

## Python Tips and tricks: "Use PyCharm IDE"

1. In place swapping of two numbers:

```
x, y = 10, 20
print(x, y)

x, y = y, x
print(x, y)

#1 (10, 20)
#2 (20, 10)
```

2. Chaining of comparison operators:

```
n = 10
result = 1 < n < 20
print(result)

# True

result = 1 > n <= 9
print(result)

# False
```

3. Use of ternary operator

```
[on_true] if [expression] else [on_false]
```

- a. for conditional assignment:

```
def small(a, b, c):
    return a if a <= b and a <= c else (b if b <= a and b <= c else c)

print(small(1, 0, 1))
print(small(1, 2, 2))
print(small(2, 2, 3))
print(small(5, 4, 3))

#Output
#0 #1 #2 #3
```

- b. for list comprehension:

```
[m**2 if m > 10 else m**4 for m in range(50)]

#=> [0, 1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000, 121, 144, 169, 196, 225, 256, 289,
324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900, 961, 1024, 1089, 1156, 1225,
1296, 1369, 1444, 1521, 1600, 1681, 1764, 1849, 1936, 2025, 2116, 2209, 2304, 2401]
```

4. Dictionary/Set Comprehension:

```
testDict = {i: i * i for i in xrange(10)}
testSet = {i * 2 for i in xrange(10)}

print(testSet)
print(testDict)

#set([0, 2, 4, 6, 8, 10, 12, 14, 16, 18])
#{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
```

5. Simplify If statement

```
if m in [1,3,5,7]:
```

instead of:

```
if m==1 or m==3 or m==5 or m==7:
```

6. Using enumeration method: To find an index while you're inside a loop.

```
testlist = [10, 20, 30]
for i, value in enumerate(testlist):
    print(i, ': ', value)

#1-> 0 : 10
#2-> 1 : 20
#3-> 2 : 30
```

7. Return multiple values from functions

```
# function returning multiple values.
def x():
    return 1, 2, 3, 4

# Calling the above function.
a, b, c, d = x()

print(a, b, c, d)

#-> 1 2 3 4
```

8. Create dictionary from two related sequences –

```
t1 = (1, 2, 3)
t2 = (10, 20, 30)

print(dict (zip(t1,t2)))

#-> {1: 10, 2: 20, 3: 30}
```

9. Modules

```
# 1- Module definition => save file as my_function.py
def minmax(a,b):
    if a <= b:
        min, max = a, b
    else:
        min, max = b, a
    return min, max
```

```
# 2- Module Usage
import my_function
x,y = my_function.minmax(25, 6.3)

print(x)
print(y)
```

10. Zipping and unzipping lists and iterables

```
>>> a = [1, 2, 3]
>>> b = ['a', 'b', 'c']
>>> z = zip(a, b)
>>> z
[(1, 'a'), (2, 'b'), (3, 'c')]
>>> zip(*z)
[(1, 2, 3), ('a', 'b', 'c')]
```