01-HelloPython-

September 8, 2020

1 Python tutorial #1

1.1 Hello World!

[1]:
```
print('Hello World')
print('Hello World {} + {} = {}'.format(2, 3, 2+3))
```

```
Hello World
Hello World 2 + 3 = 5
```

1.2 Basic data types

[2]:
```
x = 3
print(type(x)) # Prints "<class 'int'>"
print(x) # Prints "3"
```

```
<class 'int'>
3
```

[3]:
```
print(x + 1) # Addition; prints "4"
print(x - 1) # Subtraction; prints "2"
print(x * 2) # Multiplication; prints "6"
print(x ** 2) # Exponentiation; prints "9"
```

```
4
2
6
9
```

1.3 For statement

range:

[4]:
```
A = range(5)
print(A)
```

```
range(0, 5)
```

- A

[5]:
```
print(A[2])
```

```
2
```

[7]:
```
for i in range(5):
  #print(i, a[i])
  print('{} ----- {}'.format(i,A[i]))
```

```
0 ----- 0
1 ----- 1
2 ----- 2
3 ----- 3
4 ----- 4
```

Exercise

[8]:
```
for i in range(9):
    print('{} x {} = {}'.format(2, i+1, 2*(i+1)))
```

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
```

1.4 Operators

[10]:
```
print((1,2,3) * 3)
print([1, 2, 3] * 3)
print('Hello ' * 3)
```

```
(1, 2, 3, 1, 2, 3, 1, 2, 3)
[1, 2, 3, 1, 2, 3, 1, 2, 3]
Hello Hello Hello
```

1.5 Containers

Python includes several built-in container types: lists, dictionaries, sets, and tuples.

1.5.1 Tuple

A simple immutable (,) ordered sequence of items

[11]:
```
# -*- coding: utf-8 -*-
# creating a tuple

months = ('January','February','March','April','May','June',\
'July','August','September','October','November','December')

print(months[0])
print("index of 7 ==> " , months[7])
```

```
January
index of 7 ==>  August
```

[12]:
```
# iterate through them:
for item in months:
 print (item)
```

```
January
February
March
April
May
June
July
August
September
October
November
December
```

[13]:
```
t = ('john', 32, (2,3,4,5), 'hello')
print(t)
print(t[2])
print(t[2][1])
print(t[:2]) # index  X
print(t[2:]) # index  O
print(t[-1])
print(t[-2])
```

```
('john', 32, (2, 3, 4, 5), 'hello')
(2, 3, 4, 5)
3
('john', 32)
((2, 3, 4, 5), 'hello')
hello
(2, 3, 4, 5)
```

1.5.2 List

Mutable( ) ordered sequence of items of mixed types

[14]:
```
li = ['hallym', 1, 3.141572, 'hello']
print(li)
```

```
['hallym', 1, 3.141572, 'hello']
```

[15]:
```
li[1] = 45
print(li)
```

```
['hallym', 45, 3.141572, 'hello']
```

[16]:
```
li.append('September')
print(li)
```

```
['hallym', 45, 3.141572, 'hello', 'September']
```

(append)

-

[17]:
```
v = []
```

-

[18]:
```
for i in range(0,3):
 v.append(i*5)
 print(i, v)
```

```
0 [0]
1 [0, 5]
2 [0, 5, 10]
```

+

[19]:
```
print((1, 2, 3) + (4, 5, 6))
print([1, 2, 3] + [4, 5, 6])
print("Hello" + " " + "World")
```

```
(1, 2, 3, 4, 5, 6)
[1, 2, 3, 4, 5, 6]
Hello World
```

*** **

The * operator produces a new tuple, list or string that "repeats" the original content.

[20]:
```
y = 2.5
print(type(y)) # Prints "<class 'float'>"
print(y, y + 1, y * 2, y ** 2) # Prints "2.5 3.5 5.0 6.25"
```

```
<class 'float'>
2.5 3.5 5.0 6.25
```

1.5.3 Enumeration ()

[21]:
```
for i, val in enumerate(v):
 print('{} ---> {}'.format(i, val))
```

```
0 ---> 0
1 ---> 5
2 ---> 10
```

[22]:
```
v2 = [ 'A', 'B', 'C', '0', '1', '2', '3']
print(v2)
```

```
['A', 'B', 'C', '0', '1', '2', '3']
```

[23]:
```
for i, val in enumerate(v2):
 print('{} ---> {}'.format(i, val))
```

```
0 ---> A
1 ---> B
2 ---> C
3 ---> 0
4 ---> 1
5 ---> 2
6 ---> 3
```

[ ]:
```

```