

# 0x09. Python - Everything is object

## PythonOOP

- By: Guillaume
- Weight: 1
- Project over - took place from Jun 28, 2022 6:00 AM to Jun 29, 2022 6:00 AM
- An auto review will be launched at the deadline

### *In a nutshell...*

- **Auto QA review:** 61.1/94 mandatory & 36.35/59 optional
- **Altogether: 105.05%**
  - Mandatory: 65.0%
  - Optional: 61.61%
  - Calculation:  $65.0\% + (65.0\% * 61.61\%) == 105.05\%$



## Background Context

Now that we understand that everything is an object and have a little bit of knowledge, let's pause and look a little bit closer at how Python works with different types of objects.

BTW, have you ever modified a variable without knowing it or wanting to? I mean:

```
>>> a = 1
```

```
>>> b = a
>>> a = 2
>>> b
1
>>>
```

OK. But what about this?

```
>>> l = [1, 2, 3]
>>> m = l
>>> l[0] = 'x'
>>> m
['x', 2, 3]
>>>
```



This project is a little bit different than the usual projects. The first part is only questions about Python's specificity like "What would be the result of...". You should **read all documentation first (as usual :))**, then take the time to **think and brainstorm with your peers** about what you think and why. **Try to do this without coding anything for a few hours.**

Trying examples in the Python interpreter will give you most of the answers without having to think about it. **Don't go this route.** First read, then think, then brainstorm together. Only then you can test in the interpreter.

It's important that you truly understand the reasons behind the answers of all those tasks so that you can apply the same logic to other variables and other variable types.

Note that during interviews for Python positions, **you will most likely have to answer to these type of questions.**

All your answers should be only one line in a file. No space before or after the answer.

# Resources

## Read or watch:

- 9.10. Objects and values
- 9.11. Aliasing
- Immutable vs mutable types
- Mutation (*Only this chapter*)
- 9.12. Cloning lists
- Python tuples: immutable but potentially changing

# Learning Objectives

At the end of this project, you are expected to be able to **explain to anyone, without the help of Google:**

## General

- Why Python programming is awesome
- What is an object
- What is the difference between a class and an object or instance
- What is the difference between immutable object and mutable object
- What is a reference
- What is an assignment
- What is an alias
- How to know if two variables are identical
- How to know if two variables are linked to the same object
- How to display the variable identifier (which is the memory address in the CPython implementation)
- What is mutable and immutable
- What are the built-in mutable types
- What are the built-in immutable types
- How does Python pass variables to functions

## Copyright - Plagiarism

- You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
- You will not be able to meet the objectives of this or any following project by copying and pasting someone else's work.
- You are not allowed to publish any content of this project.
- Any form of plagiarism is strictly forbidden and will result in removal from the program.

# Requirements

## Python Scripts

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted/compiled on Ubuntu 20.04 LTS using python3 (version 3.8.5)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/python3`
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should use the pycodestyle (version `2.8.*`)
- All your files must be executable
- The length of your files will be tested using `wc`

## `.txt` Answer Files

- Only one line
- No Shebang
- All your files should end with a new line

## Tasks

### 0. Who am I?

mandatory

Score: 65.0% (Checks completed: 100.0%)

What function would you use to print the type of an object?

Write the name of the function in the file, without `()`.

#### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `0-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 1. Where are you?

mandatory

Score: 65.0% (Checks completed: 100.0%)

How do you get the variable identifier (which is the memory address in the CPython implementation)?

Write the name of the function in the file, without ().

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `1-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 2. Right count

mandatory

Score: 65.0% (Checks completed: 100.0%)

In the following code, do `a` and `b` point to the same object? Answer with `Yes` or `No`.

```
>>> a = 89
>>> b = 100
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `2-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 3. Right count =

mandatory

Score: 65.0% (Checks completed: 100.0%)

In the following code, do `a` and `b` point to the same object? Answer with `Yes` or `No`.

```
>>> a = 89
>>> b = 89
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `3-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 4. Right count =

mandatory

Score: 65.0% (Checks completed: 100.0%)

In the following code, do `a` and `b` point to the same object? Answer with `Yes` or `No`.

```
>>> a = 89
>>> b = a
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `4-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 5. Right count =+

mandatory

Score: 65.0% (Checks completed: 100.0%)

In the following code, do `a` and `b` point to the same object? Answer with `Yes` or `No`.

