**Python Documents**

**What is Python?**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.

# **Python 3 Installation & Setup**

**On Window**

* 1. Download the 3.6.0+ version of python from  **<https://www.python.org/downloads/>**
  2. Run the Installer and tick the **Add Python (version) to PATH** checkbox
  3. Click **Customize installation**
  4. Tick all the **checkbox**
  5. Next

**After Installation**

Check python successfully install or not

1. Open command prompt type

# Check the system Python version

$ python --version

**or**

$ python3 --version or python -V

It will show you as a result

New version 3.10.5

1. Open a text editor and create a file with extension file\_name.py

File format :- file\_name.py

1. Write code in file program1.py

print(“hello world”)

1. Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run

C:\Users\Your Name\python program1.py

or

C:\Users\Your Name\python3 program1.py

1. The Output should read

hello world

**What are Keywords in Python?**

Keywords in python are reserved words that have special meaning. They are generally used to define type of variables. Keywords cannot be used for variable or function names. There are following 33 keywords in python-

And, Or, Not, If, Elif, Else, For, While, Break, As, Def, Lambda, Pass, Return, True, False, Try, With, Assert, Class, Continue, Del, Except, Finally, From, Global, Import, In, Is, None, Nonlocal, Raise, Yield

**Name some commonly used built-in modules in Python**

os

sys

math

random

data time

JSON

**What are local variables and global variables in Python?**

**Global Variables:**

Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

**Local Variables:**

Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space.

a=2

def add():

b=3

c= a+b

print(c)

add()

**Python Indentation**

Indentation refers to the spaces at the beginning of a code line.

Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

**Ex**.

if 5 > 2:

print("Five is greater than two!")

**Creating a Comment**

Comments starts with a #, and Python will ignore them:

**Ex.**

1.

#This is a comment

print("Hello, World!")

2.

print("Hello, World!") #This is a comment

**Multiline Comment**

1.

"""

This is a comment

written in

more than just one line

"""

It is also known as doc string.

To access it through function …for example:

1.

def func():

"""Hello

World"""

print(func.\_\_doc\_\_ )

**Variables**

Variables are containers for storing data values.

**Ex.**

1.

a = 5 #numeric

b = "John" #string

c = True #boolean

d = 5.2 #float

print(a, b, c, d)

2.

x, y, z = "Orange", "Banana", "Cherry"

**List**

mylist = ["apple", "banana", "cherry"]

print(mylist )

list1 = ["abc", 34, True, 40, "male"]

print(list1 )

**Dictionary**

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(thisdict )

**Python User Input**

Python allows for user input.

That means we are able to ask the user for input.

The method is a bit different in Python 3.6 than Python 2.7.

Python 3.6 uses the input() method.

Python 2.7 uses the raw\_input() method.

**Ex.**

username = input("Enter username:")

print("Username is: " + username)

**Python If ... Else**

**Ex.1**

a = 33

b = 200

if b > a:

print("b is greater than a")

**Indentation**

Python relies on indentation (whitespace at the beginning of a line) to define scope in the code. Other programming languages often use curly-brackets for this purpose.

**Ex.2**

a = 33

b = 33

if b > a:

print("b is greater than a")

else:

print("a is greater than b")

**Ex.3**

a = 200

b = 33

if b > a:

print("b is greater than a")

elif a == b:

print("a and b are equal")

else:

print("a is greater than b")

**Practice Questions**

1. Write a program to check whether a person is eligible for voting or not. (accept age from user)

2. Write a program to calculate the electricity bill (accept number of unit from user) according to the following criteria :

