Complete Python Developer in 2020 : Zero to Mastery

by Andrei Neagoie

01 15/08/2020 Thu

Wk-31 • (213-152)

-	5	12	19	26
6	13	20	27	
7	14	21	28	
1	8	15	22	29
2	9	16	23	30
3	10	17	24	31
4	11	18	25	-

Human Language ⟷ Machine Language (0's, 1's)

- High Level language (eg. Python, Java)
- Low level language (eg. Assembly)

Programming language → Interpreter (line by line) or compiler (all at once)

Byte code

Machine Language ← Virtual Machine ← Byte code

→ Repl.it, glot.io } websites for running online code. We can use them directly instead of downloading all the required softwares.

→ The Python language is itself written in other languages such as C, Java, .net, Python.
When we download Python from Python.org, we actually download cPython (written in C).

→ Data Types

1. Fundamental Data types

int	bool	list
float	str	tuple

Education is a progressive discovery of our own ignorance. - Will Durant

September '19 Wed 04
Wk-36 • (247-118)

Set
dict
complex (it is complex no. which is real and imaginary no.)

2. Custom Types
Class (Here we create our own types)

3. Specialized data types
These are special packages and modules, that we can use from Library

4. None (Nothing)

+ Naming convention
- For variables
  - Start with a letter or underscore
  - Can contain numbers, letters or underscore
  - It is case sensitive
  - For constant use all caps. eg. PI = 3.14
  - For private variables start with underscore.
  - We should not use double underscore to start the name. These are called 'dunder' variables
    eg. __name__

+ Expression : Part of code which produces a value
age = 20 + 5 * 3
(20 + 5 * 3 : expression)

It is impossible to walk rapidly and be unhappy. - Howard Murphy

Statement : It is an entire line of code.

05 Thu

Wk-36 • (248-117)

30	2	9	16	23
–	3	10	17	24
–	4	11	18	25
–	5	12	19	26
–	6	13	20	27
–	7	14	21	28
1	8	15	22	29

→ Strings are immutable.
we cannot re-assign a part of the strings. Once created it will remain same. Unless we create a new string with same variable name.
eg. name = "saurabh"
name[0] = "g" → error

name = "saurabh"
name = "gaurabh" → this works.
(or)
name = name + "Kumar"
print(name) → Saurabh Kumar

→ Build in Function Vs. Build in Methods
- Methods are same as functions, but they are owned by something.
- for eg. Python's String methods like find(), capitalize(), etc can be used only on strings.
and methods have different syntax. They are preceded by dot (.).
eg. name = "saurabh"
name = name.capitalize()

- Similarly there are methods for lists.
such as pop(), clear(), remove(), append(), insert(), extend(), sort(), copy(), count().
But here, be carefull as to what they return.
Some methods return 'none', some returns values.

September '19 — Fri 06 — Wk-36 • (249-116)

→ List, Dictionary are data structures.
- List is in ordered memory.
- Dictionary is in random place in memory.
- Dictionary key has to be immutable, hence it can be a boolean value, a string, or a number, but it cannot be a list. It can be ~~tuple~~ tuple as well. A Key has to be unique as well. Otherwise it will overwrite the previous key: value pair.

→ Tuple is another data structures.
They are immutable list. and are ordered.

→ Set
It is an unordered collection of unique objects.
Hence we cannot use my_set[0], like we do with lists

→ Truthy and Falsy
>>> bool(....)
This will always equals 'True'. Expect for the empty sets such as:

None
0
0.0
[]
''
{}
()

for eg. bool('') = False
bool('Hello') = True

07 Sat

Wk-36 • (250-115)

30	2	9	16	23
–	3	10	17	24
–	4	11	18	25
–	5	12	19	26
–	6	13	20	27
–	7	14	21	28
1	8	15	22	29

→ Ternary Operator

>>> condition_if_true if condition else condition_if_else

→ >
<
==
!=
>=
<=
and
or
not

eg. print (1 > 0) → True
print (1 == 0) → False
print ('a' == 'a') → True

print (7 > 6 > 5 > 4) → True
print (1 == 1 and 0 == 0) → True
print (not 1 == 1) → False
print (not (0 == 1)) → True

→ ASCII

>>> ord('a') → 97
>>> chr(97) → a

08 Sun

Wk-36 • (251-114)

→ print (1 == 1) → True
print (1 is 1) → True
print ([] == []) → True
print ([] is []) → False

'is' checks whether they both are in the same memory location
'==' checks for equality.

Laughing is the sensation of feeling good all over and showing it principally in one spot. - Josh Billings

September '19

Mon 09

Wk-37 • (252-113)

with strings and no's, memory location is same.
for eg. print ('a' is 'a') → True
but for lists, it creates another ~~location~~ list in a different memory location.

