

Before we start, let us see how to print and read from user

```
In [1]: print("We can print using the print statement")
        print("We ", "can ", "also ", "print multiple things together")
```

We can print using the print statement
We can also print multiple things together

```
In [2]: # We can take input from user and print it as well.
        name = input("What's your name? ")
        print("Hello!", name)
```

What's your name? Abhishek
Hello! Abhishek

Variables and Operators

We will learn about how to declare variables and use operators on them. We will use the following operators (there are more out there!)

- Assignment (=)
- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Division (/)
- Equality (==)

```
In [3]: # Declare two variables
        number_1 = 10
        number_2 = 5
```

```
In [4]: # Add two numbers
        summation = number_1 + number_2
        print("Addition of numbers is", summation)
```

Addition of numbers is 15

```
In [5]: # Subtract two numbers
        subtraction = number_1 - number_2
        print("Subtraction of numbers is", subtraction)
```

Subtraction of numbers is 5

```
In [6]: # Multiply two numbers
        multiplication = number_1 * number_2
        print("Multiplication of numbers is", multiplication)
```

Multiplication of numbers is 50

```
In [7]: # Divide two numbers
        division = number_1 / number_2
        print("Division of numbers is", division)
```

Division of numbers is 2.0

```
In [8]: # We can check if two things are equal using ==
        print(1 == 1)
        print(1 == 2)
```

True
False

Data types

Data types define what type of data is stored in a variable. We have following data types in python.

- integer
- float
- string
- boolean

```
In [9]: # This is integer data type, which is used to store integer values
        integer_1 = 10
        integer_2 = 20
```

```
In [10]: # This is float data type, which can be used to store decimal values
decimal_1 = 1.5
decimal_2 = 3.5

In [11]: # This is string data type, which is used to store texts
string_1 = "Hello stranger!"
string_2 = "I dont know your name"

In [12]: # This is boolean data type, which stores only 2 values, true or false
boolean_1 = True
boolean_2 = False

In [13]: # We can add, subtract, multiply, divide two integers together, or a float and integer, or a float and float
integer_integer_sum = integer_1 + integer_2
integer_float_difference = integer_1 - decimal_1
float_float_multiply = decimal_1 * decimal_2
integer_integer_divide = integer_2 / integer_1

print(integer_integer_sum, integer_float_difference, float_float_multiply, integer_integer_divide)

30 8.5 5.25 2.0
```

Lists, Dictionary and set

Lists

A list is simply a list of different things(like string, integers, decimal numbers)

```
In [14]: # Let us create a simple list
list_1 = ["This is the first element", 2, 3.3, "Fourth element"]
print("Entire list is: ", list_1)

Entire list is:  ['This is the first element', 2, 3.3, 'Fourth element']

In [15]: # A List is numbered starting from zero, weird right! And so on..
first_element = list_1[0]
second_element = list_1[1]
print("First element of list is: ", first_element)
print("Second element of list is: ", second_element)

First element of list is:  This is the first element
Second element of list is:  2

In [16]: # We can add an element to list using append
list_1.append(5)
print("Updated list_1 is: ", list_1)

Updated list_1 is:  ['This is the first element', 2, 3.3, 'Fourth element', 5]

In [17]: # we can also remove a particular element from list
list_1.remove(3.3)
print("3.3 is now removed from the list: ", list_1)

3.3 is now removed from the list:  ['This is the first element', 2, 'Fourth element', 5]

In [18]: # we can also remove pop to remove element from a particular position
list_1.pop(3)
print("Removed the 4th element from list_1: ", list_1)

Removed the 4th element from list_1:  ['This is the first element', 2, 'Fourth element']
```

Dictionary

A dictionary is like a mapping, where we map a key to a value.

```
In [19]: # Let us create a simple dictionary
dictionary_1 = {"key1": "value1", 2: "value2", "list": [1, "We can store list inside dictionary"]}
print("Entire dictionary is: ", dictionary_1)

Entire dictionary is:  {'key1': 'value1', 2: 'value2', 'list': [1, 'We can store list inside dictionary']}

In [20]: # In order to get the value for particular key, we can use the following syntax
key1_value = dictionary_1["key1"]
print("Value associated with key key1 is: ", key1_value)

Value associated with key key1 is:  value1

In [21]: # We can also reference the second element of list in the dictionary in the following way
second_list_element = dictionary_1["list"][1]
```

```
print("Second element of list in the dictionary is: ", second_list_element)
```

Second element of list in the dictionary is: We can store list inside dictionary

```
In [22]: # We can easily add another key with value in the dictionary by assigning the value
dictionary_1["new_key"] = "new_value"
print("Dictionary after key addition is: ", dictionary_1)
```

Dictionary after key addition is: {'key1': 'value1', 2: 'value2', 'list': [1, 'We can store list inside dictionary'], 'new_key': 'new_value'}

```
In [23]: # Similar to list, we can use pop to delete a key and its value from dictionary
dictionary_1.pop("new_key")
print("Dictionary after key deletion is: ", dictionary_1)
```

Dictionary after key deletion is: {'key1': 'value1', 2: 'value2', 'list': [1, 'We can store list inside dictionary']}

Sets

A set contains unique elements. We can store things of any data type.

```
In [24]: # Let us create an empty set
empty_set = set()
print("This is an empty set: ", empty_set)
```

This is an empty set: set()

```
In [25]: # We can also create a set the following way if we have certain initial values
new_set = {"Hello", 2, 3.3}
print("New set is: ", new_set)
```

New set is: {3.3, 2, 'Hello'}

```
In [26]: # we convert list_1 into a set
set_from_list = set(list_1)
print("Converted set from list is: ", set_from_list)
```

Converted set from list is: {2, 'Fourth element', 'This is the first element'}

```
In [27]: # Unlike list, we cannot have duplicates in set
# we add 2 to new_set which already has 2. There is no effect.
new_set.add(2)
print("Set after addition is: ", new_set)
```

Set after addition is: {3.3, 2, 'Hello'}

```
In [28]: # To remove a element from set, we can use discard
new_set.discard(3.3)
print("Set after deletion is: ", new_set)
```

Set after deletion is: {2, 'Hello'}

Conditions

Conditions are important as they tell us when we need to perform some operations.

For example, if we want say hello to person with name Jane, but hey to any other person, we can do this using a condition block

```
In [29]: # In python, we use 'if' to represent a condition. Let us try this with an example
name = input("What's your name? ")
```

What's your name? Abhishek

```
In [30]: # We use == operator to check if two things are same
if name == "Jane":
    print("Hello! Jane")
else:
    print("Hey!", name)
```

Hey! Abhishek

```
In [31]: # We can also use them to check if an element exists in a list
if 2 in list_1:
    print("We found 2 in list_1")
else:
    print("Oops! 2 is not in list_1")
```

We found 2 in list_1

```
In [32]: # This works in dictionaries and set as well!
```

```
if "key1" in dictionary_1:  
    print("We found key1 in dictionary_1")  
else:  
    print("Oops! key1 doesn't exist in dictionary_1")  
  
if 3.3 in new_set:  
    print("We found 3.3 in new_set")  
else:  
    print("Oops! 3.3 doesn't exist in new_set")
```

We found key1 in dictionary_1
Oops! 3.3 doesn't exist in new_set