Welcome to Data Science Online Bootcamp

Day 4

dφ

Democratizing Data Science Learning

Learning Objectives

Functions

Packages

Python Methods

Functions?

- This something we have been indirectly using since **Day0!**

- print( ) and input( ) are some functions that you've already used. And these are inbuilt functions.

```
print ("hello world")
input ("enter your age")
```

- If you want, you can even create your own function to get some desired output.

- Let's learn more about functions in the video mentioned in the next slide.

Functions


DataCamp
Intro to Python for Data Science
Functions
Nothing new!
type()
Piece of reusable code
Solves particular task
Call function instead of writing code yourself

Examples of in-built functions

```
In [25]: x
Out[25]: [0, 2, 4, 5, 6]
```

```
In [28]: max(x)
Out[28]: 6

In [29]: len(x)
Out[29]: 5

In [30]: round(1.68,1)
Out[30]: 1.7

In [31]: round(1.68)
Out[31]: 2

In [32]: sorted(x, reverse=True)
Out[32]: [6, 5, 4, 2, 0]
```

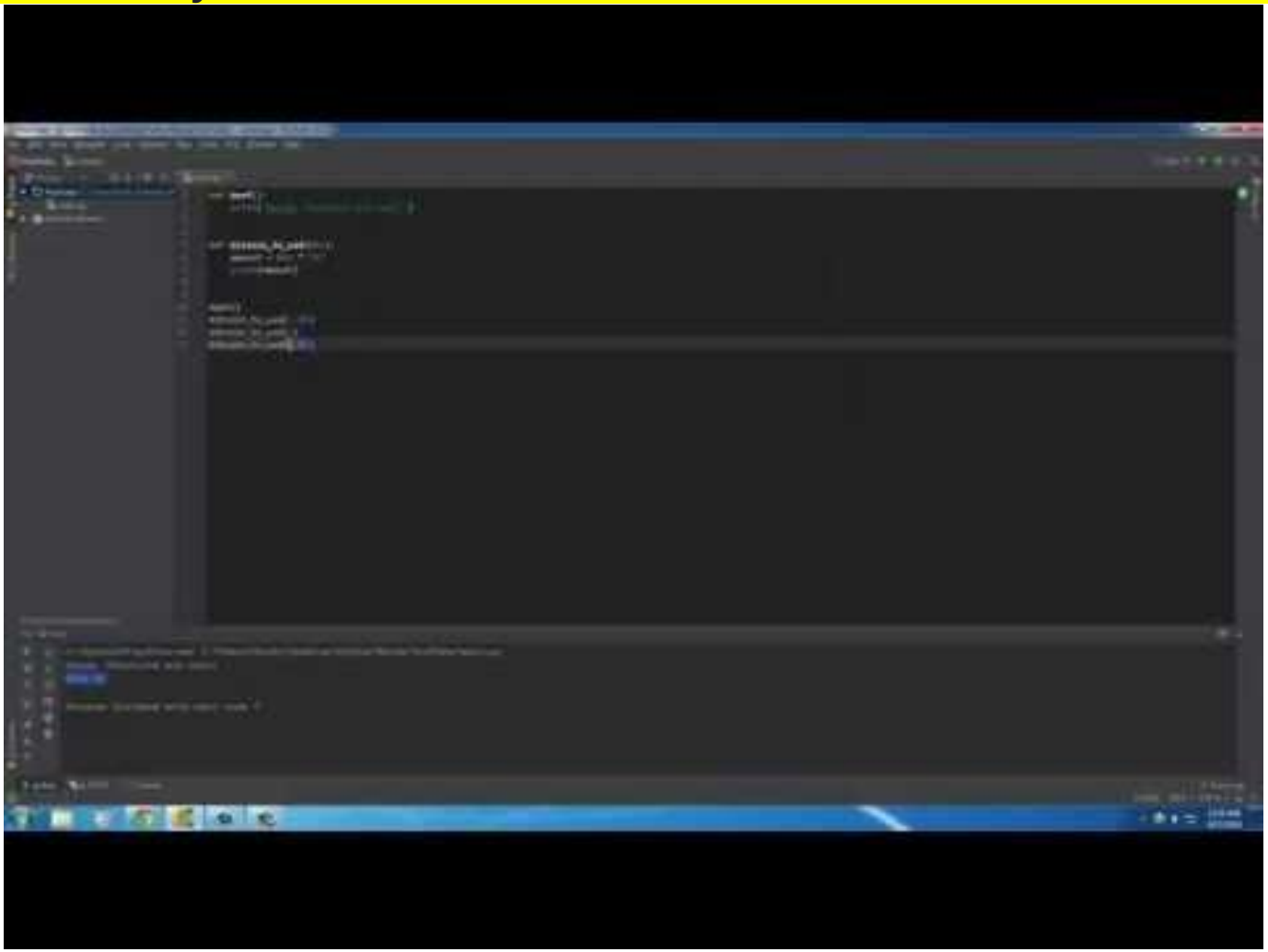
Find out the purpose of the 2nd argument in round function.

