**FAQ - Introduction to Python**

**1. Can't open the ".ipynb" file because no app is associated with it.**

*Tags- #technicalissue #error #ipynb #colab*

This error message is displayed when the PC can't find an app associated with the ".ipynb" file to open it. These files cannot be opened directly by double-clicking on them and have to be opened using specific software/web links.

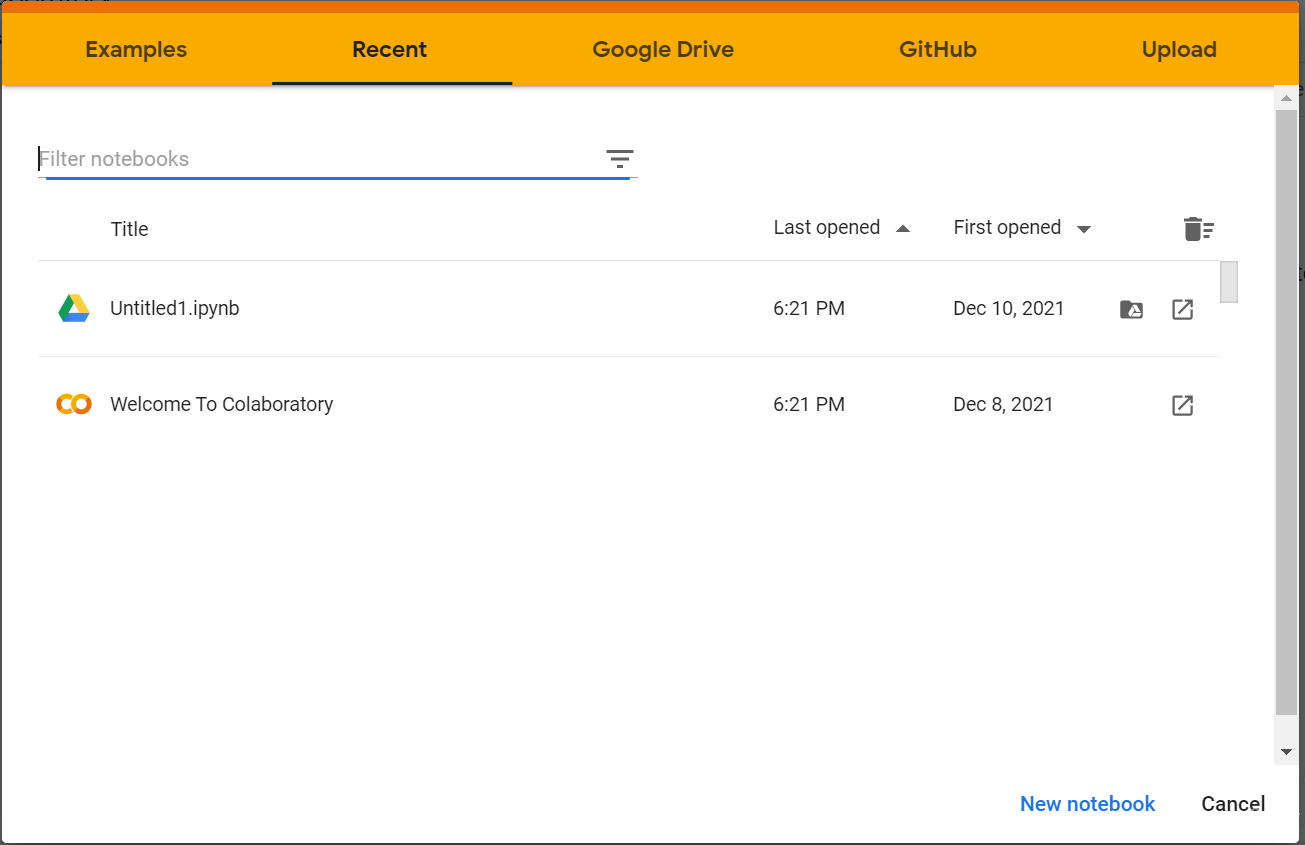
The steps involved in opening a .ipynb file using Google Colab are mentioned below.

**Note**: As Colab implicitly uses Google Drive for storing your notebooks, ensure that you are logged in to your Google account before proceeding further.

Step 1: Open the following URL in your browser

<https://colab.research.google.com/?utm_source=scs-index>

Your browser would display the following screen (assuming that you are logged into your Google account):



Step 2 - Select the “UPLOAD” option (it allows us to upload files from our local directory) and in the dialog box that pops up, go to the folder where the file has been downloaded (usually Downloads) and double click on the file name to open it.

