Object-Oriented programming for Beginners

LEAPS Computing 2018

Teun Krikke
tfk2@hw.ac.uk
(slides originally made by Rajiv Murali)
Heriot-Watt University

Learning Outline

- **Setting up PyCharm**
  - Simple HelloWorld program.
- **Basic Object-Oriented Programming Concept**
  - What is an Object?
  - What is a Class?
- **Language Basics**
  - Variables
  - Operators
  - Expressions, Statements and Blocks
  - Control Flow Statements
    i. Conditional Statements,Loop statements
  - Methods

Simple PyCharm Start-Up

Creating a Python Project in PyCharm IDE

- Step 1: Start PyCharm -> "Create new project"
- Step 3: Enter a Project name, e.g. “PythonTutorial”.
  - Step 3a: Change the project interpreter to new environtment using virtualenv
  - Step 3b: change the base interpreter to Anaconda Python 3 (drop down menu)
  - Step 3c: and click “Create”
- Step 4: Right-click on the project folder and select New > Python File.

Creating a Python Project in PyCharm IDE

- Step 5: Enter “HelloWorldDemo” as the name, and click “Finish”.
- Step 6: Edit the HelloWorldDemo.py file as follows:

```
class HelloWorldDemo(object) :
    def __init__(self):
        super(HelloWorldDemo, self).__init__()
    def print_hello(self):
        print('Hello World')
if __name__ == '__main__':
    hwd = HelloWorldDemo()
    hwd.print_hello()
```

- Step 7: Run the HelloWorldDemo.

Assignment HelloWorld

- In this assignment I want you to use the HelloWorldDemo Class to:
  - output the following message
    - : “Hello World! my name is #your name here#!”

Object-Oriented Programming

Object-Oriented Programming

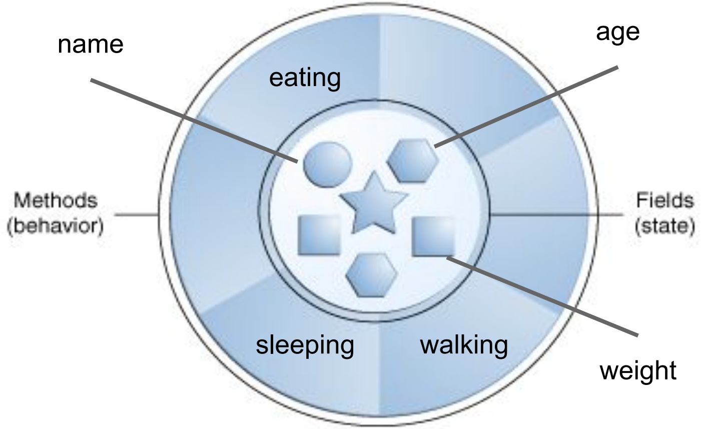
- Python is an Object-Oriented Programming Language.
- Before you begin writing Python code, you will need to learn a few basic object-oriented concepts.
- This lesson will introduce you to:
    - what is an Object?
    - what is a Class?
- To help better understand these concepts, we will relate them to examples Python code that you will run in your PyCharm IDE.

What is an Object?

- Look around right now and you'll find many examples of real-world objects:
    - your desk, your books, your computers and even yourselves.
- Objects share two characteristics: State and Behaviour.
- For example an object Person may have:
    - state - name, age, weight, etc.
    - behavior - walking, eating, sleeping, etc.
- Take a minute and think of some examples of Objects around you, with some state and behaviour.

Objects in Software

- Similar to real-world objects, software objects also consist of state and behaviour.
- A software object stores its...
  - states in fields, also known as variables in most programming languages.
  - while its behaviour is exposed through methods.



an Object - Student

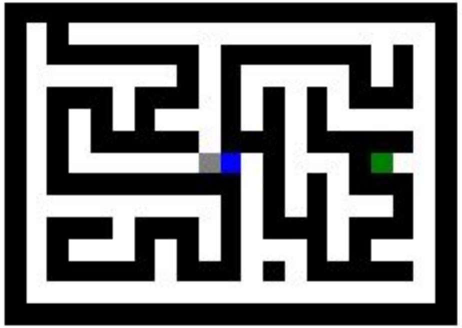
- Methods help to operate an object's internal state.

What is a Class?