→ Iterable

List, tuple, set, dictionary, string, these are iterable. It means we can check one by one each item in the collection.

eg. for i in [1,2,3,4]:
    print (i)

for i in range(0,100):
    print (i)

→ Print() by default ends with a new line "\n".
we can solve this with:

print ("*", end = "")

↳ now it will end with none, so next print statement will continue from that same position.

→ Function

>>> def function_name():
    ...
    ...
    ...

>>> function_name() → to call a function
>>> print (function_name) → this is without the curly brackets, hence it will not take any action. It will just print the memory location of the fun

10 Tue

Wk-37 • (253-112)

30	2	9	16	23
–	3	10	17	24
–	4	11	18	25
–	5	12	19	26
–	6	13	20	27
–	7	14	21	28
1	8	15	22	29

Muharram (India)

→ Parameters Vs. Arguments

```
def function_name(age, name):
    print(f'Your age is {age}, and name is {name}')

function_name("27", "Saurabh")
```

age, name → Positional Parameters.

"27", "Saurabh" → Positional Arguments.

```
function_name(name="John", age="20")
```

This is called as Keyword arguments, here we don't need to follow the order.

Default parameters

```
def function_name(age="27", name="Jojo")
    . . .

function_name()
function_name("20")
```

Here if we don't give the required arguments, the default parameters will be considered.

→ To pass multiple positional and keyword arguments use '*args' and '**kwargs'.

Life must be measured by thought and action, not by time. - John Lubbock, Conductor

September '19 — Wed 18 — Wk-38 • (261-104)

→ Object Oriented Programming

- Everything in Python is an object.
- We use different methods on the objects to perform an action.
- Objects have methods and attributes that we can access using dot (.) method.
  for eg. 'hello'.replace("e","0")

- A class is instanciated into instances (i.e. objects).
  for eg. my_list = [1, 2, 3]
  my_list.clear()
  here we used class 'list', to create our own object 'my_list'. And then we used a class "list" method 'clear()' using dot extention.
  similarly we create our own classes and objects.

→ Class

```
class NameOfClass():
    class_attribute = 'value'
    def __init__(self, param1, param2):
        self.param1 = param1
        self.param2 = param2
    def method(self):
        # code
    @classmethod
    def cls_method(cls, param1, param2):
        # code
    @staticmethod
    def stc_method(param1, param2):
        # code
```

"My one aim was to do a thing well and to excel if possible. - Josephine Demott Robinson

19 Thu

Wk-38 • (262-103)

30 2 9 16 23
– 3 10 17 24
– 4 11 18 25
– 5 12 19 26
– 6 13 20 27
– 7 14 21 28
1 8 15 22 29

→ 4 pillars of OOP

a) Encapsulation
- It is the binding of data & functions that manipulate that data. And we encapsulates it into one big object so that we keep everything in that box.
- Data & functions are nothing but attributes & methods

b) Abstraction
- It is hiding of info. and giving only what is necessary eg. we use len() function, we know how to use it, but we don't have to know how it calculates the length. We can go find it, but we don't need to.

c) Inheritance

d) Polymorphism
↓ many ↓ forms

→ Dunder or Magic methods

→ Functional Programming
It separates data and functions (i.e. attributes and methods) and store them separately in different blocks.

Nature gave us one tongue and two ears so we could hear twice as much as we speak. - Epictetus

September '19 — Wed 25 — Wk-39 • (268-097)

→ Pure Function

1. With same I/P, it will always produce same O/P no matter how many times we pass the I/P.
2. Function should not produce any side effect. Meaning it should not touch anything in the outside world. It should not interact with outside world, by using outside variables (other than the one passed to it), or by printing something.

→ Map, filter, zip and reduce.

→ Lambda expressions

These are anonymous functions (without any name), used only one time, so it runs only once, and doesn't get stored in the memory.

→ Decorators

- These are used to superboost a function.
- We define them as

```
def my_decorator(func):
    def wrap_func():
        func()
    return wrap_func

@my_decorator
def function_name():
    . . .
```

No law or ordinance is mightier than understanding. - Pluto

26 Thu

Wk-39 • (269-096)

30	2	9	16	23
–	3	10	17	24
–	4	11	18	25
–	5	12	19	26
–	6	13	20	27
–	7	14	21	28
1	8	15	22	29

→ Build in Exceptions

Exceptions are the errors in the code.

