Welcome to Data Science Online Bootcamp

Day 0

dφ

Democratizing Data Science Learning

Learning Objectives

1. Adding comments in Python

2. Variables and Operators

3. Input and Output in Python

4. Data Types

5. Numeric and String Data Types

Comments in Python

- When writing code in Python, it's important to make sure that your code can be easily understood by others, say by your friend who wants to see your code.

- Python ignores everything after the hash mark and up to the end of the line. You can insert them anywhere in your code!
- A shortcut for adding comments is by using CTRL + /

```
In [14]: #assign a value 5 to b
         b = 5

In [12]: b

Out[12]: 5
```

To write a comment in Python, simply put # before your desired comment

Operators

- Operators are used to perform operations on variables and values.

- Python supports the following types of operators:

  - Arithmetic Operators
  - Comparison (Relational) Operators
  - Assignment Operators
  - Logical Operators
  - Bitwise Operators
  - Membership Operators
  - Identity Operators

- Let's look at the commonly used types for now.

Operators

Arithmetic Operator

- Used with numeric values to perform common mathematical operations
- The addition and subtraction operations that you performed yesterday come under this category
- Example:

```
x =2
y=3
x +  y # Addition
x * y  # Multiplication
```

```
In [2]: 4**2 #Exponentiation
Out[2]: 16

In [3]: 18%7 #Modulo
Out[3]: 4
```

Operators

Comparison Operator

- Used to compare two values
- Gives a boolean result (True/False)

Numeric Calculations:

```
In [1]: 2 < 3
Out[1]: True

In [2]: 2==3
Out[2]: False

In [3]: 2 <= 3
Out[3]: True
```

Other Comparisons:

```
In [4]: "rahul"< "rohan"
Out[4]: True
```

Operators

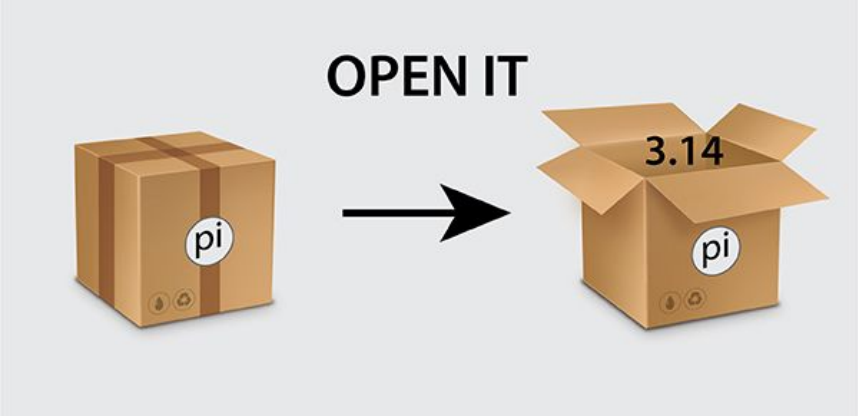
- To learn more about operators, visit: https://www.w3schools.com/python/python_operators.asp

- Cheat Sheet: https://www.codecademy.com/learn/learn-python-3/modules/learn-python3-syntax/cheatsheet

  Tip: *If you are unable to follow run the code and make out the difference*

Variables

- A variable can be considered a storage container for data.
- Every variable will have a name.
- It is a good way to store information while making it easy to refer to that information in our code later.



- For instance, instead of working with the number 3.14, we can assign it to a variable pi and use it as many times as we want later.

Variables

- The equal sign (=) is used to assign values to variables.

- The syntax for assigning values to a variable is as follows:

  Variable name = value or information

- Example:

  x = 5

  y = "John"

Variables and Types

Data Types in Python

- Before we proceed to discuss what data types in Python are, there are some basic questions that we would discuss. What is data?

- Let's say you are going to meet a friend at her office. When you go to visit her office, the guard asks you to make an entry in the register before you enter the office. A typical entry register asks for the following information –

Visitor's name	Visitor's phone number	Visitor's address	Entry time
Karen	32 000 000	Leuven	8:30 am

- The above information that you just provided is data.

Data Types in Python

- We see that the data entry in the previous slide has different varieties:
  - some are english letters,
  - some are numerical digits, and
  - there are some special characters, dash (-) and colon (:).
  - In this example, our data is divided into 4 categories – name, phone number, address, and time.
  - This **categorisation of data**, based on their characteristic & our need, is called **data types**.

