Python is High level general programming language.

Python is open source language.

Platform independent

1. **what can Python do?**

* Python can be used on a server to create web applications.
* Python can be user alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be user for rapid prototyping, or for production-ready software development.
* Python can be use for rapid prototyping or for production ready software development

1. **Python Indentation**

* Indentation refers to the spaces at the beginning of a code line.
* Python uses indentation to indicate a block of code
* Programming languages the indentation in code is for readability only, the indentation in python is very important.

1. **Variable in python?**

* A variable is nothing but a reserved memory to store values.
* Variables are used to store the data.
* Memory allocated when the values are stored in variables.
* Every variable must have some type
* Python has no command for declaring a variable.

**Data types**

Int X = 100

String x = “Welcome”

**Text** Type: str

**Numeric Types**: int, float

**Sequence Type**: list, tuple

**Mapping Type**: dict

**Set Types:** set

**Boolean Type**: bool

**Variable Names**

* Variable name must start with a letter or the underscore character
* A variable name cannot start with a number
* A variable name can only contain alpha-numeric characters and underscores
* Variable name are case sensitive
* A variable name cannot be any of the Python Keywords.

**Multi Words Variable Names**

* Variable names with more than one word can be difficult to read.
* There are several techniques you can use to make more readable

**Camel Case:**  Each word, except the first, starts with a capital letter

**Pascal Case:** Each word starts with a capital letter.

**Snake Case:** Each word is separated by an underscore character

Many Values to multiple variables: Python allows you to multiple variable in one line

**Python output variable:**

The python print () function is often used to output variables

**Global Variables:**

Variable that are created outside of a function Global variable can be used by everyone,both inside of functions and outside

DAY 2 : Python Programming

-----------------------------------------------------------

Delete the variables

Operators --> A symbol which will perform an operation between 2 or more variables

a=100

B=200

Print(a+b)

1. Arithmetic
2. Relational
3. Logical

DAY 3 Python Programming:

Control statement

1. Conditional statements- if if..else elif
2. Looping statements- while, for
3. Jumping statements- break, continue

Print(1)

Print(2)

Range() function in python:

Range(10) which is shows 0------to 10

Range(1,10) which is shows start with 1………….. to 9

Looping statements

* While loop
* For loop

DAY 4 PYTHON PROGRAMMING

Collections

* List
* Tuple
* Set
* Dictionary

List

A list is a collection which is order and changeable in python lists are written with square brackets[] list is mutable.

Tuple:

A tuple is a collection which is ordered and unchangeable in python tuples are written with round brackets tuple is immutable

DAY 5-PYTHON PROGRAMMING

1 List

2 Tuple

3 Set

4 Dictionary

Set:

A set is a collection which is unordered and UN-indexed. In Python sets are written { }

* Set is mutable

Dictionary:

A dictionary is a collection which is unordered,changeable(mutable) and indexed In python dictionaries are written with curly brackets and they have keys and values{}

Key Value

Product1: 100

Product2: 200

Product3: 500

DAY6 PYTHON PROGRAMMING

Functions

Functions means set of statements which will perform a task.

1. Function declaration/creation
2. Calling the function/invoking function

Def---> create function

Function name() -------> function call

1. Function does not take arguments not returns any value(None)
2. Function does not take arguments but returns some value
3. Function takes arguments but no return value.
4. Function takes arguments and also return value

Global & Local variables

* The variables create outside of function are called as global variable
* The variables create inside of the function called as local variable.

2 types of arguments/parameters we can pass to the function

1. Positional arguments
2. Keyword arguments

DAY 7 PYTHON PROGRAMMING

Java

Python- structured + object oriented

OOPS(Object oriented Programming concepts)

Class ,

Object

Inheritance

Overloading

Polymorphism

Class Object

Employee ------------------> John,Scott,Marry etc………….