- Syntax error
- Name error : variable name is not defined
- Type error : for eg adding 'int' and 'str'
- Index error : Index out of range
- Key error : when we use a key of a dictionary that doesn't exist.
- Zero division error :
- Value error : invalid literal for int()

→ Generators

They generate a sequence of iterables.
for eg. range(100)
but they don't store all the values in it in the memory. It just generate one value at a time, and don't keep the previous values in the memory.

```
def generator_func(num):
    for i in range(num):
        yield i

for item in generator_func(100):
```

No man is justified in doing evil on the ground of expediency. - Theodore Roosevelt

September '19 — Fri 27 — Wk-39 • (270-095)

Grand Final Eve (Australia), Meskel (Ethiopia)

→ Modules

- Each .py is a module.
- Naming convention is snake case (same as that of a variable).
- When we import, we are importing a .py file from the same folder as of the present file. And python compile the importing file using a compiler, and keeps the compiled version in cache memory. It creates a '__pycache__' folder in that folder only, which will have the compiled file in '.pyc' format i.e. cpython compiler file. If we change something in the importing file, then the file is again compiled, otherwise the python will use the compiled file from cache memory every time.

Package

- Package is a folder containing modules (ie. '.py' file
- A package should have a '__init__.py' file in it it can be completely empty, but that folder should have that file, then only python realises that it is a package.

→ __name__

It returns the module name.

for a file → utility

for a file in a folder → shopping.shopping_cart

for the file which is being run → __main__

No matter how great your words may be, you will be judged by your actions. - Brahma Kumaris

28 Sat
Wk-39 • (271-094)

30 2 9 16 23
– 3 10 17 24
– 4 11 18 25
– 5 12 19 26
– 6 13 20 27
– 7 14 21 28
1 8 15 22 29

Mahalaya (India)

→ Built in modules (Standard Library)
When we install python, we get access to all these built in modules
All we have to do is use import to use them.

→ Pypi.org
We can see and use packages developed by other coders on this website. for eg. matlab, pandas, etc.

→ From python 3.7, dictionaries are ordered.
The order in which we give key : value pair, that order is guaranteed to be stored.
Set however are still unordered.

→ Debugging ~~tools~~
- Linting
It is built in IDE's. When we type code, then only it gives us error messages.
- Use of IDE / editors
- PEP8 : Code formating, which is just spacing acc. to standard, for easy understanding and visuals
- import pdb
pdb.set_trace()

29 Sun
Wk-39 • (272-093)

Not everything that is faced can be changed; but nothing can be changed until it is faced. - James Baldwin

October '19

Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday

01 | 02 | 03 | 04 | 05 | 06

→ print("\n How \n are \n You") → How
are
You

print(r"\n How\n are\n You") → \n How\n are\n You

07 | 08 | 09 | 10 | 11 | 12 | 13

↓
raw it tells python to use the string as it is

→ Resume Builder

14 | 15 | 16 | 17 | 18 | 19 | 20

1. Github
2. Website
3. 1→2 big projects
4. Blog

21 | 22 | 23 | 24 | 25 | 26 | 27

→ Testing

> python3 -m unittest

This command will run all the test files at once present in that directory.

28 | 29 | 30 | 31

> python3 -m unittest -v

This will do the same as above. And in addition gives us info about the methods/test it is running.

PREVIOUS MONTH

30 2 9 16 23
– 3 10 17 24
– 4 11 18 25
– 5 12 19 26
– 6 13 20 27
– 7 14 21 28
1 8 15 22 29

NEXT MONTH

– 4 11 18 25
– 5 12 19 26
– 6 13 20 27
– 7 14 21 28
1 8 15 22 29
2 9 16 23 30
3 10 17 24

01 Tue

Wk-40 • (274-091)

–	7	14	21	28
1	8	15	22	29
2	9	16	23	30
3	10	17	24	31
4	11	18	25	–
5	12	19	26	–
6	13	20	27	–

National Day (Hong Kong), Independence Day (Nigeria)

→ Scripting

> pip install Pillow

To work on images.

↳ or "OpenCV" → this is the most popular and advance ~~module~~ package to work on images / videos. Used in machine learning

→ Web Scraping

- A website has 3 files behind the scenes:
  a) Java script → for behaviours such as submitting form, drop down menu
  b) HTML → text and content
  c) CSS → For colors and styling

  browser requests for these files from the machine (server) and displays the result.
- In a website url, add '/robots.txt' this will list all the things we are allowed / disallowed to scrap ethically.
- All the ~~title~~ links not mentioned in the 'disallow' section, we are allowed to scrap.
- This is how Search engines like Google works, they look for the searched item in the website allowed scrapping links and post the result.

→ Package : Beautiful Soup → used to clean up the data
: Requests → Used to get data from the website

Framework : Scrapy → Beautiful Soup has limited functionalities with Scrapy, we can use much more things, collect huge amount of data and store more efficiently.

