Python Data Types

Data Types	Classes	Description
Numeric	int, float, complex	holds numeric values
String	str	holds sequence of characters
Sequence	list, tuple, range	holds collection of items
Mapping	dict	holds data in key-value pair form
Boolean	bool	holds either `True` or `False`
Set	set, frozeenset	hold collection of unique items

Since everything is an object in Python programming, data types are actually classes and variables are instances(object) of these classes.

Python Numeric Data type

In Python, numeric data type is used to hold numeric values.

Integers, floating-point numbers and complex numbers fall under Python numbers category. They are defined as `int`, `float` and `complex` classes in Python.

- `int` - holds signed integers of non-limited length.
- `float` - holds floating decimal points and it's accurate up to **15** decimal places.
- `complex` - holds complex numbers.

We can use the `type()` function to know which class a variable or a value belongs to.
Let's see an example,

```
num1 = 5

print(num1, 'is of type', type(num1))

num2 = 2.0

print(num2, 'is of type', type(num2))

num3 = 1+2j

print(num3, 'is of type', type(num3))
```

OUTPUT

```
5 is of type <class 'int'>

2.0 is of type <class 'float'>

(1+2j) is of type <class 'complex'>
```

Python Set Data Type

Set is an unordered collection of unique items. Set is defined by values separated by commas inside braces `{ }`. For example,

```
# create a set named student_id

student_id = {112, 114, 116, 118, 115}

# display student_id elements
```

```
print(student_id)

# display type of student_id

print(type(student_id))
```

Here, we have created a set named `student_info` with **5** integer values. Since sets are unordered collections, indexing has no meaning. Hence, the slicing operator `[]` does not work.

Python Variables, Constants and Literals

Python Variables

In programming, a variable is a container (storage area) to hold data. For example,

```
Number = 10
```

Here, `number` is the variable storing the value **10**.

Assigning values to Variables in Python

As we can see from the above example, we use the assignment operator `=` to assign a value to a variable.

```
# assign value to site_name variable

site_name = 'google.com'

print(site_name)

# Output: google.com
```

In the above example, we assigned the value `'google.com'` to the `site_name` variable. Then, we printed out the value assigned to `site_name`.

Changing the Value of a Variable in Python

```
# assigning a new value to site_name
site_name = 'apple.com'

print(site_name)
```

Example: Assigning multiple values to multiple variables

```
a, b, c = 5, 3.2, 'Hello'

print(a)  # prints 5
print(b)  # prints 3.2
print(c)  # prints Hello
```

Rules for Naming Python Variables

- Constant and variable names should have a combination of letters in lowercase (a to z) or uppercase (**A to Z**) or digits (**0 to 9**) or an underscore `(_)`. For example:

```
snake_case
MACRO_CASE
camelCase
CapWords
```

- Create a name that makes sense. For example, `vowel` makes more sense than `v`.
- If you want to create a variable name having two words, use underscore to separate them. For example:

```
- my_name
- current_salary
```

- Python is case-sensitive. So `num` and `Num` are different variables. For example,

```
var num = 5
var Num = 55
print(num) # 5
print(Num) # 55
```

- Avoid using keywords like `if`, `True`, `class`, etc. as variable names.

Python Constants

A constant is a special type of variable whose value cannot be changed.

In Python, constants are usually declared and assigned in a module (a new file containing variables, functions, etc which is imported to the main file). Let's see how we declare constants in separate file and use it in the main file,

Create a **constant.py**:

```
# declare constants
PI = 3.14
GRAVITY = 9.8
```

Create a **main.py**:

```
# import constant file we created above
import constant

print(constant.PI) # prints 3.14
print(constant.GRAVITY) # prints 9.8
```

In the above example, we created the **constant.py** module file. Then, we assigned the constant value to `PI` and `GRAVITY`

Python program to swap two variables

x = 5

y = 10

To take inputs from the user

#x = input('Enter value of x: ')

#y = input('Enter value of y: ')

create a temporary variable and swap the values

temp = x

```
x = y
y = temp

print('The value of x after swapping: {}'.format(x))
print('The value of y after swapping: {}'.format(y))



# Python Program to calculate the square root

# Note: change this value for a different result
num = 8

# To take the input from the user
#num = float(input('Enter a number: '))

num_sqrt = num ** 0.5
print('The square root of %0.3f is %0.3f'%(num ,num_sqrt))
```