PYTHON COURSE

ENTRY LEVEL

Basics of programming in Python 3.10

This course will cover part of the arguments found in PCEP™ – Certified Entry-Level Python Programmer Certification

INTRODUCING VARIABLES

VARIABLES IN PYTHON

Used to store values in memory in order to build dynamical programs

You can think variables as boxes in which a certain value of some type is stored.

- Python, as a **dynamically-typed language**, **don't require** programmers to **declare** the **variable type**, so a variable can hold any data-type, from the simpler ones to the most complex.
- **Data-type of the value stored in a variable can change** during the program if needed.
- In Python, you don't need a specific keyword **to declare a variable, you only need to give it a name and a value.**
- You **must assign a value to the variable when declared.**

PYTHON

VARIABLES IN PYTHON

to **define a variable in a python program**, you'll need a so called "operator". In this case, **you'll need the "assignment operator", =**

```
# Assign a float value to the variable x
x = 34.5

# Assign a string value to the variable name
name = "Patric"

# Assign an integer value to the variable age
age = 34

# Assign a boolean value to the variable its_young
its_young = False
```

variable's name

literal value assigned to the variable

assignment operator

VARIABLES IN PYTHON

when you define variables in a program, you create **“references” in memory that “points” to the assigned value**, so, in the same program, **you can use declared variable’s identifiers (names) to perform operations on/with.**

```
# Print in console values stored in choosen variables

print(name)
print(age)
```

we’ll use variable’s name instead of literal value

```
> python3 variables.py
Patric
34
```

PYTHON

VARIABLES IN PYTHON

if you'll **try to use a variable that is not defined in the program**, you'll get a **NameError**

```
# No variable called b is defined in this program

print(b)
```

```
> python3 test_error.py
Traceback (most recent call last):
  File "/Users/andreaceccarelli/Desktop/test_error.py", line 17, in <module>
    print(b)
NameError: name 'b' is not defined
```

MEMORY MANAGEMENT IN PYTHON

We're starting to understand how to store values in memory to be used in programs, so a little digression on how python memory management works we'll be helpful to write efficient and smarter code.

AUTOMATIC MEMORY MANAGEMENT

- Python is an **high-level programming language**, it uses an automatic system to manage memory, and will not require programmers to handle on their own.
- For this purpose, python uses a dedicated portion of computer's memory called **"Heap"**
- The Interpreter will automatically allocate and deallocate memory in the heap using a method called "Reference counting Garbage Collection".

MEMORY MANAGEMENT IN PYTHON

GARBAGE COLLECTION

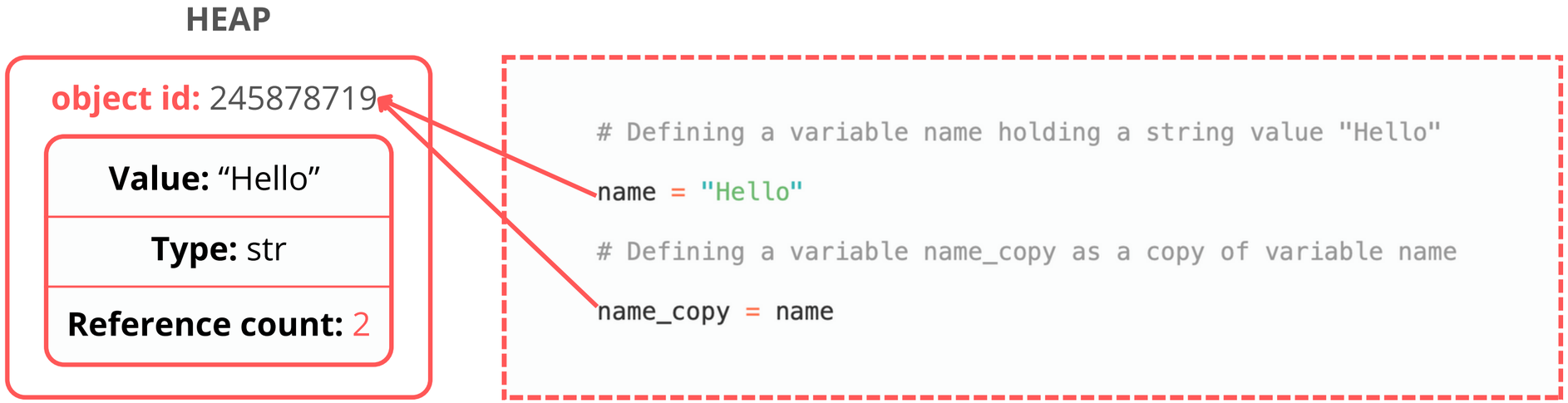
Automatically reclaiming memory that is no longer in use by the program