We can also create a new Python notebook (.ipynb file) by selecting the “NEW NOTEBOOK” option shown in the image above.

**2. Is Python case sensitive?**

*Tags- #python #casesensitive*

Yes, Python is a case-sensitive language. Case sensitive means any computer function or a program that differentiates between upper and lowercase letters is called a case-sensitive program so the term ‘**H**ello**W**orld’ and ‘**h**ello**w**orld’ are not the same.

Example: case\_sensitive, CASE\_SENSITIVE, and Case\_Sensitive are each a different variable.

**3. How do I write comments in Python?**

*Tags- #comments #python*

We can use ‘#’ for single-line comments.

# This is a single line comment

For multi-line comments, we can use three quotes.

'''This is   
 a multiline   
 comment in Python'''

**4. What are comparison operators in Python?**

*Tags- #comments #python #comparisonoperators*

Comparison operators in Python allow us to compare variables and give output as Boolean value (True or False).

The table below describes the different comparison operators in Python and shows some examples too. Consider a=5 and b=6.

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| != | If the values of two operands are not equal, then the condition becomes true. | a!=b is true |
| == | If the values of two operands are equal, then the condition becomes true. | a==b is false |
| > | If the value of the left operand is greater than the value of the right operand, then the condition becomes true. | a>b is false |
| >= | If the value of the left operand is greater than or equal to the value of the right operand, then the condition becomes true. | a>=b is false |
| < | If the value of the left operand is less than the value of the right operand, then the condition becomes true. | a<b is true |
| <= | If the value of the left operand is less than or equal to the value of the right operand, then the condition becomes true. | a<=b is true |

**5. What is the difference between a list, a tuple, and a dictionary?**

*Tags- #difference #list #tuple #dictionary #set*

|  |  |  |
| --- | --- | --- |
| **List** | **Tuple** | **Dictionary** |
| A collection of items of any data type    Example:  X=["a", 2, True, "b"] | A collection of items of any data type    Example:  X=("a", 2, True, "b") | A collection of key-value pairs (just like a real-world dictionary has a word and its meaning as pair)    Example:  X={1:'Jan', 2:'Feb', 3:'Mar'} |
| Mutable (can be edited) | Immutable (cannot be edited) | Mutable (can be edited) |
| Supports indexing | Supports indexing | Supports indexing using keys |
| Function: list() | Function: tuple() | Function: dict() |

**6. What do different kinds of brackets do in Python?**

*Tags- #brackets #python*

Primarily, three types of brackets used in Python:

* Parenthesis **() -**to define tuples, order of operations, and function calls
* Square Brackets **[] -**to define mutable datatypes (like lists) and for indexing/slicing
* Curly Brackets **{} -**to define dictionaries

**7. What is indexing?**

*Tags- #indexing #list #tuple*

Indexing is used to access the items/data stored inside data structures like lists, tuples, etc.

lst = [1,2,4,5] #list  
dct = {'one':1, 'two':2, 'three':3} #dictionary  
tupl = (1,2,3,4,5) #tuple

To access the value 2 from all of the above data structures, we can use indexing in the following manner:

lst[1] # use index position to access the value  
dct['two'] #use the key name to access the value  
tupl[1] # use index position to access the value

**8. What do 32 and 64 signify in int32 and int64?**

*Tags- #int32 #int64*

 1) int32 has a range of 2^32. That is it supports integers in the range from [-2,147,483,648 to +2,147,483,647 to 2,147,483,648 to +2,147,483,648].

Thus, it takes 32 bits to be represented i.e it takes 4 bytes of space.

