|  |  |
| --- | --- |
| Numbers Coloring Book |  |

### **Q1. What is the difference between list and tuples in Python?**

|  |  |
| --- | --- |
| **LIST vs TUPLES** | |
| **LIST** | **TUPLES** |
| Lists are mutable i.e they can be edited. | Tuples are immutable (tuples are lists which can’t be edited). |
| Lists are slower than tuples. | Tuples are faster than list. |
| Syntax: list\_1 = [10, ‘Chelsea’, 20] | Syntax: tup\_1 = (10, ‘Chelsea’ , 20) |
| Lists consume a lot of memory | Tuples consume less memory when compared to lists |

### **Q2.What is pep 8?**

***Ans:***PEP stands for **Python Enhancement Proposal.**It is a set of rules that specify how to format Python code for maximum readability.

**Q3. How is Memory managed in Python?**

### ***Ans:*** Memory in Python is managed by Python private heap space. All Python objects and data structures are located in a private heap. This private heap is taken care of by Python Interpreter itself, and a programmer doesn’t have access to this private heap. Python memory manager takes care of the allocation of Python private heap space. Memory for Python private heap space is made available by Python’s in-built garbage collector, which recycles and frees up all the unused memory.

**Q3.** What are Python Modules?

Files containing Python codes are referred to as Python Modules. This code can either be classes, functions, or variables and saves the programmer time by providing the predefined functionalities when needed. It is a file with “.py” extension containing an executable code.

### **Q4. Explain Inheritance in Python with an example?**

As Python follows an **object-oriented**programming paradigm, classes in Python have the ability to inherit the properties of another class. This process is known as inheritance. Inheritance provides the **code reusability feature**. The class that is being inherited is called a **superclass** or the parent class, and the class that inherits the superclass is called a **derived** or child class. The following types of inheritance are supported in Python:

* **Single inheritance**: When a class inherits only one superclass
* **Multiple inheritance**: When a class inherits multiple superclasses
* **Multilevel inheritance**: When a class inherits a superclass, and then another class inherits this derived class forming a ‘parent, child, and grandchild’ class structure
* **Hierarchical inheritance**: When one superclass is inherited by multiple derived classes

### **What is a dictionary in Python?** Python dictionary is one of the supported [data types in Python](https://intellipaat.com/blog/tutorial/python-tutorial/python-datatypes/). It is an unordered collection of elements. The elements in dictionaries are stored as key-value pairs. Dictionaries are indexed by keys.

dict={‘Country’:’India’,’Capital’:’New Delhi’, }

**What is \_\_init\_\_ in Python?**

\_\_init\_\_ is a contructor method in Python and is automatically called to allocate memory when a new object/instance is created. All classes have a **\_\_init\_\_** method associated with them. It helps in distinguishing methods and attributes of a class from local variables.

# class definition

**class** **Student**:

**def** **\_\_init\_\_**(self, fname, lname, age, section):

self.firstname = fname

self.lastname = lname

self.age = age

self.section = section

# creating a new object

stu1 = Student("Sara", "Ansh", 22, "A2")

### **What are Python packages?**

A Python package refers to the collection of different sub-packages and modules based on the similarities of the function.

### **What are decorators in Python?**

In Python, decorators are necessary functions that help add functionality to an existing function without changing the structure of the function at all. These are represented by **@decorator\_name** in Python and are called in a bottom-up format.

### **Explain split(), sub(), subn() methods of “re” module in Python?**

These methods belong to the [Python RegEx or ‘re’ module](https://intellipaat.com/blog/tutorial/python-tutorial/python-regex-regular-expressions/) and are used to modify strings.

split(): This method is used to split a given string into a list.

sub(): This method is used to find a substring where a regex pattern matches, and then it replaces the matched substring with a different string.

subn(): This method is similar to the sub() method, but it returns the new string, along with the number of replacements.

**Do we need to declare variables with data types in Python?**

No. Python is a dynamically typed language, I.E., Python Interpreter automatically identifies the data type of a variable based on the type of value assigned to the variable.

