

PYTHON CODE SHEET

BASICS

Print

Prints a string into console

```
print("Hello World")
```

Input

Prints a string into the console,
and ask the user for a string input.

```
input('Enter your age')
```

Comments

Adding a # symbol in font of text
lets you make comments on a line of code.
The computer will ignore your comments.

```
#This is a comment  
print("This is code")
```

Variables

A Variable give a name to a piece of data.
Like a box with a label, it tells you what's
inside the box.

```
my_name = "Ravi"  
my_age = 15
```

PYTHON CODE SHEET

DATA TYPES

Integers

Integers are whole numbers.

```
my_number = 354
```

Floating Point Numbers

Floats are numbers with decimal places. when you do a calculation that results in a fraction e.g. $4 \div 3$ the result will always be a floating point.

```
my_float = 3.14149
```

Strings

A string is just a string of characters it should be surrounded by double quotes

```
my_string = "Hello"
```

Booleans

They represent the truth values **False** and **True**. The Boolean type is a subtype of plain integers, and Boolean values **False** and **True** behave like the values **0** and **1**, respectively.

```
A = True
```

```
B = bool(0)
```

```
#The value of B is False
```

Complex Numbers

Complex number is represented by complex class. It is specified as (**Real part**) + (**Imaginary part**). Python represents complex as a pair of floating point numbers.

```
com = 2 + 3j
```

PYTHON CODE SHEET

DATA TYPES

Lists

A List in Python represents a group of comma separated value of any data type between square brackets.

```
L=[1,2,3,4,5]  
A=['a','e','i','o','u']
```

Tuples

Tuples are represented as group of comma-separated values of any data type within parantheses .

```
p=(0,2,4,6,8)  
q=('a','e','i','o','u')
```

Dictionaries

The dictionaries is an unsorted set of comma separated **key : value** pairs, within { } with the requirement that within a dictionary, no two keys can be same.

```
A={'a':1,'e':2,'i':3}
```

NOTE :- the **type()** function is used to determine the type of data type.
e.g. if `a=5`; `type(a)` will return `<class 'int'>`.

PYTHON CODE SHEET

STRING OPERATION

String Concatenation

You can add strings to create a new string. This is called concatenation. It results in a new string.

```
"Hello" + "Ravi"
```

String Replication

To use a * operator with strings, you need two types of operand : a string and a number, i.e., as **number * string** or **string * number**, where string operand tells the string to be replicated and number operand tells the number of times, it has to be printed.

```
3*'Hi '  
#will return 'Hi Hi Hi'
```

String Slicing

String Slice refers to part of a string containing some contiguous characters from the string.

```
s='Hello World'  
print(s[6:11])  
# will print 'World'
```

PYTHON CODE SHEET

BASIC TECHNIQUES

Escaping a String

As the double quotes are special, it denotes a String, if you want to use it in a String, you need to escape it with a "\\".

```
word="She said \"Hi\" "  
print (word)  
# prints: She said "Hi"
```

F-Strings

You can insert a variable into a string using f-strings. The syntax is simple, just insert the variable in-between a set of curly braces {}.

```
days=365  
print(f'There are{days}  
in a year')
```

Converting Data Types

You can convert a variable from 1 Data type to another.

Converting to float:

`float()`

Converting to integer:

`int()`

Converting to string:

`str()`

```
n=123  
new=float(n)  
print(new)#result 123.0
```

Checking Data Types

You can use the `type()` function to check. to check what is the data type of a particular variable.

```
n=3.14159  
type(n) #result float
```

PYTHON CODE SHEET

OPERATORS

Arithmetic Operators

You can do Mathematical calculation with Python as long as you know the right operators.

Operator	Meaning	Example
+	Add two operands or unary plus	$x + y + 2$
-	Subtract right operand from the left or unary minus	$x - y - 2$
*	Multiply two operands	$x * y$
/	Divide left operand by the right one (always results into float)	x / y
%	Modulus - remainder of the division of left operand by the right	$x \% y$ (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line	$x // y$
**	Exponent - left operand raised to the power of right	$x ** y$ (x to the power y)

PYTHON CODE SHEET

Assignment Operators

Arithmetic operators are used in Python to assign values to variables.

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5

Logical Operators

Logical Operators are **and**, **or**, **not** operators.

Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

PYTHON CODE SHEET

Comparison Operators

Comparison operators are used to compare values. It returns either True or False according to the condition.

Operator	Meaning	Example
>	Greater than - True if left operand is greater than the right	<code>x > y</code>
<	Less than - True if left operand is less than the right	<code>x < y</code>
==	Equal to - True if both operands are equal	<code>x == y</code>
!=	Not equal to - True if operands are not equal	<code>x != y</code>
>=	Greater than or equal to - True if left operand is greater than or equal to the right	<code>x >= y</code>
<=	Less than or equal to - True if left operand is less than or equal to the right	<code>x <= y</code>

PYTHON CODE SHEET

CONDITIONALS

If

This is the basic syntax to test if a condition is true. If so, the indented code will be executed, if not it will be skipped.

```
n=5
if n>2:
    print('More than 2')
```

The Syntax of if statement is given below:

```
if test expression:
    statement(s)
```

Else

This is way to specify some code that will be executed if a condition false

```
age=21
if age > 18:
    print('Can drive')
else :
    print('Dont drive')
```

The Syntax of else statement is given below:

```
if test expression:
    Body of if
else:
    Body of else
```

PYTHON CODE SHEET

Elif

In addition to the initial If statement condition, you can add extra condition to test if the first condition is false.

Once an elif condition is true, the rest of the elif condition are no longer checked and are skipped.

The Syntax of elif statement is given below:

```
if test expression:
    Body of if
elif test expression:
    Body of elif
else:
    Body of else
```

```
w = 'sunny'
if w == 'rain':
    print("Umbrella")

elif w == 'sunny':
    print('Sunglasses')
elif w == 'snow':
    print('Gloves')
```

PYTHON CODE SHEET

LOOPS

While Loop

This is a loop that will keep repeating itself
Until the while condition becomes false.

```
a=5  
while n<100:  
    n += 1
```

The Syntax of **while** loop in python is given below:

```
while expression:  
    statements
```

For Loop

For loop give you more control than while loops.
executed if a condition false.

```
String='Hello'  
for i in String:  
    print(i)
```

The Syntax of **for** loop in python is given below:

```
for iterating_var in sequence:  
    statement(s)
```