Animal ---------------------> Dog, Horse,Elephant etc…………………

Class -----> collection of variables(attributes) & Methods(Behavior)

A class is blue print

Logical entity

Does not occupy space in the memory

Object -----> object is an instance of class

Physical entity

Occupy certain amount space in the memory

For one class we can create multiple objects.

Objects are independent…..

Function vs Method

Function--> we can create without class

Method--> creates in side the class

2 types of methods we can define within the class…

1. Instance Method(We can call only through object)
2. Static Method(We can directly call using class)

Annotation @static-method

Global

Local

Class variable

Method & Constructor

Method: Give any name

Return the value/s

Method can take argument/Parameters

We have to use an object to invoke the method

Constructor:\_constructor name is fixed

Def \_init\_(self):

Constructor will not return any value

Constructor can also take arguments or parameters

Constructor will be call at the time of at the object creation itself.

Variable name

* A variable name must start with latter or the underscore character
* A variable name cannot start with a number
* A variable name can only contain alpha-numeric characters and underscore
* Variable name are case sensitive
* A variable name cannot be any of the Python keyword.

Multi WORDS VARIABLE NAMES

Variable names with more than one word ca be difficult to read

There are several techniques you can use to make them more readable:

CAMEL CASE

Each word, except the first,starts with a capital letter

Pascal case

Each word start with a capital letter:

Snake Case

Each word is separated by an underscore character:

Many Values to Multiple Variables

Python allows you to assign values to multiple variables in one line:

Output Variables

The python print() function is often used to output variable

Global Variables

Variables that are created out side of a function (as in all of the examples above) are known as global variables

Global variables can be sued by everyone both inside and function and out side.

If you create a variable with the same name inside a function,this variable will be local,and can oly be used inside the function,the global

variable with the same name will remain as it was global and with the

original value

DAY 8 PYTHON PROGRAMMING

Class

Object

Method

Constructor

Inheritance

Acquiring all the attributes(variables) &behavior(methods)from one class to another class is called as inheritance

Class A ----->a,b,c m1() m2()-------> A is parent of B class (Base class/Super class)

Class B(A)-------> x,y,z m3()------> B is child of a class (Sub class/Derived class)

Objectives of Inheritance

* Code re-usability
* Avoid duplication

Types of Inheritance

1. Single
2. Multi level
3. Hierarchy
4. Multiple

Built-In Data Types

|  |  |
| --- | --- |
| Text Type: | str |
| Numeric Types: | int, float, complex |
| Sequence Types: | list, tuple, range |
| Mapping Type: | dict |
| Set Types: | set, frozenset |
| Boolean Type: | bool |
| Binary Types: | bytes, bytearray, memoryview |
| None Type: | NoneType |

Python numbers:

* Int
* Float
* Complex

Variables of numeric types are created when you assign a value to them

To verify the type of any object in python use the type() function

Specify a Variable Type

1. Int() = Constructs an integer number form an integer literal,a float literal(by removing all decimals), or a string literal
2. Float() = Constructs a float number from an integer literal a float literal or string literal (providing the string represents a float or an integer)
3. Str() = Constructs a string from a wide variety of data types,including strings,integer literals and float literals

Strings

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

‘Hello’ in the same as “Hello”.

Display a string literal with the print() function:

Strings are Arrays:

String in python are arrays of the bytes representing Unicode characters

String get character at position 1 (the first character has the position 0):

Looping Through a String

String are arrays we can loop through the characters in a string with a for loop.

Check String:

Check the certain phrase or character in present in a string we can use the keyword “in”.

Check if Not

Check if a certain phrase or character is not present in string we can use the keyword not in.