Let's see how to write functions

- In Python a function is defined using the def keyword
- In the next tutorial, we will be learning:
  - How to write a function
  - How to call a function - Of course, if we have created a function, we need to use it somewhere by calling it right? ;)
  - We will also learn how to write a function that would convert bitcoin into USD :P
- **Note:** The tutor used some other user-friendly local python ide (similar to Colab or jupyter notebook), so don't press the panic button looking at the new coding interface. You can comfortably run the same code on collab/jupyter notebook (or any python environment) and it will work.

Writing functions

Note: The tutor used a user-friendly python ide (similar to Colab/jupyter notebook), so don't press the panic button looking at the new coding interface. You can comfortably run the same code on Colab/Notebook and it will work..

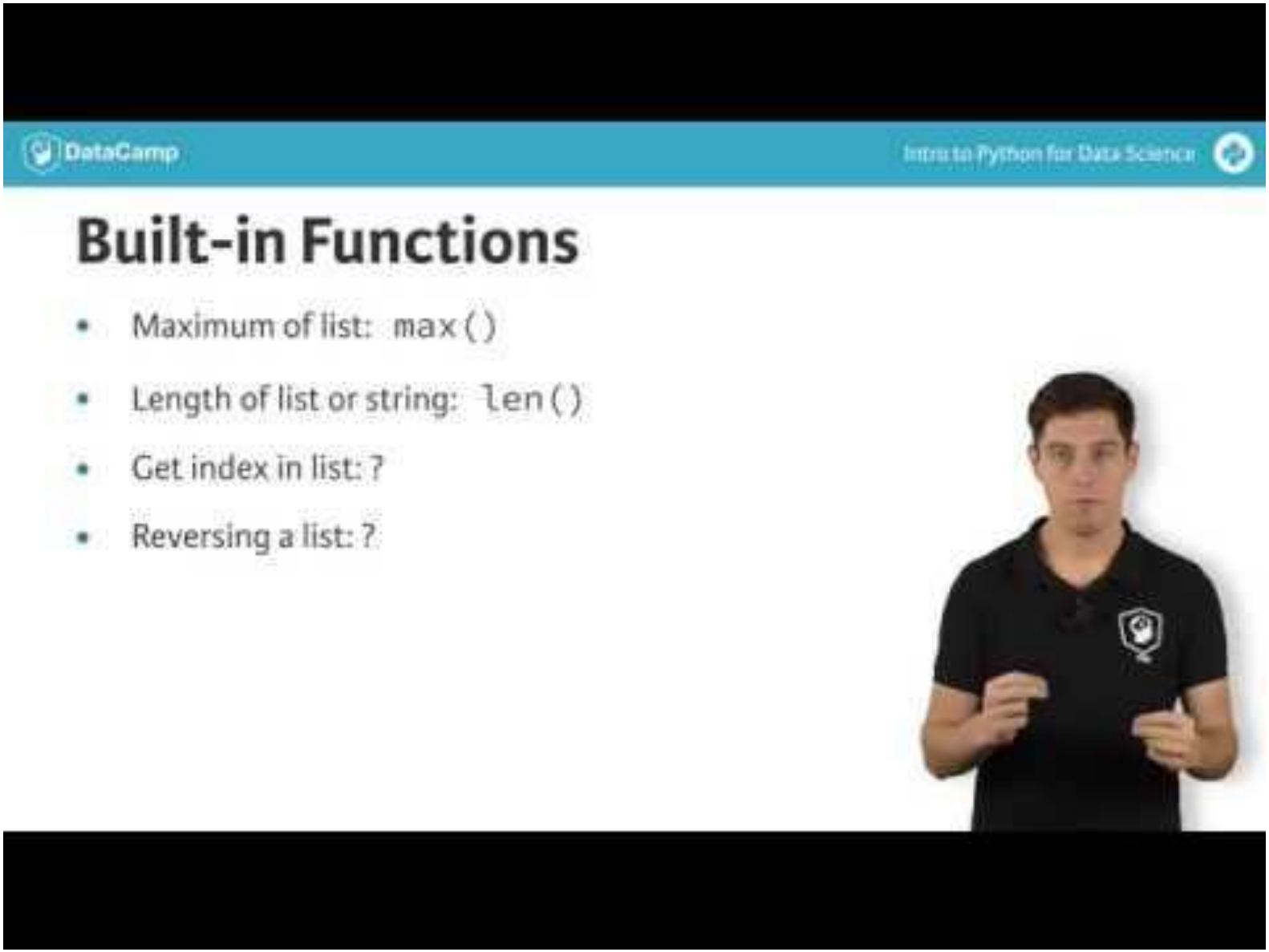


Interested to learn more about Functions?

To learn about all the exciting things Python Functions can do, please visit the below link.

https://www.w3schools.com/python/python_functions.asp

Methods in Python



Methods in Python

- In Python, everything is an object (be it a list, string etc).
- Objects have various methods associated with it (depending on type of the object)

```
In [33]: x.index(2)
Out[33]: 1

In [34]: x.count(1)
Out[34]: 0

In [35]: word = 'abcdef'
         word.capitalize()
Out[35]: 'Abcdef'

In [36]: word.replace("c", "xy")
Out[36]: 'abxydef'
```

Functions and Methods: are they same?

- You might have a question bugging you: “Why on Earth do we have both functions and methods, when they practically do the same thing?”

- Firstly, start with the obvious. There is a clear difference in the syntax:
  - A function looks like this: function(something)
  - And a method looks like this: something.method()

- Namely: a method always belongs to an object (e.g. in the x.index(2) method .index() needed the x object to be applicable), while a function doesn’t necessarily.

- All methods are functions, but not all functions are methods!

- If this makes no sense to you (yet), don’t you worry. I promise, the idea will grow on you as you use Python more and more – especially when you start to define your own functions and methods.

Python Packages

Need for Python Packages

If we keep all of our code in the same file, it will result in:

- Huge code base: Ends up messy
- Lots of code you won't use
- Maintenance problem