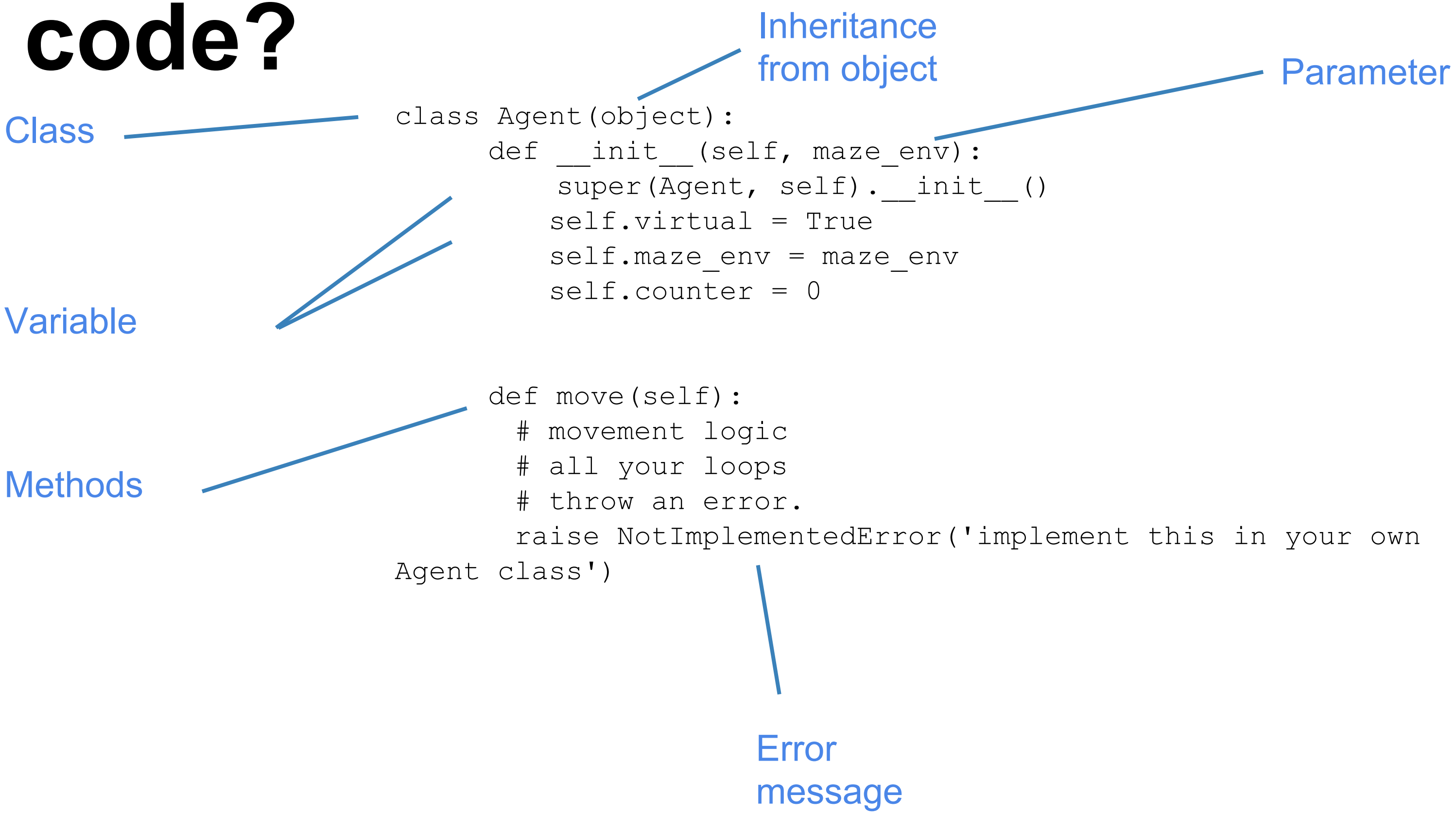
- Often there are many individual objects of the same kind.
- There are billions of people in existence, and all people fall under the kind human.
- A Class helps to capture the common features of objects.
- And from a Class, many instances of Objects may be created.
- In object-oriented terms, we say that an Object person is an instance of a Class Human.

Maze agents

- Agent finds its own way through a maze to the goal
- Maze = world, Agent = person/robot



What does Class Agent look like in code?



Inheritance from object

Parameter

Class

Variable

Methods

Error message

```
class Agent(object):
     def __init__(self, maze_env):
         super(Agent, self).__init__()
         self.virtual = True
         self.maze_env = maze_env
         self.counter = 0


     def move(self):
       # movement logic
       # all your loops
       # throw an error.
       raise NotImplementedError('implement this in your own
Agent class')
```

The self

- Being philosophical
- Reference to the current instance of the class
- Calling methods and variables on instance, eg

```
class Agent(object):
    def __init__(self, codename):
        self.codename = codename

>>> frank = Agent("007")
>>> frank.codename
"007"
>>> Agent("008").codename
"008"
>>> teun = Agent("009")
>>> teun.codename
"009"
```

Creating an object from Agent Class…

- We want to create a custom Agent from our agent class...
- But the Agent class is not a complete application!
- It does not contain a *main* method.
- The responsibility of creating and using agents from the agent class belongs to another application….

Agent creation

- The Agent_factory Class helps creates an instance frank of the Agent class. We then call frank’s move() method.
- Because the Agent is in a different file we need to import it. Which basically says to the Agent_factory where to find the class Agent. In our case in a file called Agent.

```
from Agent import Agent

if __name__ == '__main__':
    frank = Agent()
    frank.move()
```

Program Execution

1. Creates an instance frank from the Agent Class.
2. Moves frank.

Assignment AgentDemo

- In this assignment I want you to use the AgentDemo Class to:
    - 1. Move frank forward 4 steps
    - 2. Move frank in a square.
    - 3. Print the *maze* information.

Extra install

```
Pip install flask
pip install jupyter_contrib_nbextensions
```

Cozmo Animation Explorer

Try to run a couple of animations in the animation explorer.

To start go to the Cozmo-Animation-Explorer and right click animation-explorer.py in the menu click run

Cozmo explorer tool

Play around with the explorer tool

To start go to the Cozmo-Explorer-tool and right click explorer-tool.py in the menu click run

Assignment 1

Pick up a cube and shake it.

Are there any events shown in the browser?

Assignment 2

Using the explorer tool and cozmo pick up a cube and place it on another

Are there any events shown in the browser?

Assignment 2

Drive the cozmo to a cliff (**IT SHOULD NOT FALL OF THE DESK!!!!**)

Are there any events shown in the browser?