Data Types in Python

- Some of the data types in python include:
  - **Integer:** whole numbers, positive or negative numbers. Eg: 100
  - **Float:** Floating-point numbers are real numbers, rational or irrational. In most cases, this means numbers with decimal fractions. Example: 123.45
  - **String:** Strings are sequences of characters, or text, enclosed in quotes. Example: "any text", “karen”

Data Types in Python

- For further reading about **operators and data types**, visit: https://www.dummies.com/programming/python/python-all-in-one-for-dummies-cheat-sheet/

- For practice and different data type examples, visit: https://www.w3schools.com/python/python_datatypes.asp

Getting the Data Type

You can get the data type of any object by using the type( ) function:

```
In [4]: a = 4
        type(a)

Out[4]: int

In [5]: b = 2.5
        type(b)

Out[5]: float

In [6]: text = 'hello'
        type(text)

Out[6]: str

In [7]: boolvar = True
        type(boolvar)

Out[7]: bool
```

Python Strings

- Strings are sequence of characters.
- Let us see some examples of String: My name is Rahul, Rahul, Go home. All these are examples of String.
- In Python, Strings are called **str**.
- There is a specific way of defining String in Python – it is defined within single quote (‘) or double quotes (“).

Accessing String Elements

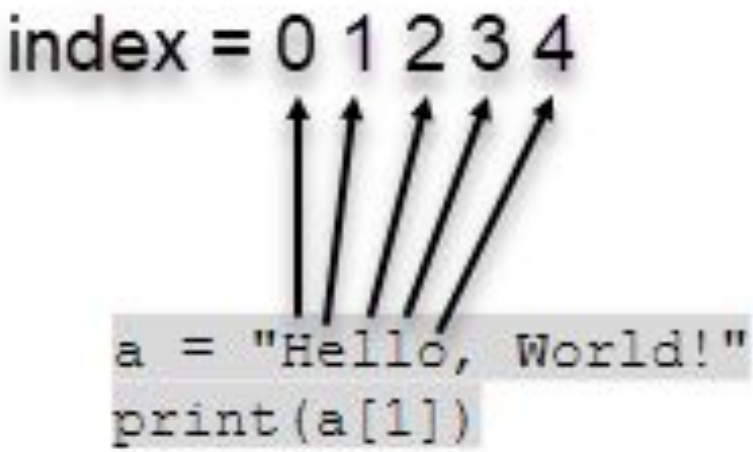
- Square brackets can be used to access elements of the string.
- **Remember that the first character has index 0.**
- Example:

```
a = "Hello, World!"
print(a[1])
```

Index refers to position of a character in a string. In python index number starts from 0.

Will give an output **e**. Can you understand why?

Accessing String Elements



Hope you got the answer to the previous question now!

String Slicing

- We can also call out a range of characters from the string using string slicing.
- Specify the start index and the end index, separated by a colon, to return a part of the string. Note that the character of the end index is not included.
- Suppose we want to print World from the string “Hello World”. We can do so as below:

```
In [3]: a = "Hello, World!"
        print(a[7:12])

        World
```

Negative Indexing

- If we have a long string and we want to pinpoint an item towards the end, we can also count backwards from the end of the string, starting at the index number -1

- Printing ‘r’ from the string :

```
a = "Hello, World!"
print(a[-4])
```

- Get the characters from position 5 to position 1, starting the count from the end of the string:

```
print(a[-5:-2])
```

Will give an output :

```
orl
```

String Concatenation

- String concatenation means adding strings together.

- Use the + character to add a variable to another variable:
- Example:

1.
```
In [9]: 'ab' + 'cd' #concatenation

Out[9]: 'abcd'
```

2.
```
x = "Python is "
y = "awesome"
z =  x + y
print(z)
```

OUTPUT
Python is awesome

String Length

- To get the length of a string, use the len( ) function.
- Getting length of the string a :

```
a = "Hello, World!"
print(len(a))
```

OUTPUT:

```
13
```

String Methods

- Python has a set of built-in methods that you can use on strings.

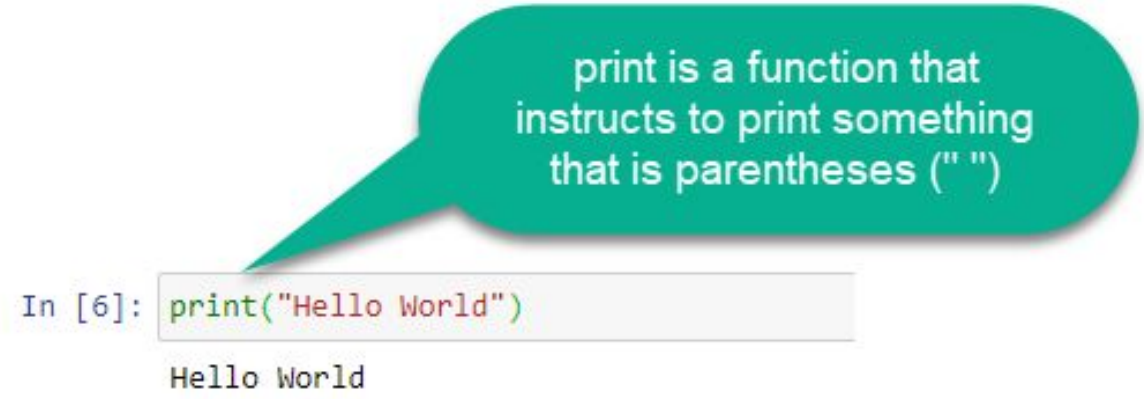
- **Must learn:** Learn about important string methods from the below cheatsheet:

