
import, in, is, lambda, not, or, pass, print, raise,
return, try, while

Variable's Types

Variables in Python are **Dynamically-typed**

<code>type(1)</code>	<code># >> <class 'int'></code>
<code>type("สวัสดีครับ")</code>	<code># >> <class 'str'></code>
<code>type(None)</code>	<code># >> <class 'NoneType'></code>

Variable's Types

Variables in Python are **Dynamically-typed**

<code>type(1)</code>	<code># >> <class 'int'></code>
<code>type("สวัสดีครับ")</code>	<code># >> <class 'str'></code>
<code>type(None)</code>	<code># >> <class 'NoneType'></code>
<code>type(int)</code>	<code># >> <class 'type'></code>
<code>type(type(int))</code>	<code># >> <class 'type'></code>

Numbers and Math

Numbers and Math

3 # >> 3 (int)

3.0 # >> 3.0 (float)

Python has two numeric types
int and **float**

Numbers and Math

3 # >> 3 (int)

3.0 # >> 3.0 (float)

1 + 1 # >> 2

2 - 1 # >> 1

100 * 2 # >> 200

10 / 5 # >> 2.0

10 / 4 # >> 2.5

Numbers and Math

3	# >> 3 (int)
3.0	# >> 3.0 (float)
1 + 1	# >> 2
2 - 1	# >> 1
100 * 2	# >> 200
10 / 5	# >> 2.0
10 / 4	# >> 2.5
7 // 3	# >> 2 (integer division) หารพิเศษ
7 % 3	# >> 1 (integer modulus)
2 ** 5	# >> 32 (exponentiation)

Booleans

Booleans

True

False

>> True

>> False

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Boolean is a subtype of `int`, where
`True == 1` and `False == 0`

Booleans

True

```
# >> True
```

False

```
# >> False
```

not True

```
# >> False
```

True and False

```
# >> False
```

True or False

```
# >> True (Short-circuit)
```

1 == 1

```
# >> True
```

2 * 2 == 5

```
# >> False
```

1 != 2

```
# >> True
```

4 * 3 != 1

```
# >> False
```

Booleans

True

```
# >> True
```

False

```
# >> False
```

not True

```
# >> False
```

True and False

```
# >> False
```

True or False

```
# >> True (Short-circuit)
```

1 == 1

```
# >> True
```

2 * 2 == 5

```
# >> False
```

1 != 2

```
# >> True
```

4 * 3 != 1

```
# >> True
```

1 < 3

```
# >> False
```

1 < 5 < 10

```
# >> True (1 < 5 and 5 < 10)
```

Comments

#

Single line comments start with a '#'

"""

Multiline strings can be written
using three "s, and are often used as
function and module comments

"""