Modern python for physics students

Introduction

What is Python

- Python is a coding standard
- Python Interpreter is a program that executes Python code
- save the Python code in a file with a .py extension
- the Python Interpreter executes the code line by line

Why Can't I use excel

- You can, and probably should

When to use Python instead of excel

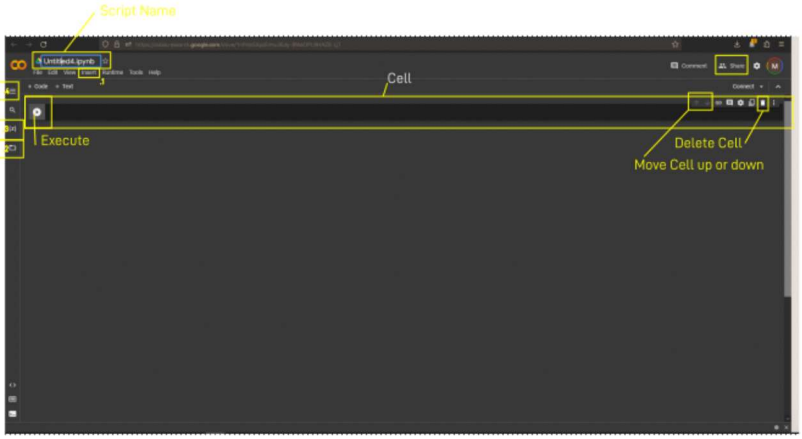
- Large Datasets
- Nested data
- Complicated functions
- Input output
- User Interface
- Web Applications

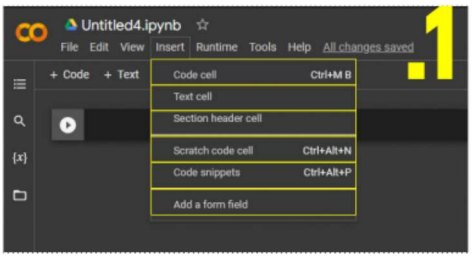
Python Notebooks

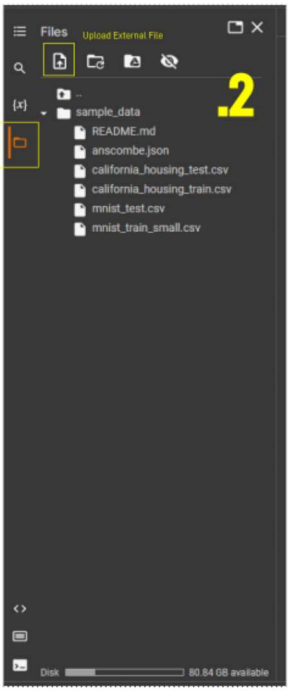
- The Word Processor of python code
- Defacto standard on sharing scientific python code
- File Type: `.ipynb`

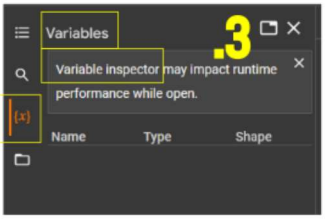
Colab: Google hosted Notebooks (VM)

-









- 

- its actually a VM

Setup

Step 0: The most simple program

-
```
print('hello world')
>>> hello world
```

Step 1: The interpreter

- https://colab.research.google.com/
- file → new notebook
- Change Name to Test.ipynb

- write print('hello world')
- execute
- 

Basic Types

- Comment: arn't executed
  - ```python
    # this is a comment

    a = 1 # this is a line comment

    """
        this is a multi-line comment called a doc string
        it could be many lines
    """
    ```
- string:
  - ```python
    string1 = "micha"
    string2 = 'my house is in the backyard'
    emptyString = ""
    concat = "hello" + "world"
    string_literal = f"my name is {name}"
    ```
- int:
  - ```python
    a = 1
    b = -3
    c = 99999
    d = 0
    ```
- float:
  - ```python
    dist = 0.00341
    temp = 51.012
    val = -31.023
    val2 = 1.00
    ```
- bool:
  - ```python
    a = True
    b = False
    ```
- Custom Types:
  - Objects, Series, DataFrame, ...

Basic Data-structures

- **Variables**
  - a variable is a name that represents a value or an object in memory.
  - Variables are used to store and manipulate data in a program.
  - Variables can be created, accessed or deleted
  - **Create variable**
    - ```python
      ```

```
x = 10
name = 'micha'
temp1 = 0.001
```

- **Access Variables**
  - ```
    print(x)
    ```
- **Change Variables**
  - ```
    x = 10
    print(x)
    x = 100
    print(x)
    ```

Lists

- a list allows you to store a collection of values in a single variable
- values can be accessed, changed, or removed
- Example:
  - ```python
    my_list = [1, 2, 3, 4, 5]
    ```

Slicing

Accessing

- pass an index into square brackets
- ```python
  print(my_list[0])
  var1 = my_list[1]
  last = my_list[-1]
  ```

Changing

- ```python
  my_list[0] = 3
  my_list[1] = 0
  ```

Append

- ```python
  my_list.append(6)
  my_list.append('seven')
  ```

Merge

- ```python
  my_list_2 = [6, 7, 8, 9, 10]
  list_total = my_list + my_list_2
  ```

Nesting

- ```python
  nested_list = [[1,2,4], [5,8,1], [0,0,0]]
  ```

Dicts

- dictionary allows you to store a collection of key-value pairs.
- Each key in a dictionary is unique and maps to a specific value
- ```python
  dict1 = {"key1": "val1", "key2": "val2"}
  ```
  ```

```
my_dict = {'apple': 2.50, 'banana': 1.75, 'orange': 1.00}
```

Loops

- A `for` loop is used to iterate over a sequence of values, such as a list or a range of numbers. Here's an example:

python
```
for i in range(5):
    print(i)
```

python
```
for index, value in enumerate(['micha', 'itamar', 'noga']):
  print(f"index: {index}, value: {value}")
```

Functions

- A function in Python is a block of reusable code that performs a specific task.
- Functions are defined using the `def` keyword, followed by the function name, any parameters (or arguments) that the function takes, and a colon. Here's an example:

python
```
def square(x):
  return x ** 2

def multiply (x,y):
  return x*y

val1 = multipy(5,4.53)
val2 = square(3)
print(val1)
print(val2)
```

python
```
def square(x):
  return x ** 2

def multiply (x,y):
  return x*y

val1 = multiply(5,4.53)
val2 = square(3)

for i in range(100):
  print(square(i))
```

Pandas

- Pandas is a Python library that is used for data analysis and manipulation.
- It provides a wide range of tools for working with structured data, including data frames and series.
- A data frame is a two-dimensional table of data, with columns and rows
- A series is a one-dimensional array-like object that can hold any data type.

python
```
import pandas as pd
```

Create a dataframe

python
```
a = [3,7,0,1,44,12]
b = list(range(6))
```

```
pd_dict = {'column_1':b, 'column_2':a}
df = pd.DataFrame(pd_dict)
print(df)
```

Read excel

python

```
results = pd.read_excel('results.xlsx')
```

write excel

python

```
df.to_excel('output.xlsx', index=False)
```

MatplotLib

python

```
plt.plot(results.Time, results.t1, label='t1')
plt.legend()
plt.show()
```

Unlinked References