### **What is the difference between / and // operator in Python?**

* **/**: is a division operator and returns the Quotient value.

10/3

3.33

* **//**: is known as floor division operator and used to return only the value of quotient before decimal

10//3

3

**What do file-related modules in Python do? Can you name some file-related modules in Python?**

Python comes with some file-related modules that have functions to manipulate text files and binary files in a file system. These modules can be used to create text or binary files, update their content, copy, delete, and more.

Some file-related modules are os, os.path, and shutil.os. The os.path module has functions to access the file system, while the shutil.os module can be used to copy or delete files.

**What does len() do?**

len() is an inbuilt function used to calculate the length of sequences like list, [python string](https://intellipaat.com/blog/tutorial/python-tutorial/python-strings/), and array.

my \_list=[1,2,3,4,5]

len(my\_list)

### **How will you remove duplicate elements from a list?**

To remove duplicate elements from the list we use the set() function.

Consider the below example:

demo\_list=[5,4,4,6,8,12,12,1,5]

unique\_list = list(set(demo\_list))

output:[1,5,6,8,12]

### **How can files be deleted in Python?**

You need to import the OS Module and use os.remove() function for deleting a file in python.  
consider the code below:

import os

os.remove("file\_name.txt")

### **What is slicing in Python?**

* As the name suggests, ‘slicing’ is taking parts of.
* Syntax for slicing is **[start : stop : step]**
* **start** is the starting index from where to slice a list or tuple
* **stop** is the ending index or where to sop.
* **step** is the number of steps to jump.
* Default value for **start** is 0, **stop** is number of items, **step** is 1.
* Slicing can be done on **strings, arrays, lists**, and **tuples**.

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print(numbers[1 : : 2]) #output : [2, 4, 6, 8, 10]

### What is an Interpreted language?

An Interpreted language executes its statements line by line. Languages such as Python, Javascript, R, PHP, and Ruby are prime examples of Interpreted languages. Programs written in an interpreted language runs directly from the source code, with no intermediary compilation step.

### What is pass in Python?

The pass keyword represents a null operation in Python. It is generally used for the purpose of filling up empty blocks of code which may execute during runtime but has yet to be written. Without the **pass** statement in the following code, we may run into some errors during code execution.

**def** **myEmptyFunc**():

# do nothing

**pass**

myEmptyFunc() # nothing happens

## Without the pass keyword

# File "<stdin>", line 3

# IndentationError: expected an indented block

### What is docstring in Python?

Documentation string or docstring is a multiline string used to document a specific code segment.

The docstring should describe what the function or method does.

**What is the difference between Python Arrays and lists?**

* Arrays in python can only contain elements of same data types i.e., data type of array should be homogeneous. It is a thin wrapper around C language arrays and consumes far less memory than lists.
* Lists in python can contain elements of different data types i.e., data type of lists can be heterogeneous. It has the disadvantage of consuming large memory

### What are decorators in Python?

**Decorators** in Python are essentially functions that add functionality to an existing function in Python without changing the structure of the function itself. They are represented the @decorator\_name in Python and are called in a bottom-up fashion. For example:

### What are Dict and List comprehensions?

Python comprehensions, like decorators, are **syntactic sugar** constructs that help **build altered** and **filtered lists**, dictionaries, or sets from a given list, dictionary, or set. Using comprehensions saves a lot of time and code that might be considerably more verbose (containing more lines of code). Let's check out some examples, where comprehensions can be truly beneficial:

* **Performing mathematical operations on the entire list**

my\_list = [2, 3, 5, 7, 11]

squared\_list = [x\*\*2 **for** x **in** my\_list] # list comprehension

# output => [4 , 9 , 25 , 49 , 121]

squared\_dict = {x:x\*\*2 **for** x **in** my\_list} # dict comprehension

# output => {11: 121, 2: 4 , 3: 9 , 5: 25 , 7: 49}

* **Performing conditional filtering operations on the entire list**

my\_list = [2, 3, 5, 7, 11]

squared\_list = [x\*\*2 **for** x **in** my\_list **if** x%2 != 0] # list comprehension