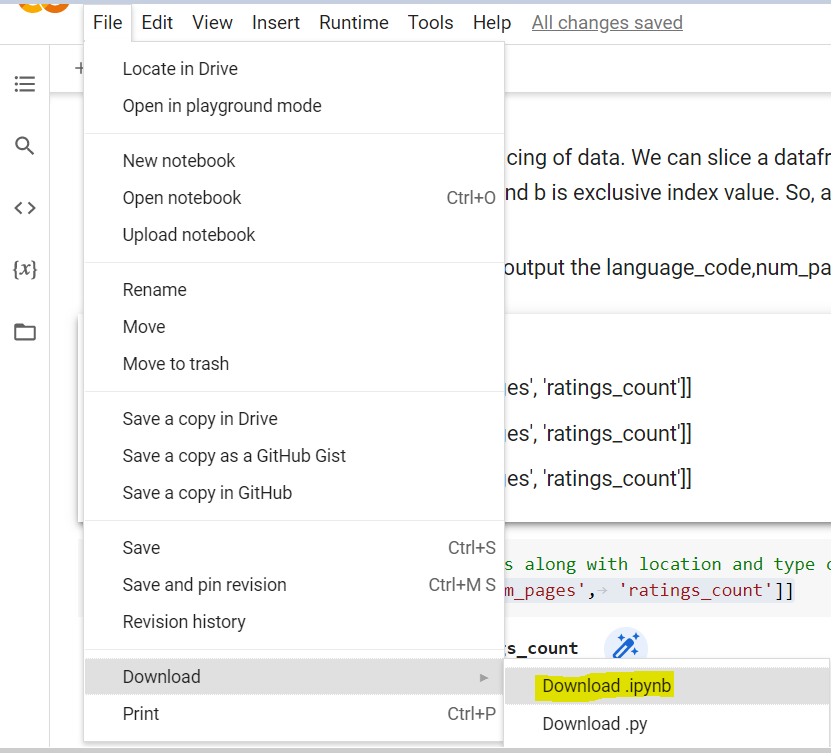
 2) int64 has a range of 2^64. That is, it can hold values between [-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807].

Thus, it takes 64 bits to be represented and thus takes 8 bytes of space.

**9. How to download a .ipynb file?**

*Tags- #technicalissue #ipynb #error #download*

Please click on File >> Download >> Download .ipynb

****

**10. How to convert a Colab notebook with the .ipynb extension to HTML?**

*Tags- #technicalissue #ipynb #html*

Please download [this document](https://olympus.mygreatlearning.com/courses/66857/files/4650221/download?verifier=TP1UTonqAVNPriLtxsrRXA5YHFFxtXlzQ6UtMEMj&wrap=1) and follow the instructions as mentioned in the document.

**11. How to convert the Colab notebook to PDF?**

*Tags- #technicalissue #ipynb #pdf #convert*

You can choose to print the Colab notebook by clicking on the ‘Files’ tab and selecting ‘Print’. In the dialog box that pops up, you can select “Save as PDF” from the drop-down beside the ‘Destination’ option and click on ‘Save’ to save the file in .pdf format.

**12. How does the while loop work?**

*Tags- #whileloop #looping #exit*

Explanation: A while loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

The syntax of a while loop in Python is as follows:

while condition:  
 statement(s)

Here, statement(s) may be a single statement or a block of statements. The condition may be any expression and the loop iterates while the condition is true. When the condition becomes false, program control passes to the line immediately following the loop.

In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.

Example:

count = 0  
while (count < 5):  
 print('The count is:',count)  
 count = count + 1  
print("Good bye!")

Output :

The count is: 0  
The count is: 1  
The count is: 2  
The count is: 3  
The count is: 4  
Good bye!

The block here, consisting of the print and increment statements, is executed repeatedly until the count is no longer less than 5. With each iteration, the current value of the index count is displayed and then increased by 1.

**13. What is the purpose of the ‘i’ in a for loop?**

*Tags- #looping #forloop*

The variable 'i' in a for loop is an iterator that iterates over the various elements in the sequence over which the for loop iterates.

Consider the following example.

for i in range(5, 21, 5):  
 print(i)

In the for loop above, 'i' is the iterator variable that will take values from the sequence (starting from 5 up to 21 with steps of 5, i.e., 5, 10, 15, and 20) one at a time and execute the code (print the value) for each of them.

You can use any letter/name instead of 'i', but make sure to use the same for the rest of the code.

**14. Why does the following error occur and how to rectify it?**

**TypeError**: can only concatenate str (not "int") to str

*Tags- #error #code #int*

A string in Python can only be added (concatenated) to another string. It cannot be added to other datatypes. In order to concatenate a string with a value of a different datatype, we first have to convert the value to a string and then concatenate them.

**15. What is ‘//’ in Python?**

*Tags- #division*

The double slash (//) operator is known as the**floor division operator**. It will divide the value to the left of the operator by the value to its right and only keep the whole number component.

For example, if a=9 and b=5, then a//b = 9//5 = 1. If we used the / operator, we would get a/b =9/5 = 1.8