

Summary of Everything Learnt in Python Class

Python Introduction

Python is an interpreted high-level general-purpose programming language. It is object-oriented which helps programmers to write easy to understand code for small and large scale projects. Python is one of the most popular programming language due to its simpler syntax.

Uses of Python

1. Used on a server to create web applications.
2. Used alongside software to create workflows.
3. Used to connect to database systems and also read and modify files.
4. Used to handle big data and perform complex mathematics.
5. Used for prototyping or for production-ready software development.

Getting started with Python

To check if python is installed in your pc run the following in the Command Line or Terminal.

```
python --version
```

Launching your codes written in python with extension .py from command line.

```
Python [filename].py
```

Writing your first piece of code in python.

```
print("Hello World!") or print ('Hello World!')
```

This gives us the output (Hello World!)

Summary of Everything Learnt in Python Class

Comments

Comments are used in Python code to explain what the code does which makes the code more readable and also to prevent execution when testing code. It is useful to write comments for your codes as it makes easier for other programmers working on the same code to understand about the code and apply some changes in the code.

Creating Comment

- Single-line comment:

Initialize # before something you want to comment. E.g:

```
#This is a single line comment
```

Above shown example is a way to initialize a single line comment. # tells the python to ignore the rest of the line that begins with #.

- Multi-line comment:

To initialize a multi-line comment we enclose the lines in “"""(lines to be commented)"""”. Eg:

```
"""
    This is a
    multi-line comment.
"""
```

Variables

In simpler terms variables can be defined as containers to store value. Variables in python can store numbers, strings, character and Boolean values.

```
A = 5
```

```
print(A)
```

```
#Gives the output 5.
```

```
a = "Alaska"
```

```
print(a)
```

```
#Gives the output Alaska.
```

Summary of Everything Learnt in Python Class

Steps to declare a variable

1. Variables can be combination of alphabets, digits or underscores.
2. The first character of variable must be an alphabet or underscore.
3. There should be no special characters i.e. !, -, * e.t.c. except '_' in a variable name.
4. There should be no commas or spaces in the variable name.

Some examples of legal variables are: myvar, my_var, _, MyVar, _my_var, myvar2 etc.

Some effective methods to write a variable:

- **Camel case:** camelCase = 1, firstName = "Ram"
- **Pascal case:** PascalCase = "Athena"
- **Snake case:** snake_case = "Solo Leveling"

Some examples of illegal variables are: my*var, my#var, 1myvar e.t.c.

Declaring many values in multiple variables:

```
x, y, z = "orange", 399.99, True
```

Declaring one value to multiple variables:

```
x, y, z = 100
```

Strings

String in python are surrounded by either single quotation marks, or double quotation marks. Examples of string in python:

```
string1 = """Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua."""
```

```
string2 = '''Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.'''
```

Working with strings in python

```
txt = "The best things in life are free!"
```

```
print("expensive" in txt) #gives output False.
```

Summary of Everything Learnt in Python Class

'in' checks if expensive lies in txt and returns the Boolean value either true or false.

```
print("expensive" not in in txt) #gives output True.
```

'not in' checks if expensive is not in txt and return the Boolean value either true or false

```
a = "Hello World"
```

```
print(len(a)) #gives the length of string in a
```

```
a = "Hello, World!"
```

```
print(a[1]) #gives the output e
```

```
a = "Hello, World!"
```

```
print(a[-1]) #gives the output ! from backward and the index begins with -1 from back.
```

```
a = "Hello, World!"
```

```
print(a[1:5])
```

```
#gives the output "ello" beginning at index from 1 to 4.
```

Note: In computer programming indexes begins with 0.

Casting

Use type() to know the datatype.

```
x = '3'
```

```
print(type(x))
```

```
#Gives the output <class 'str'>
```

```
x = int(x)
```

```
print(type(x))
```

```
#Gives the output <class 'int'>
```

Summary of Everything Learnt in Python Class

Numbers

```
x = 1 #int
y = 2.8 #float
z = 1j #complex

print(type(x))

#gives the type integer
print(type(y))

#gives the type float
print(type(z))

#gives the type complex
```

Working with numbers

```
x = 3+5j
y = 5j
z = -5j

print(x+y) #gives the output (3+10j)
print(x*y) #gives the output (-25+15j)
```

User input in python

The input() function allows user to input data.

Syntax: input(prompt)

Prompt between the parenthesis is a string i.e. default message before input.

```
x = input('Enter your name:')

#asks the user to input a name
print('Hello, ' + x)

#gives the output Hello (name) as given by the user
```

Summary of Everything Learnt in Python Class

```
y = int(input('Enter your number:'))  
  
#input() asks input to user in string so int()function  
converts into integer.  
print(y)  
  
#gives the output number given by the user  
print(y**2)  
  
#gives the square of the number given by the user
```

Math in python

Python has built-in module for mathematical tasks. The `math` module contains a set of methods like `acos()`, `asin()`, `tan()`, `fsum()` e.t.c. Syntax to use math module:

```
math.method()
```

To use math module use:

```
import math
```

```
math.cos(1)
```

```
#gives the cosine value of 1 i.e. 0.5403023058681398  
math.acos(0.54)
```

```
#gives the cos inverse value of 0.54 i.e.  
1.0003592173949747
```

```
a = 12.999
```

```
math.floor(a)
```

Random in python

Python has built-in module to make random numbers.

Syntax: `random.method()`

To use random module we must import it first.

Summary of Everything Learnt in Python Class

```
import random
print(random.randint(2,100))
#generates a random number between 2 and 100
```

Formatting

The `format()` method is used to format the specified value and insert them inside the string's place holder.

```
txt = "For only {price:.2f} dollars!"
print(txt.format(price = 49))
#output: For only 49.00 dollars!
```

#named indexes:

```
txt1 = "My name is {fname}, I'm {age}".format(fname = "John", age = 36)
print(txt1)
#output: My name is John, I'm 36
```

#numbered indexes:

```
txt2 = "My name is {0}, I'm {1}".format("John",36)
print(txt2)
#output: My name is John, I'm 36
```

#empty placeholders:

```
txt3 = "My name is {}, I'm {}".format("John",36)
print(txt3)
#output: My name is John, I'm 36
```

#Use "<" to left-align the value:

Summary of Everything Learnt in Python Class

```
txt = "We have {:<8} chickens."
```

```
print(txt.format(49))
```

```
#output: We have 49      chickens.
```

```
#Use "<" to right-align the value:
```

```
txt = "We have {:>8} chickens."
```

```
print(txt.format(49))
```

```
#output: We have      49 chickens.
```

```
#Use "^" to center-align the value:
```

```
txt = "We have {:^8} chickens."
```

```
print(txt.format(49))
```

```
#output: We have    49    chickens.
```

```
# print integer and float value
```

```
print("Number 1 : %2d, Number 2 : %5.2f" % (1, 05.333))
```

```
#output: Number 1 :  1, Number 2 :  5.33
```

```
# print octal value
```

```
print("%o" % (25))
```

```
#output: 31
```

```
# print exponential value
```

```
print("%10.3E" % (356.08977))
```

```
#output: 3.561E+02
```