# output => [9 , 25 , 49 , 121]

squared\_dict = {x:x\*\*2 **for** x **in** my\_list **if** x%2 != 0} # dict comprehension

# output => {11: 121, 3: 9 , 5: 25 , 7: 49}

### What is lambda in Python? Why is it used?

Lambda is an anonymous function in Python, that can accept any number of arguments, but can only have a single expression. It is generally used in situations requiring an anonymous function for a short time period. Lambda functions can be used in either of the two ways:

* Assigning lambda functions to a variable:

mul = **lambda** a, b : a \* b

print(mul(2, 5)) # output => 10

* Wrapping lambda functions inside another function:

**def** **myWrapper**(n):

**return** **lambda** a : a \* n

mulFive = myWrapper(5)

print(mulFive(2)) # output => 10

### How do you copy an object in Python?

In Python, the assignment statement (= operator) does not copy objects. Instead, it creates a binding between the existing object and the target variable name. To create copies of an object in Python, we need to use the **copy** module. Moreover, there are two ways of creating copies for the given object using the **copy** module -

**Shallow Copy** is a bit-wise copy of an object. The copied object created has an exact copy of the values in the original object. If either of the values is a reference to other objects, just the reference addresses for the same are copied.  
**Deep Copy** copies all values recursively from source to target object, i.e. it even duplicates the objects referenced by the source object.

**from** copy **import** copy, deepcopy

list\_1 = [1, 2, [3, 5], 4]

## shallow copy

list\_2 = copy(list\_1)

list\_2[3] = 7

list\_2[2].append(6)

list\_2 # output => [1, 2, [3, 5, 6], 7]

list\_1 # output => [1, 2, [3, 5, 6], 4]

## deep copy

list\_3 = deepcopy(list\_1)

list\_3[3] = 8

list\_3[2].append(7)

list\_3 # output => [1, 2, [3, 5, 6, 7], 8]

list\_1 # output => [1, 2, [3, 5, 6], 4]

### What is pickling and unpickling?

Python library offers a feature - **serialization** out of the box. Serializing an object refers to transforming it into a format that can be stored, so as to be able to deserialize it, later on, to obtain the original object. Here, the **pickle** module comes into play.

**Pickling:**

* Pickling is the name of the serialization process in Python. Any object in Python can be serialized into a byte stream and dumped as a file in the memory. The process of pickling is compact, but pickle objects can be compressed further. Moreover, pickle keeps track of the objects it has serialized, and the serialization is portable across versions.
* The function used for the above process is pickle.dump().

**Unpickling:**

* Unpickling is the complete inverse of pickling. It deserializes the byte stream to recreate the objects stored in the file and loads the object to memory.
* The function used for the above process is pickle.load()

**What is the difference between .py and .pyc files?**

.py files contain the source code of a program. Whereas .pyc file contains the bytecode of your program. We get bytecode after compilation of .py file (source code). .pyc files are not created for all the files that you run. It is only created for the files that you import.

### **Explain split() and join() functions in Python?**

* You can use **split()** function to split a string based on a delimiter to a list of strings.
* You can use **join()** function to join a list of strings based on a delimiter to give a single string.

string = "This is a string."

string\_list = string.split(' ') #delimiter is ‘space’ character or ‘ ‘

print(string\_list) #output: ['This', 'is', 'a', 'string.']

print(' '.join(string\_list)) #output: This is a string.

### **What are negative indexes and why are they used?**

* Negative indexes are the indexes from the end of the list or tuple or string.
* **Arr[-1]** means the last element of array **Arr[]**

arr = [1, 2, 3, 4, 5, 6]

#get the last element

print(arr[-1]) #output 6

#get the second last element

print(arr[-2]) #output 5