Unit Price

First 100 units no charge

Next 100 units Rs 5 per unit

After 200 units Rs 10 per unit

**Python List Methods**

* **list.append(x)** # [append](https://tutorial.eyehunts.com/python/python-append-list-to-another-list-example-code/)x to end of list
* **list.extend(iterable)** # append all elements of iterable to list
* **list.insert(i, x)** # [insert](https://tutorial.eyehunts.com/python/python-insert-list-into-list-at-index-example-code/)x at index i
* **list.remove(x)**# [remove](https://tutorial.eyehunts.com/python/python-remove-first-element-from-list-example-code/)first occurance of x from list
* **list.pop([i])** # [pop element](https://tutorial.eyehunts.com/python/python-list-pop-first-element-example-code/) at index i (defaults to end of list)
* **list.clear()** # [delete all elements](https://tutorial.eyehunts.com/python/python-clear-list-remove-all-items-in-python-list/)from the list
* **list.index(x[, start[, end]])** # return index of element x
* **list.count(x)** # return number of occurances of x in list
* **list.reverse()** # reverse elements of list in-place (no return)
* **list.sort(key=None, reverse=False)** # sort list in-place
* **list.copy()** # return a shallow copy of the list

**Python For Loops**

**Ex1.**

fruits = ["apple", "banana", "cherry"]

for x in fruits:

print(x)

**Ex2.**

for x in "banana":

print(x)

Range Function

for x in range(2, 6):

print(x)

for i in range(2,6,2):

“““ Here third argument is for skipping numbers ”””

Print(i)

**Nested Loops**

adj = ["red", "big", "tasty"]

fruits = ["apple", "banana", "cherry"]

for x in adj:

for y in fruits:

print(x, y)

**Practice Questions**

1. Write a program to print first 10 even numbers.

2. Write a program to display all the numbers which are divisible by 11 but not by 2 between 100 and 500.

**Creating a Function**

In Python a function is defined using the def keyword:

def my\_function():

print("Hello from a function")

my\_function()

**Arguments**

def my\_function(fname):

print(fname + " Refsnes")

my\_function("Emil")

my\_function("Tobias")

my\_function("Linus")

**Practice Questions**

1. Pass two value of integer and return sum of them.

2. Pass list to the function and print max and min value of list

**Arrays**

Arrays are used to store multiple values in one single variable:

**Ex.**  cars = ["Ford", "Volvo", "BMW"]

**Access the Elements of an Array**

x = cars[0]

**Modify the value of the first array item:**

cars[0] = "Toyota"

The Length of an Array

x = len(cars)

**Looping Array Elements**

**Ex.**

for x in cars:

print(x)

Adding Array Elements

**Ex.**

cars.append("Honda")

Removing Array Elements

**Ex.**

cars.pop(1) or cars.remove("Volvo")

**Practice Questions**

1. Find the largest and smallest elements of an array..

2. Write a program to check if elements of an array are same or not it read from front or back. E.g.- 2 3 15 15 3 2

3.Write a code to match two list through looping . If number matches append the list in A list and if not append it in B list . (Take the list of same length and they should be ascending order ).

**Python Dictionaries**

**Ex1.**

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

print(thisdict)

Print the "brand" value of the dictionary:

print(thisdict["brand"])

**Length of dictionary**

To determine how many items a dictionary has, use the len() function:

print(len(thisdict))

**Change the value**

You can change the value of a specific item by referring to its key name:

**Ex.**

thisdict["year"] = 2018

**Update**

Update the "year" of the car by using the update() method:

**Ex.**

thisdict.update({"year": 2020})

**Adding Items**

Adding an item to the dictionary is done by using a new index key and assigning a value to it:

**Ex.**

thisdict["color"] = "red"

**Removing Items**

There are several methods to remove items from a dictionary:

**Ex.**

thisdict.pop("model")

print(thisdict)

or

del thisdict["model"]

print(thisdict)

**Python - Loop Dictionaries**

**Loop Through a Dictionary**

You can loop through a dictionary by using a for loop.

When looping through a dictionary, the return value are the keys of the dictionary, but there are methods to return the values as well.

