**Introduction**

Python is an interpreted, object-oriented, high-level programming language. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.

Often, programmers fall in love with Python because of the increased productivity it provides. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

**Applications developed using Python**

YouTube.

DropBox.

Google.

Quora.

Instagram.

**Install python on windows system**

<https://www.howtogeek.com/197947/how-to-install-python-on-windows/>

**Install python on ubuntu system**

<https://linoxide.com/linux-how-to/install-python-ubuntu/>

**Python Tutorials**

<https://www.w3schools.com/python/python_intro.asp>

**Creating Variables**

Unlike other programming languages, Python has no command for declaring a variable.

A variable is created the moment you first assign a value to it.

Example

x = 5

print(x)

Output: 5

y = "John"

print(y)

Output:”John”

# **Python Strings**

Strings in python are surrounded by either single quotation marks, or double quotation marks.

Strings can be output to screen using the print function.

For example: print("hello").

Output:”hello”

Square brackets can be used to access elements of the string.

Example: Get the character at position 1 (remember that the first character has the position 0):

a = "Hello, World!"

print(a[1])

Output:”e”

**Substring(Slicing string)** - Get the characters from position 2 to position 5 (not included):

Example

b = "Hello, World!"

print(b[2:5])

Output:”llo”

The **len()** method returns the length of a string:

Example

a = "Hello, World!"

print(len(a))

Output:13

The **split()** method splits the string into substrings if it finds instances of the separator:

Example

a = "Hello, World!"

print(a.split(","))

Output:['Hello', ' World!']

The **strip()** method removes any whitespace from the beginning or the end:

Example

a = " Hello, World! "

print(a.strip())  
Output:"Hello, World!"

The **replace()** method replaces a string with another string:

Example

a = "Hello, World!"

print(a.replace("H", "J"))

Output:”Jello, World!”

Example:

a = "Hello, World!”

a.replace(",","")

Output:”Hello World!”

**Data Types in Python**

There are 4 types of data types in python namely Lists, Dictionaries, Tuples & Sets.

1. **Lists -** List is a collection which is ordered and changeable. Allows duplicate members.
2. **Dictionaries -** Dictionary is a collection which is unordered, changeable and indexed. No duplicate members.
3. **Tuples -** Tuple is a collection which is ordered and unchangeable. Allows duplicate members.
4. **Sets -** Set is a collection which is unordered and unindexed. No duplicate members.

## **List**

A list is a collection which is ordered and changeable. In Python lists are written with square brackets. List is a mutable data type. I.e. value of the list can be altered or modified.

Example  
thislist = ["apple", "banana", "cherry"]  
print(thislist)

Output:['apple', 'banana', 'cherry']

**Access Items of the list**

You access the list items by referring to the index number:

Example

Print the second item of the list:

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

print(thislist[1])

Output:banana

**Change Item Value**

To change the value of a specific item, refer to the index number:

Example

Change the second item:

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

thislist[1] = "blackcurrant"

print(thislist)

Output:['apple', 'blackcurrant', 'cherry']

**Loop Through a List**

You can loop through the list items by using a for loop:

Example

Print all items in the list, one by one:

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

for x in thislist:

print(x)

Output:

apple

banana

cherry

**Check if Item Exists**

To determine if a specified item is present in a list use the **in** keyword:

Example

Check if "apple" is present in the list:

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

if "apple" in thislist:

print("Yes, 'apple' is in the fruits list")

Output:Yes, 'apple' is in the fruits list

**List Length**

To determine how many items a list has, use the **len()** method:

Example

Print the number of items in the list:

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

print(len(thislist))

Output:3

**Add Item in the list**

To add an item to the end of the list, use the **append()** method:

Example

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

thislist.append("orange")

print(thislist)

Output:['apple', 'banana', 'cherry', 'orange']

**Add item at specific index in the list**

To add item in list at specific index or position **insert()** method is used

Example

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

thislist.insert(1, “pinaple")

print(thislist)

Output:['apple', ‘pinaple’, 'banana', 'cherry', 'orange']

**Remove Item from the list**

The **del** keyword removes the specified index:

Example

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

del thislist[0]

print(thislist)

Output:['banana', 'cherry']

**Remove all items from the list**

The **del list[:]** empties the list

Example

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

del thislist [:]

print(thislist)

Output:[]

**Delete the entire list**

The **del** keyword can also delete the list completely

Example

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

del thislist

Output:<type 'list'>