```
>>> a = 89
>>> b = a + 1
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `5-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 6. Is equal

mandatory

Score: 65.0% (Checks completed: 100.0%)

What do these 3 lines print?

```
>>> s1 = "Best School"
>>> s2 = s1
>>> print(s1 == s2)
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `6-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 7. Is the same

mandatory

Score: 65.0% (Checks completed: 100.0%)

What do these 3 lines print?

```
>>> s1 = "Best"
>>> s2 = s1
>>> print(s1 is s2)
```

### Repo:

- GitHub repository: [alx-higher\\_level\\_programming](#)
- Directory: [0x09-python-everything\\_is\\_object](#)
- File: [7-answer.txt](#)

Done! Help Check your code Get a sandbox QA Review

## 8. Is really equal

mandatory

Score: 65.0% (Checks completed: 100.0%)

What do these 3 lines print?

```
>>> s1 = "Best School"
>>> s2 = "Best School"
>>> print(s1 == s2)
```

### Repo:

- GitHub repository: [alx-higher\\_level\\_programming](#)
- Directory: [0x09-python-everything\\_is\\_object](#)
- File: [8-answer.txt](#)

Done! Help Check your code Get a sandbox QA Review

## 9. Is really the same

mandatory

Score: 65.0% (Checks completed: 100.0%)

What do these 3 lines print?

```
>>> s1 = "Best School"
```



```
>>> s2 = "Best School"
>>> print(s1 is s2)
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `9-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 10. And with a list, is it equal

mandatory

Score: 65.0% (Checks completed: 100.0%)

What do these 3 lines print?

```
>>> l1 = [1, 2, 3]
>>> l2 = [1, 2, 3]
>>> print(l1 == l2)
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `10-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 11. And with a list, is it the same

mandatory

Score: 65.0% (Checks completed: 100.0%)

What do these 3 lines print?

```
>>> l1 = [1, 2, 3]
>>> l2 = [1, 2, 3]
>>> print(l1 is l2)
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `11-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 12. And with a list, is it really equal

mandatory

Score: 65.0% (Checks completed: 100.0%)

What do these 3 lines print?

```
>>> l1 = [1, 2, 3]
>>> l2 = l1
>>> print(l1 == l2)
```

### Repo:

- GitHub repository: [alx-higher\\_level\\_programming](#)
- Directory: [0x09-python-everything\\_is\\_object](#)
- File: [12-answer.txt](#)

Done! Help Check your code Get a sandbox QA Review

## 13. And with a list, is it really the same

mandatory

Score: 65.0% (Checks completed: 100.0%)

What do these 3 lines print?

```
>>> l1 = [1, 2, 3]
>>> l2 = l1
>>> print(l1 is l2)
```

### Repo:

- GitHub repository: [alx-higher\\_level\\_programming](#)
- Directory: [0x09-python-everything\\_is\\_object](#)
- File: [13-answer.txt](#)

Done! Help Check your code Get a sandbox QA Review

## 14. List append

mandatory

Score: 65.0% (Checks completed: 100.0%)

What does this script print?

```
l1 = [1, 2, 3]
```

```
l2 = l1
l1.append(4)
print(l2)
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `14-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 15. List add

mandatory

Score: 65.0% (Checks completed: 100.0%)

What does this script print?

```
l1 = [1, 2, 3]
l2 = l1
l1 = l1 + [4]
print(l2)
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `15-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 16. Integer incrementation

mandatory

Score: 65.0% (Checks completed: 100.0%)

What does this script print?

```
def increment(n):
    n += 1

a = 1
increment(a)
```

```
print(a)
```

### Repo:

- GitHub repository: [alx-higher\\_level\\_programming](#)
- Directory: [0x09-python-everything\\_is\\_object](#)
- File: [16-answer.txt](#)

Done! Help Check your code Get a sandbox QA Review

## 17. List incrementation

mandatory

Score: 65.0% (Checks completed: 100.0%)

What does this script print?

```
def increment(n):
    n.append(4)

l = [1, 2, 3]
increment(l)
print(l)
```

### Repo:

- GitHub repository: [alx-higher\\_level\\_programming](#)
- Directory: [0x09-python-everything\\_is\\_object](#)
- File: [17-answer.txt](#)

Done! Help Check your code Get a sandbox QA Review

## 18. List assignation

mandatory

Score: 65.0% (Checks completed: 100.0%)