Python-Slicing Strings:

You can return a range of characters by using the slice syntax specify the start index and the end index separated by a colon to return a part of string

Negative Indexing

Use negative indexes to start the slice from the end of the string

Python Modify Strings

Upper case : The upper() method return the string in upper case

Lower case : The Lower() method return the string the lower case

Remove white space: The strip() method removes any white space form the beginning or the end

Replace string: replace() method a string with another string

Split string: split() method returns a list where the text between the specified separator become the list items

Python-String Concatenation

String concatenation: to concatenate or combine two strings you can use the + operator:

String Format: as we learned we can not combine string and number

We can combine strings and numbers by using the format() method

Format() method take the passed arguments,formats them, and places them in the string where the placeholders{} are:

Escape character

An escape character is a backslash \ followed by the character you want to insert

|  |  |  |
| --- | --- | --- |
| **Code** | **Result** | **Try it** |
| \' | Single Quote | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_escape2" \t "https://www.w3schools.com/python/_blank) |
| \\ | Backslash | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_backslash" \t "https://www.w3schools.com/python/_blank) |
| \n | New Line | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_newline" \t "https://www.w3schools.com/python/_blank) |
| \r | Carriage Return | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_r" \t "https://www.w3schools.com/python/_blank) |
| \t | Tab | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_t" \t "https://www.w3schools.com/python/_blank) |
| \b | Backspace | [Try it »](https://www.w3schools.com/python/showpython.asp?filename=demo_string_b" \t "https://www.w3schools.com/python/_blank) |
| \f | Form Feed |  |
| \ooo | Octal value | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_octal" \t "https://www.w3schools.com/python/_blank) |
| \xhh | Hex value |  |

Boolean Values

* Boolean represent one of two values: True or False
* You can evaluate any expression in python and get one of two answer True or False.
* When you compare two values, the expression is evaluated and python returns the boolean answer

PYTHON OPERATORS

Operators are used to perform operation on variables and values we use the + operator to add together two values

## Python Arithmetic Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Example** | **Try it** |
| + | Addition | x + y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_add" \t "https://www.w3schools.com/python/_blank) |
| - | Subtraction | x - y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_sub" \t "https://www.w3schools.com/python/_blank) |
| \* | Multiplication | x \* y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_mult" \t "https://www.w3schools.com/python/_blank) |
| / | Division | x / y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_div" \t "https://www.w3schools.com/python/_blank) |
| % | Modulus | x % y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_mod" \t "https://www.w3schools.com/python/_blank) |
| \*\* | Exponentiation | x \*\* y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_exp" \t "https://www.w3schools.com/python/_blank) |
| // | Floor division | x // y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_floordiv" \t "https://www.w3schools.com/python/_blank) |

Python Assignment operators

Assignment operators are used to assign values to variables:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Example** | **Same As** | **Try it** |
| = | x = 5 | x = 5 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass1" \t "https://www.w3schools.com/python/_blank) |
| += | x += 3 | x = x + 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass2" \t "https://www.w3schools.com/python/_blank) |
| -= | x -= 3 | x = x - 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass3" \t "https://www.w3schools.com/python/_blank) |
| \*= | x \*= 3 | x = x \* 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass4" \t "https://www.w3schools.com/python/_blank) |
| /= | x /= 3 | x = x / 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass5" \t "https://www.w3schools.com/python/_blank) |
| %= | x %= 3 | x = x % 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass6" \t "https://www.w3schools.com/python/_blank) |
| //= | x //= 3 | x = x // 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass7" \t "https://www.w3schools.com/python/_blank) |
| \*\*= | x \*\*= 3 | x = x \*\* 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass8" \t "https://www.w3schools.com/python/_blank) |
| &= | x &= 3 | x = x & 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass9" \t "https://www.w3schools.com/python/_blank) |
| |= | x |= 3 | x = x | 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass10" \t "https://www.w3schools.com/python/_blank) |
| ^= | x ^= 3 | x = x ^ 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass11" \t "https://www.w3schools.com/python/_blank) |
| >>= | x >>= 3 | x = x >> 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass12" \t "https://www.w3schools.com/python/_blank) |
| <<= | x <<= 3 | x = x << 3 |  |

