# Basic Data type and Function Cheatsheet

Here's a quick reference for data types in Python.

## Data types

| Data type | Meaning | Example |
|---|---|---|
| string | Text | 'Hello', 'Testing 123' |
| integer | Numbers | -5, 4, 3, 2, 0 |
| float | Decimals | 2.4, 5.2, 1000.00 |

## Flow Control

### Comparison operators

| Operator | Meaning | Example |
|---|---|---|
| == | Equals | a == b |
| != | Not Equal | a != b |
| < | Less than | a < b |
| > | Greater than | a > b |
| <= | Less than or Equal to | a <= b |
| >= | Greater than or Equal to | a >= b |

## Comments

### Single-line comments

Placing a # symbol in front of the text you want to be a comment causes Python to ignore everything from that point until the end of the current line.

```
1   # Single Line comment
2
```

Run

Reset

### Multi-line comments

Python does not really have a method for multi line comments, so a # symbol can be used on every line.

```
1   # This is a multiline comment
2   # which can be used for long comments
3
```

Run

Reset

### Inline/code comments

The # symbol will cause Python to ignore everything from that point until the end of the current line, so inline comments can be created in this way.

```
1   x = 1 # assigns value of 1 to x
2
```

Run

Reset

## Built-in Functions

### print()

This function looks for the default output device, your terminal, and displays the value passed to it.

```
print("Hello")

```

Run

Reset

### input()

This function looks for the default input device, your keyboard, and captures the value. This value can then be assigned or used.

```
print("Where do you live?")
location = input()
print("So you live in " + location)
```

Run

Reset

### len()

This function returns the length or the count of the elements contained within the structure it is applied on. This may be a string, array, list, tuple, dictionary or any sequence.

```
len("Hello")
5
```

Run

Reset

### str()

This function can be used to convert the provided value into a **`String`**

```
str(55)
'55'

```

Run

Reset

### int()

This function can be used to convert the provided value into an **`int`**

```
int('75')
75
```

Run

Reset

### float()

This function can be used to convert the provided value into a **`float`**

```
some_int = 10
float(some_int)
10.0

```

Run

Reset

## Creating Functions

Functions in Python require a keyword to define them : **`def`** followed by an identifier (a name) this forms the function signature. The body of the function contains the code to run when the function is called.

```
1
2   def say_hello():
3       return "Hello there!"
4
5   # With parameters
6   def say_hello(you):
7       return "Hello " +  you
8
```