

# Python Tips and Tricks for Writing Better Code

## Tip1 : Ternary Conditionals

A **ternary operator** that is part of the syntax for basic conditional expressions in several programming languages. An expression `a ? b : c` evaluates to `b` if the value of `a` is true, and otherwise to `c`.

```
x = 1 if conditions else 0
```

## Tip2 : Underscore Placeholders

We can make big number easier to read using underscore placeholder

```
num1 = 10_000_000_000
num2 = 100_000_000

total = num1 + num2
print(f"{total:,}")
# 10,100,000,000
```

## Tip3 : Context Managers

Using context manager, we can manage resources automatically. We can applying this statement to file object, database connection, and so on.

```
with open('test.txt', 'r') as f:
    file_contents = f.read()
```

## Tip4 : Enumerate

The enumerate statement is used when we want a counter while we are looping over something.

```
names = ['Corey', 'Chris', 'Dave', 'Travis']

for i, name in enumerate(names, start=1):
    print(i, name)

# 1 Corey 2 Crhis 3 Dave 4 Travis
```

## Tip5 : Zip

The zip function makes looping two lists at once. Zip will stop if the shortest list is exhausted.

```
names = ['Peter Parker', 'Clark Kent', 'Wade Wilson', 'Bruce wayne']
heroes = ['Spiderman', 'Superman', 'Deadpool', 'Batman']
universes = ['Marvel', 'DC', 'Marvel', 'DC']

for name, hero, universe in zip(names, heroes, universes):
    print(f"{name} is actually {hero} in {universe}.")

# Peter Parker is actually Spiderman in Marvel.
```

## Tip6 : Unpacking

```
# Unpacking
a, b = (1, 2)
print(a)
print(b)
# 1 2
```

We can set prefix of variable name as underscore denoting this variable will not be used.

```
a, _ = (1, 2)
print(a)
# 1
```

By setting asterisk in front of variable name, we can unpack multiple remainders.

```
a, b, c = (1, 2, 3, 4, 5)
print(a)
print(b)
print(c)
# 1 2 [3, 4, 5]
```

## Tip7 : Setattr/Getattr

The setattr() built in function sets the value of the attribute of an object.

```
class Person():
    pass

person = Person()

first_key = 'first'
first_val = 'Corey'
setattr(person, first_key, first_val)

print(person.first)
# Corey
```

We can get the attribute of an object

```
first = getattr(person, first_key)
print(first)
```

## Tip8 : GetPass

Many programs that interact with user via the terminal need to ask the user for password values without showing the user types on the screen. The getpass module provides a portable way to handle such password prompts securely.

```
from getpass import getpass

username = input("username :")
password = getpass("password :")
```

## Tip9 : Python dash m

python -m will search system.path for the named module and executes its contents as the main module.

```
CLI
$ python -m password
```

## Tip10 : Help/Dir

The dir() function is the powerful inbuilt function in Python that returns a list of valid methods and ascribe of any objects including modules, lists, strings, and others.

The help() function returns the assistance explanations that linked to the Python module, object, and method but if it called with the approachable arguments.

```
import os
print(help(os))
print(dir(os))
```