## Python Comparison Operators

Comparison operators are used to compare two values:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Example** | **Try it** |
| == | Equal | x == y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare1" \t "https://www.w3schools.com/python/_blank) |
| != | Not equal | x != y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare2" \t "https://www.w3schools.com/python/_blank) |
| > | Greater than | x > y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare4" \t "https://www.w3schools.com/python/_blank) |
| < | Less than | x < y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare5" \t "https://www.w3schools.com/python/_blank) |
| >= | Greater than or equal to | x >= y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare6" \t "https://www.w3schools.com/python/_blank) |
| <= | Less than or equal to | x <= y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare7" \t "https://www.w3schools.com/python/_blank) |

## Python Logical Operators

Logical operators are used to combine conditional statements:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Description** | **Example** | **Try it** |
| and | Returns True if both statements are true | x < 5 and  x < 10 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_logical1" \t "https://www.w3schools.com/python/_blank) |
| or | Returns True if one of the statements is true | x < 5 or x < 4 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_logical2" \t "https://www.w3schools.com/python/_blank) |
| not | Reverse the result, returns False if the result is true | not(x < 5 and x < 10) | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_logical3" \t "https://www.w3schools.com/python/_blank) |

## Python Identity Operators

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location:

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Description** | **Example** | **Try it** |
| is | Returns True if both variables are the same object | x is y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_identity1" \t "https://www.w3schools.com/python/_blank) |
| is not | Returns True if both variables are not the same object | x is not y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_identity2" \t "https://www.w3schools.com/python/_blank) |

## Python Bitwise Operators

Bitwise operators are used to compare (binary) numbers:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operator** | **Name** | **Description** | **Example** | **Try it** |
| & | AND | Sets each bit to 1 if both bits are 1 | x & y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_and" \t "https://www.w3schools.com/python/_blank) |
| | | OR | Sets each bit to 1 if one of two bits is 1 | x | y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_or" \t "https://www.w3schools.com/python/_blank) |
| ^ | XOR | Sets each bit to 1 if only one of two bits is 1 | x ^ y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_xor" \t "https://www.w3schools.com/python/_blank) |
| ~ | NOT | Inverts all the bits | ~x | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_not" \t "https://www.w3schools.com/python/_blank) |
| << | Zero fill left shift | Shift left by pushing zeros in from the right and let the leftmost bits fall off | x << 2 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_left_shift" \t "https://www.w3schools.com/python/_blank) |
| >> | Signed right shift | Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off | x >> 2 |  |

## Operator Precedence

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Try it** |
| () | Parentheses | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_parentheses" \t "https://www.w3schools.com/python/_blank) |
| \*\* | Exponentiation | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_exponent" \t "https://www.w3schools.com/python/_blank) |
| +x  -x  ~x | Unary plus, unary minus, and bitwise NOT | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_bitwise_not" \t "https://www.w3schools.com/python/_blank) |
| \*  /  //  % | Multiplication, division, floor division, and modulus | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_multiplication" \t "https://www.w3schools.com/python/_blank) |
| +  - | Addition and subtraction | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_subtraction" \t "https://www.w3schools.com/python/_blank) |
| <<  >> | Bitwise left and right shifts | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_shift" \t "https://www.w3schools.com/python/_blank) |
| & | Bitwise AND | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_bitwise_and" \t "https://www.w3schools.com/python/_blank) |
| ^ | Bitwise XOR | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_bitwise_xor" \t "https://www.w3schools.com/python/_blank) |
| | | Bitwise OR | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_bitwise_or" \t "https://www.w3schools.com/python/_blank) |
| ==  !=  >  >=  <  <=  is  is not  in  not in | Comparisons, identity, and membership operators | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_like" \t "https://www.w3schools.com/python/_blank) |
| not | Logical NOT | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_not" \t "https://www.w3schools.com/python/_blank) |
| and | AND | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_and" \t "https://www.w3schools.com/python/_blank) |
| or | OR |  |

DAY 9 MODULES AND PACKAGES

Module---> collection of functions classes(Variables+methods)

* How to create a module
* How to call the functions from one module to another module…
* Same functions in 2 modules

Import module Name

Form module name import functions,classes

Packages:

Collections of modules.