  https://www.codecademy.com/learn/learn-python-3/modules/learn-python3-strings/cheatsheet

  Tip: *If you are unable to follow, run the code and make out the difference*

Input and Output in Python

Print Function

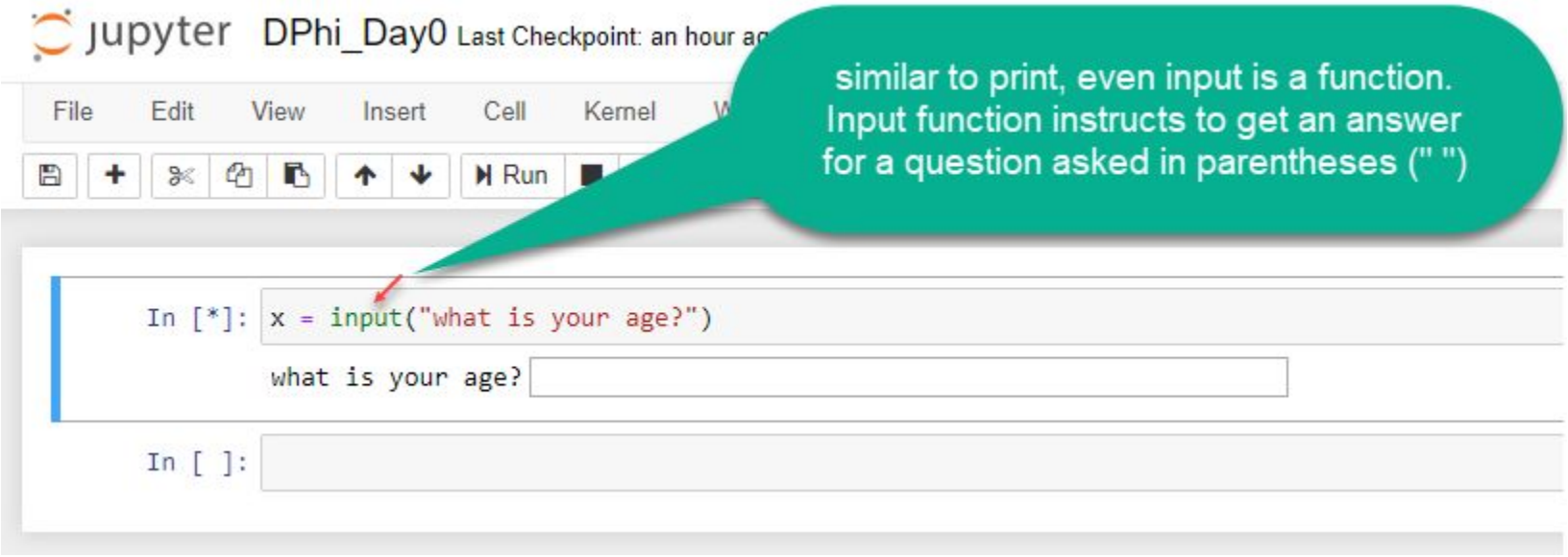


- You've already worked with the print function in your first Python program to print "Hello World"

- Example:
  print("This is a print statement")
  print(2)

Input and Output in Python

Input Function



Let's Practice!

1. Write a program to take a number from the user using a prompt "Enter a number : ".

2. Print the number taken by the user in the previous question.
   Hint: assign your program to a variable in question 1

3. Print the square of that number.

4. Create a string with value "Program" and print 'o' from it.
   Hint: recall indexing

5. Print the last 4 characters of the above string.

6. Check if the string contains all lowercase letters.

7. Print the type of the string

Let's Practice!

8. Take in first name and last name as input and store it in the variables firstname and lastname

   Suppose

   firstname = "abc"

   lastname = "xyz",

   Your task is to print the following:

   Hello abc xyz! You just delved into python.

   Hint: Google how to print multiple arguments/variables in python

That’s it for the day. Thank you!

Feel free to post any queries in the #help channel on Slack