What does this script print?

```
def assign_value(n, v):
    n = v

l1 = [1, 2, 3]
l2 = [4, 5, 6]
assign_value(l1, l2)
```

```
print(l1)
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `18-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 19. Copy a list object

mandatory

Score: 65.0% (Checks completed: 100.0%)

Write a function `def copy_list(l):` that returns a **copy** of a list.

- The input list can contain any type of objects
- Your file should be maximum 3-line long (no documentation needed)
- You are not allowed to import any module

```
guillaume@ubuntu:~/0x09$ cat 19-main.py
#!/usr/bin/python3
copy_list = __import__('19-copy_list').copy_list

my_list = [1, 2, 3]
print(my_list)

new_list = copy_list(my_list)

print(my_list)
print(new_list)

print(new_list == my_list)
print(new_list is my_list)

guillaume@ubuntu:~/0x09$ ./19-main.py
[1, 2, 3]
[1, 2, 3]
[1, 2, 3]
```

```
True
False
guillaume@ubuntu:~/0x09$ wc -l 19-copy_list.py
3 19-copy_list.py
guillaume@ubuntu:~/0x09$
```

## No test cases needed

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `19-copy_list.py`

Done! Help Check your code Get a sandbox QA Review

## 20. Tuple or not?

mandatory

Score: 65.0% (Checks completed: 100.0%)

```
a = ()
```

Is `a` a tuple? Answer with `Yes` or `No`.

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `20-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 21. Tuple or not?

mandatory

Score: 65.0% (Checks completed: 100.0%)

```
a = (1, 2)
```

Is `a` a tuple? Answer with `Yes` or `No`.

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `21-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 22. Tuple or not?

mandatory

Score: 65.0% (Checks completed: 100.0%)

```
a = (1)
```

Is `a` a tuple? Answer with `Yes` or `No`.

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `22-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 23. Tuple or not?

mandatory

Score: 65.0% (Checks completed: 100.0%)

```
a = (1, )
```

Is `a` a tuple? Answer with `Yes` or `No`.

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `23-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 24. Who I am?

mandatory

Score: 65.0% (Checks completed: 100.0%)

What does this script print?

```
a = (1)
b = (1)
a is b
```

### Repo:

- GitHub repository: `alx-higher_level_programming`

- Directory: `0x09-python-everything_is_object`
- File: `24-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 25. Tuple or not

mandatory

Score: 65.0% (Checks completed: 100.0%)

What does this script print?

```
a = (1, 2)
b = (1, 2)
a is b
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `25-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 26. Empty is not empty

mandatory

Score: 65.0% (Checks completed: 100.0%)

What does this script print?

```
a = ()
b = ()
a is b
```

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `26-answer.txt`

Done! Help Check your code Get a sandbox QA Review

## 27. Still the same?

mandatory

Score: 65.0% (Checks completed: 100.0%)



```
>>> id(a)
139926795932424

>>> a
[1, 2, 3, 4]

>>> a = a + [5]

>>> id(a)
```

Will the last line of this script print `139926795932424`? Answer with `Yes` or `No`.

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `27-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 28. Same or not?

`mandatory`

Score: 65.0% (Checks completed: 100.0%)

```
>>> a
[1, 2, 3]

>>> id (a)
139926795932424

>>> a += [4]

>>> id(a)
```

Will the last line of this script print `139926795932424`? Answer with `Yes` or `No`.

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `28-answer.txt`

Done! Help Check your code Get a sandbox QA Review

### 29. #pythonic

`#advanced`

Score: 65.0% (Checks completed: 100.0%)

Write a function `magic_string()` that returns a string “BestSchool” n times the number of the iteration (see code):

- Format: see example
- Your file should be maximum 4-line long (no documentation needed)
- You are not allowed to import any module

```
guillaume@ubuntu:~/0x09$ cat 100-main.py
#!/usr/bin/python3
magic_string = __import__('100-magic_string').magic_string

for i in range(10):
    print(magic_string())

guillaume@ubuntu:~/0x09$ ./100-main.py | cat -e
BestSchool$
BestSchool, BestSchool$
BestSchool, BestSchool, BestSchool$
BestSchool, BestSchool, BestSchool, BestSchool$
BestSchool, BestSchool, BestSchool, BestSchool, BestSchool$
BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool$
BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool$
BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool$
BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool$
BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool, BestSchool$

guillaume@ubuntu:~/0x09$ wc -l 100-magic_string.py
4 100-magic_string.py
guillaume@ubuntu:~/0x09$
```

**No test cases needed**

**Repo:**

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `100-magic_string.py`

Done! Help Check your code Get a sandbox QA Review

### 30. Low memory cost

#advanced

Score: 45.0% (Checks completed: 90.0%)

Write a class `LockedClass` with no class or object attribute, that prevents the user from dynamically creating new instance attributes, except if the new instance attribute is called `first_name`.