Nothing is so strong as gentleness; nothing is so gentle as real strength. - Francis de Sales

October '19 — Wed 02 — Wk-40 • (275-090)

Gandhi Jayanti (India)

We only webscrape when the website is not giving us the API.

→ Web development

Framework : Flask → Simple, clean, small
django → Very big, complicated

To create a virtual environment
> python3 -m venv venv
↓
name of virtual env.

This will create a virtual env. in the pwd

To activate :
> <venv>\Scripts\Activate.ps1
↓
path to the virtual env.

if permission error occurs use the below command:
> Set-ExecutionPolicy -ExecutionPolicy unrestricted -Scope CurrentUser

To deactivate :
> deactivate

→ Steps to run Flask
> pip install Flask
> $env: FLASK_APP = "server.py"
↳ file name

One of the most difficult things to give away is kindness - it is usually returned. - Cort R. Flint

> $env: FLASK_ENV = "development"
> flask run

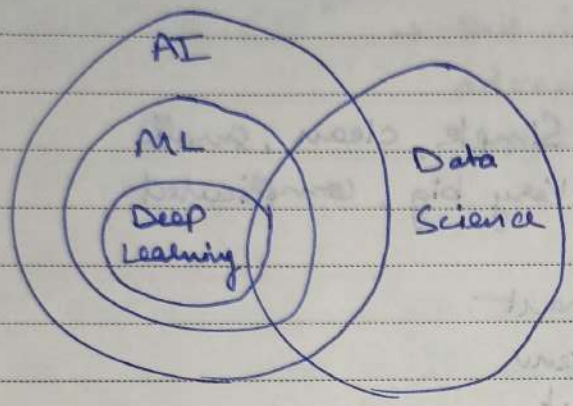
03 Thu

Wk-40 • (276-089)

–	7	14	21	28
1	8	15	22	29
2	9	16	23	30
3	10	17	24	31
4	11	18	25	–
5	12	19	26	–
6	13	20	27	–

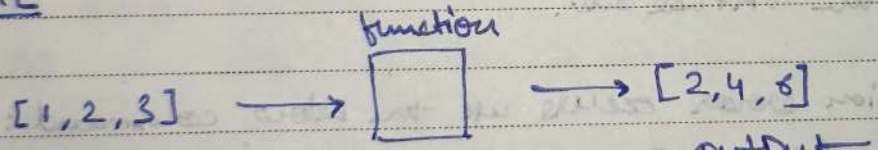
Day of German Unity (Germany)

Machine Learning & Data Science

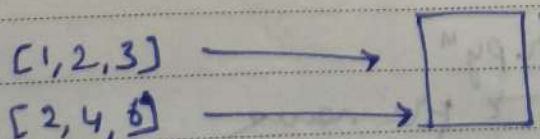


Deep learning is just one of the technique to do ML. Data Science is all about analysing data. It highly overlaps with ML.

ML



Normally we write the I/P and function, and our machine uses that info to give us the O/P.
But in ML, we give I/P and O/P to machine and it writes the function.



One of the secrets of life is to make stepping stones out of stumbling blocks. - Jack Penn

Here function is synonimous to model, Algorithm, Bot.

October '19 Fri 04

Wk-40 • (277-088)

* History of analysing data

Spreadsheets → Relational DataBase (SQL) → Big Data (mongo DB), unstructured data

↓

ML

→ Types of ML

ML

↓ ↓ ↓

Supervised
- classification
- regression

Unsupervised
- clustering
- Association rule learning

Reinforcement
- Skill acquisition
real life learning

ML is all about predicting O/P based on I/P data.

→ ML Steps :

1. Import the data
2. Clean the data
3. Split data into ~~Testing~~ Training set & Test set
4. Create a model
   or choose 3-4 models and see which one gives the best result
5. Check output
6. Improve.

Opportunities are usually disguised as hard work, so most people don't recognize them. - Ann Landers

05 Sat

Wk-40 • (278-087)

–	7	14	21	28
1	8	15	22	29
2	9	16	23	30
3	10	17	24	31
4	11	18	25	–
5	12	19	26	–
6	13	20	27	–

Tools:

1. NumPy → for working with Lists and arrays
2. Pandas → for working with tabular data
3. scikit-learn → to create the model. Its pre-built with algorithms such as classification, regression, clustering, etc
4. matplotlib → allows us to visualize the data in a better way.
5. jupyter notebooks
6. kaggle → here they have massive amount of data for free
7. seaborn → built on top of 'matplotlib', to simplify it. Easy to use.
8. bokeh → for interactive visualization.

06 Sun

Wk-40 • (279-086)

Others can stop you temporarily; only you can do it permanently. - Don Ward