Packages

- If we take an example of ourselves, we don't usually store all of our files on our computer in the same location. We use a well-organized hierarchy of directories/folders for easier access.

- Similar files are kept in the same directory, for example, we may keep all the songs in the "music" directory. Analogous to this, Python has packages for directories and modules for files.

- As our application program grows larger in size with a lot of modules, we place similar modules in one package and different modules in different packages. This makes a project (program) easy to manage and conceptually clear.

- Similarly, as a directory can contain subdirectories and files, a Python package can have sub-packages and modules.

Packages

- Directory/Folder of Python Scripts
- Where each script is a module that performs a specific function
- We can specify functions, methods, types in a script
- Thousands of packages are available in python
- For data science, the commonly used packages are:
    - Numpy: Working with arrays
    - Matplotlib: Data Visualisation
    - Scikit-learn: ML

Importing packages/modules

Follow this resource to learn about how to incorporate packages and make more sophisticated programs.

- https://www.learnpython.org/en/Modules_and_Packages
- https://www.digitalocean.com/community/tutorials/how-to-import-modules-in-python-3
- 40 sec video that explains importing package on jupyter notebook: https://www.youtube.com/watch?v=9FX9iRRwaYU

Tip: What is a Module?

A module is same as a code library - a file containing a set of functions you want to include in your application

Let's Practice!

1. Write a function addition() such that it can accept two variables and calculate the sum of it.

```
def addition(a, b):
    # You should write your function here

addition(5, 6)
#this result should print 11
```

2. Write a function display( ) such that it can accept two variables - a string s and a number n and display the string n number of times.

3. Write a Python function to sum all the numbers in a list.

   Sample List : [8, 2, 3, 0, 7]

   Expected Output : 20

   Hint: *There is a built-in method for this purpose*

Submitting your quiz answers

Learners who enrolled for the bootcamp can access DPhi Learning Platform to submit their day-wise module quizzes.

- **Link to learning platform:** https://learn.dphi.tech/

That’s it for the day. Thank you!

Remember Google is your friend in need! And ofcourse #help channel and coaches on slack!