- You are not allowed to import any module

```
guillaume@ubuntu:~/0x09$ cat 101-main.py
#!/usr/bin/python3
LockedClass = __import__('101-locked_class').LockedClass

lc = LockedClass()
lc.first_name = "John"
try:
    lc.last_name = "Snow"
except Exception as e:
    print("{} {}".format(e.__class__.__name__, e))

guillaume@ubuntu:~/0x09$ ./101-main.py
[AttributeError] 'LockedClass' object has no attribute 'last_name'
guillaume@ubuntu:~/0x09$
```

**No test cases needed**

**Repo:**

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `101-locked_class.py`

Done? Help Check your code Ask for a new correction Get a sandbox QA Review

### 31. int 1/3

#advanced

Score: 65.0% (Checks completed: 100.0%)

```
julien@ubuntu:/python3$ cat int.py
```

```
a = 1
b = 1
julien@ubuntu:/python3$
```

Assuming we are using a CPython implementation of Python3 with default options/configuration:

- How many int objects are created by the execution of the first line of the script? (103-line1.txt)
- How many int objects are created by the execution of the second line of the script (103-line2.txt)

### Repo:

- GitHub repository: [alx-higher\\_level\\_programming](#)
- Directory: [0x09-python-everything\\_is\\_object](#)
- File: [103-line1.txt](#), [103-line2.txt](#)

Done! Help Check your code Get a sandbox QA Review

### 32. int 2/3

#advanced

Score: 65.0% (Checks completed: 100.0%)

```
julien@ubuntu:/python3$ cat int.py
a = 1024
b = 1024
del a
del b
c = 1024
julien@ubuntu:/python3$
```

Assuming we are using a CPython implementation of Python3 with default options/configuration:

- How many int objects are created by the execution of the first line of the script? (104-line1.txt)
- How many int objects are created by the execution of the second line of the script (104-line2.txt)
- After the execution of line 3, is the int object pointed by **a** deleted? Answer with **Yes** or **No** (104-line3.txt)
- After the execution of line 4, is the int object pointed by **b** deleted? Answer with **Yes** or **No** (104-line4.txt)

- How many int objects are created by the execution of the last line of the script (`104-line5.txt`)

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `104-line1.txt`, `104-line2.txt`, `104-line3.txt`, `104-line4.txt`, `104-line5.txt`

Done! Help Check your code Get a sandbox QA Review

### 33. int 3/3

#advanced

Score: 65.0% (Checks completed: 100.0%)

```
julien@twix:/tmp/so$ cat int.py
print("I")
print("Love")
print("Python")
julien@ubuntu:/tmp/so$
```

Assuming we are using a CPython implementation of Python3 with default options/configuration:

- Before the execution of line 2 (`print("Love")`), how many int objects have been created and are still in memory? (`105-line1.txt`)
- Why? (optional blog post :))

Hint: `NSMALLPOSINTS`, `NSMALLNEGINTS`

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `105-line1.txt`

Done! Help Check your code Get a sandbox QA Review

### 34. Clear strings

#advanced

Score: 65.0% (Checks completed: 100.0%)

```
guillaume@ubuntu:/python3$ cat string.py
a = "SCHL"
```

```
b = "SCHL"
del a
del b
c = "SCHL"
guillaume@ubuntu:/python3$
```

Assuming we are using a CPython implementation of Python3 with default options/configuration (For answers with numbers use integers, don't spell out the word):

- How many string objects are created by the execution of the first line of the script? (106-line1.txt)
- How many string objects are created by the execution of the second line of the script (106-line2.txt)
- After the execution of line 3, is the string object pointed by `a` deleted? Answer with `Yes` or `No` (106-line3.txt)
- After the execution of line 4, is the string object pointed by `b` deleted? Answer with `Yes` or `No` (106-line4.txt)
- How many string objects are created by the execution of the last line of the script (106-line5.txt)

### Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x09-python-everything_is_object`
- File: `106-line1.txt`, `106-line2.txt`, `106-line3.txt`, `106-line4.txt`, `106-line5.txt`

Done! Help Check your code Get a sandbox QA Review

Copyright © 2022 ALX, All rights reserved.