- To perform this particular process, python uses a method called “Reference Counting”
- Each object in the heap, has a reference count, it is incremented when a new reference to that object is created and decremented when that reference is deleted or goes out of scope.
- When a an object has 0 as reference count, the garbage collector automatically delete it from the heap and frees that portion of memory

MEMORY MANAGEMENT IN PYTHON

When you define a new variable, a new reference to the object in memory (the value) is created, and that object's references count is incremented.

The reference is so considerable as an address to a space in memory which contains a certain object



*functions, methods and their variables are stored in a separate memory space called Stack.

MEMORY MANAGEMENT IN PYTHON

you can **check the id of the object pointed by variables** by using the **id(variable_name) function**

```
# Defining a variable name holding a string value "Hello"

name = "Hello"

# Defining a variable name_copy as a copy of variable name

name_copy = name


print(id(name))
print(id(name_copy))
```

```
> python3 id_check.py
4372676912
4372676912
```

same id: we've copied the reference to the object

VARIABLES IN PYTHON: TYPE CHECK

you can also **check variable** assigned value's **type** by using the function **type(variable_name)**

```
# Defining a variable name holding a string value "Hello"

name = "Hello"

print(type(name))
```

```
> python3 type_check.py
<class 'str'>
```

as you can see, **built-in data-types are treated as object (instances of classes)** and so have peculiar methods (a special type of function) that we can use to perform operations.

PYTHON

VARIABLES IN PYTHON: DEL

Using **del keyword**, you can **delete variables (and so the reference)** from your program.

```python
# Defining a variable name holding a string value "Hello"

name = "Hello"

print(name)

del name

print(name)
```

```
> python3 try_del.py
Hello
Traceback (most recent call last):
  File "/Users/andreaceccarelli/Desktop/try_del.py", line 9, in <module>
    print(name)
NameError: name 'name' is not defined
```

NAMING CONVENTIONS

NAMING: GUIDELINES

when defining **variable names**, you should **follow some guidelines** in order to make your **code more readable:**

- Variable names should be lowercase.
- No spaces are allowed, only upper/lower letters, underscores and numbers (except for the first character).
- Use underscores to separate words.
- Try using “self-reading names”.

NAMING: PEP-8

All **naming and good-coding conventions for python** are contained in a **document** called "**PEP-8**"

PYTHON ENHANCEMENT PURPOSAL - 8

Written by Guido Van Rossum, creator of python, in 2001.

peps.python.org/pep-0008/

QUESTION TIME

PYTHON

QUESTION 1:

What is the output of the following python program:

```
name = "Ryan"

print(type(name))
```

1. Ryan
2. <class: ‘str’>
3. code is erroneous, will generate NameError
4. name

PYTHON

QUESTION 2:

What is the output of the following python program:

```
name = "Ryan"

print("name")
```

1. Ryan
2. <class: ‘str’>
3. code is erroneous, will generate NameError
4. name

PYTHON

QUESTION 3:

What is the output of the following python program:

```
description =
"""this is
my fantasy"""

print(description)
```

1. code is erroneous, will produce SyntaxError.
2. description
3. code is erroneous, will generate NameError.
4. this is
   my fantasy

PYTHON

QUESTION 4:

at the end of this program, what are b and c variables' values:

```
a = 56.0
b = 43
c = b
b = a
```

1. 56.0, 43
2. 56.0, 56.0
3. 43, 43
4. 43, 56.0

PYTHON

QUESTION 5:

Choose correct answer:

1. Variables can store every data-type. Stored data-type can change during program
2. Value can be considered a reference that points to a variable
3. Only Numerical types can be stored in variables
4. Variables can store every data-type. Store data-type can’t change during program

PYTHON

QUESTION 6:

Select correct statements:

1. Variables can be defined without assigning a value
2. Variable names can be used as reference to the stored value to perform operations.
3. = is used as the “assignment operator” in order to assign values to variables
4. Trying to use an undefined variable will result in a SyntaxError