Package---->modules----->functions&classes

#pack1===>module 1==>display()

====>module 2==>show()

Python Lists

Lists are used to store multiple items in a single variable

Lists are one of 4 built in data types in python used to store collections of data the other 3 are Tuple,set and Dictionary all with different qualities and usage

List Items

List items are ordered,changeable and allow duplicate values.

List items are indexed,the first item has index and second item has index them

Ordered

There are some list methods that will change the order

Changeable:

The list is changeable,meaning that we can change,add and remove item in a list after it has been created

List length: Determine how many items a list has use the len() function:

List()Constructor:

It is also possible to use the list() constructor when creating a new list

## Python Collections (Arrays)

There are four collection data types in the Python programming language:

* ****List**** is a collection which is ordered and changeable. Allows duplicate members.
* **[Tuple](https://www.w3schools.com/python/python_tuples.asp)** is a collection which is ordered and unchangeable. Allows duplicate members.
* **[Set](https://www.w3schools.com/python/python_sets.asp)** is a collection which is unordered, unchangeable\*, and unindexed. No duplicate members.
* **[Dictionary](https://www.w3schools.com/python/python_dictionaries.asp)** is a collection which is ordered\*\* and changeable. No duplicate members.

Access Items

List items are indexed and access them referring to the index number

Negative Indexing: negative indexing means start from the end

Check the item is exists:is present in list use the in keyword

Append items

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

Insert Items

To Insetrt() method inserts an item at the specified index:

Loop through the index number:

You can also loop through the list items by referring to their index number use the range() and len() functions to create a suitable iterateble

List Comprehension

List comprehension is a shorter syntax and create new list based on the value of the existing list.Use for statement with a conditional test inside

Tuple:

Tuples are store for multiple items in a single variable

A tuple is a collection which is ordered and unchangeable

Ordered:

Tuples are ordered it means that the items have a define order and order is not change

Unchangeable:

Tuples are unchangeable meaning that we cannot change,add,remove items

Using a while loop

Tuple items by using a while loop use the len() function to determine the length of the tuple

Sets:

* Sets method store mulitpal single variable
* Set items are unordered,unchangeable,and do not allow duplicate value
* Unordered mean every time you use them and cannot be referred to by index or key
* Unchangeable meaning that we cannot change the items the set has been created
* Set cannot have two items with the same value.
* Set method written with curly{} brackets,

Access items

Access item in a set by referring to an index or a key set items using a for loop

Join two Sets

Use the union () method that returns an new set containing all item

Python Dictionaries:

Dictionary are used to store data values in key:value pairs

A dictionary is a collection which is ordered\* changeable and do not allow duplicates

Accessing items: Access the items of a dictionary by referring its key name ,inside square brace

Python If…else

* Equals: a == b
* Not Equals: a != b
* Less than: a < b
* Less than or equal to: a <= b
* Greater than: a > b
* Greater than or equal to: a >= b
* “If statement” is written by using the if keyword

Python while loops:

Python has 2 primitive loop commands:

While loops

For loops

* Break() statement : we can stop the loop even if the while condition is true we use break statement
* Continue() statement: we can stop the current iteration, and continue with the continue statement
* Else statement(): we run the block the code once the condition is no longer true.their we use the else statement

Python for loops:

For loops used for iterating over a sequences

For keyword used for loop so we can execute the statement

* Range Function(): the loop through a set of code and number of times we use range function

Python Functions:

A function is a block of code it pass data which know as parameters into a function a function can return data as a result

* Python a function defined using the def keyword
* Arbitrary Arguments, \*args add \* befor the parameter name in the function

PYTHON LAMBDA:

* A lambda function is a small anonymous function it can take any number of arguments that only have one expression
* Lambda arguments: expression
* Lambda is an anonymous function inside another function

PYTHON ARRAYS:

Arrays are used to store multiple values in one single variable

DAY 10 EXCEPTION Handling

Exception is an event which will cause program termination

Python Inheritance:

Parent class: Is the class being inherited from also called base class

Child class: Is the class that inherits form another class and it also called derived class

Use the pass keyword when you do not want to add any another properties or methods to the class.

