

Python Basic & Data Structure

VARIABLE ASSIGNMENT & DATA TYPE DETECTION

create a variable

```
full = 1
```

data type detection

```
print (type( variable_name))
```

PYTHON LIST

create a list

```
my_list = ["my ", " list", 0, 1 ]
```

getting elements in a list

```
x = ["a", " b", " c", " d"]
```

```
x[1]
```

```
x[-3] # same result!
```

slicing & dicing

```
my_list[start:end]
```

manipulating a list (update)

```
fam = ["liz", 1.73, "emma", 2]
```

```
fam[3] = 1.86
```

manipulating a list (remove)

```
del(fam[2]) # Remove "emma"
```

manipulating a list (add element)

```
x = ["a", " b", " c", " d"]
```

```
y = x + ["e", " f"]
```

PYTHON SET

create a set

```
myset = {"apple ", " banana ", " cherry "}
```

PYTHON TUPLE

create a tuple

```
mytuple = ("apple ", " banana ", " cherry ")
```

getting elements in a tuple

```
mytuple[1]
```

slicing & dicing in a tuple

```
mytuple[start:end]
```

Python if else condition

Phân biệt Python if elif và (multiple) if condition

Data Structure (Dictionary)

PYTHON DICTIONARY

create dictionary

```
d1 = {1: 'Geeks', 2: 'For', 3: 'Geeks'} # create dictionary
```

```
k = ["Fruit s", " Vegetable s", " Drinks "]
```

```
val = [{"Apple", " Banana "}, {"Carrot", " Spinach "}, {"Juice"}]
```

```
d = dict(zip(k, val)) print(d) # create a dictionary
```

get list of values, keys

```
dictionary_name.values()
```

```
dictionary_name.keys()
```

access specific value of key

```
dictionary_name['specific_key']
```

dictionary manipulation (add key & value)

```
dictionary_name['new_key'] = 'new_value'
```

dictionary update / remove key

```
dictionary_name['key_to_update'] = 'value_to_update'
```

```
del dictionary_name['key_to_remove']
```

Data Structure (String)

Python Loop

For: lặp đúng bằng số lần của tập hợp, danh sách

```
fruits = ["ap ple ", " ban ana ", " che rry ", "st -  
raw ber ry"]
```

```
for i in fruits:
```

```
    print(i)
```

While: cho đến khi condition = True

```
i = 0
```

```
while i < 6:
```

```
    i= i + 2
```

```
print(i)
```

PYTHON STRING

create a string

```
s = " GfG "
```

access character in string

```
s[1]
```

string slicing

```
s[start : end]
```

common string methods (len of string)

```
s = " Gee ks f orG eek s"
```

```
print( len(s))
```

output: 13

common string methods (upper, lower)

```
s = " Hello World"
```

```
print( s.u pper())    # output: HELLO WORLD
```

```
print( s.l ower())    # output: hello world
```

common string methods (strip)

```
s = "   Gfg   "
```

```
print( s.s trip()) # remove space from s
```

output: Gfg

another string methods

reference to this link: https://www.w3schools.com/python/python_ref_string.asp



By [hangvtk7777](#)

cheatography.com/hangvtk7777/

Published 18th February, 2025.

Last updated 18th February, 2025.

Page 1 of 2.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>