**Ex.**

for x in thisdict:

print(x)

**Ex.**

Print all values in the dictionary, one by one:

for x in thisdict:

print(thisdict[x])

**Ex.**

You can also use the values() method to return values of a dictionary:

for x in thisdict.values():

print(x)

**Ex.**

You can use the keys() method to return the keys of a dictionary:

for x in thisdict.keys():

print(x)

**Ex.**

Loop through both keys and values, by using the items() method:

for x, y in thisdict.items():

print(x, y)

**Nested Dictionaries**

child1 = {

"name" : "Emil",

"year" : 2004

}

child2 = {

"name" : "Tobias",

"year" : 2007

}

child3 = {

"name" : "Linus",

"year" : 2011

}

myfamily = {

"child1" : child1,

"child2" : child2,

"child3" : child3

}

print(myfamily )

**Practice Questions**

1.Write a Python script to add a key to a dictionary.'

Sample Dictionary : {0: 10, 1: 20}

Expected Result : {0: 10, 1: 20, 2: 30}

2.Write a Python script to concatenate following dictionaries to create a new

One.

Sample Dictionary :

dic1={1:10, 2:20}

dic2={3:30, 4:40}

dic3={5:50,6:60}

Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}

**Python Try Except**

The try block lets you test a block of code for errors.

The except block lets you handle the error.

The finally block lets you execute code, regardless of the result of the try- and except blocks.

Exception Handling

When an error occurs, or exception as we call it, Python will normally stop and generate an error message.

These exceptions can be handled using the try statement:

**Ex1.**

try:

print(x)

except:

print("An exception occurred")

**Ex2.**

try:

print("Hello")

except:

print("Something went wrong")

else:

print("Nothing went wrong")

**Ex3.**

try:

print(x)

Except Exception as e:

print("Something went wrong")

print(e)

finally:

print("The 'try except' is finished")

**Practice Questions**

1. Write a program to accept the cost price of a bike and display the road tax to be paid according to the following criteria :

Cost price (in Rs) Tax

> 100000 15 %

> 50000 and <= 100000 10%

<= 50000 5%

1. Write a program to print the following pattern

5 5 5 5 5

4 4 4 4

3 3 3

2 2

1

1. Write a program to print sum, average of all numbers, smallest and largest element of an array. [10, 12, 20, 30, 25, 40, 32, 31, 35, 50, 60]
2. Write a Python program to reverse a string.

Sample String : "1234abcd"

Expected Output : "dcba4321"

1. Take 20 integer inputs from user and print the following:

number of positive numbers

number of negative numbers

number of odd numbers

number of even numbers

number of 0.

1. Take 10 integer inputs from user and store them in an array. Again ask user to give a number. Now, tell user whether that number is present in array or not.
2. Access the value of key ‘history’ from the below dict

sampleDict = {

"class":{

"student":{

"name":"Mike",

"marks":{

"physics":70,

"history":80

}

}

}

}

1. Write a function to find factorial of a number but also store the factorials calculated in a dictionary as done in the Fibonacci series example.
2. Write a code to find palindromes.
3. Write a code to find MAX value from in array using recursion method.
4. Write a code to sort an array using recursion method.
5. Write a code for the pyramid

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

1. Take a list containing only strings. Now, take a string input from user and rearrange the elements of the list according to the number of occurrence of the string taken from user in the elements of the list.

E.g.-LIST : ["no bun","bug bun bug bun bug bug","bunny bug","buggy bug bug buggy"]

STRING TAKEN : "bug"

OUTPUT LIST:["bug bun bug bun bug bug","buggy bug bug buggy","bunny bug","no bun"]

1. Write a program in Python to produce a Star triangle
2. Print multiplication table form 1 to 10

Output:-

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

4 8 12 16 20 24 28 32 36 40

5 10 15 20 25 30 35 40 45 50

6 12 18 24 30 36 42 48 54 60

7 14 21 28 35 42 49 56 63 70

8 16 24 32 40 48 56 64 72 80

9 18 27 36 45 54 63 72 81 90

10 20 30 40 50 60 70 80 90 100

1. Write the code for the pyramid:

\* \* \* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \*

\* \*

1. Write the code to perform inheritance (Multiple,Multilevel) ?
2. Write the code to connect sqlite with python by making sample customer table in a database?
3. Write a code to append a string in a text file starting from a new line?
4. Print the key and value of dictionary through looping (only through one loop)? Hint : Enumerate function.