Add the \_\_init\_\_()Function:

So far we have created a child class that inherits the properties an methods form its parents we want to ass the init() function to the child class

Init function is called automatically every time the class is being used to create a new object

The child’s() \_init\_() function overrides the inheritance of the patent’s \_init\_() function

Super() function that will make the child class inherit all the methods and properties from its parent

Python Iterators:

1. Iterator is an object that contains a countable number of values
2. Iterator is an object that can be iterated upon meaning that you can traverse through all the values.
3. Consist the methods \_inter\_() and \_next\_()

Lists,tuples,dictionaries and sets all the iterable objects all these objects have a iter() method

Looping Through an iterator:

Using “For” loop to iterate through an iterable

The \_iter\_() method acts similar it always return the iterator object itself

The \_next\_() method allows to do operation and it return the item in the sequence

Python Polymorphism:

The word “polymorphism” means “many forms” and it refers to methods/functions/operators with the same name that can be executed on many objects or classes

Len() strings returns the numbers of characters

xpath

1. X path is defined as XML path.
2. It is a syntax or language for finding any element on the web page using XML path expression
3. X path is used to find the location of any element on a web page using HTML DOM structure.
4. X path can be used to navigate through element and attributes in Dom.
5. X path is an address of the element

Two type of x paths.

1. Relative x path starts from root html node
2. Absolute x path directly jump to element on dom.

How to capture automatically

* Firebug, firepath ---> Firefox -->deprecated/not available
* Right click on element--->Inspect-->Highlight HTML code -->Right click -->Copy x path
* Selector Hub

2 reason to prefer relative x path

1. If developer introduced new element than absolute x path will be broken.
2. If devloper changed the location then absolute x path will be broken
3. Which x path is pref-red?why?

--> Relative

X path options:

And

Or

Contains()

Starts-with()

Text()

X path axes

Self

Parent

Child

Ancestor

Descendant

Following

Following-sibling

Preceding

Preceding-sibling

Day 15:

1. Get commands
2. Conditional commands
3. Browser commands
4. Navigational commands
5. Wait commands

Application commands:

1. Title:To capture the title of the current webpage
2. Get():Opening the application URL
3. Current\_URL:To capture the current url of the web page.
4. Page\_source:To capture source code of the Page

Conditional commands

is\_displayed()

Is\_enabled()

Is \_selected()

Browser commands

Close(): close single browser window

Quit():close multiple browser window

Navigational commands

Back()

Forward()

Refresh()

Diff between find element vs find elements

Find element found the single element in dom

Find elements use multiple element in html dom

Day 16

Wait commands:

1. .Time .sleep(time):

Adv

1. Simple to use

DisAdv

* Performance of the script is very poor
* If the element is not available within the time mentioned,still there is a chance of getting exception

1. Implicit wait

Adv

1. Single statement
2. Performance will not be reduced.(if the element is available withing the time it proceed to execute further statement.

DisAdvantage

1. If the element is not available within the time mentioned,still there is a chance of getting exception.
2. Explicit wait

Links

1. Internal
2. External
3. Broken link

Day 23:

Boostrap Dropdown

How to capture screen shot of a webpage

How to handle cookies

DAY 24

Data Driven testing

Openpyxl= we can work with excel files(.xlsx)

1. Read data from excel
2. How to write data into excel
3. Data driven test case