Python Seminar 2018

Ana Bulović
Jorin Diemer
Jens Hahn

Plan for today

1. Homework
2. How to use Python
3. Definition of a class
4. Homework

How to use Python

- Ipython
- Sublime + terminal
- PyCharm (IDEs)
- Jupyter notebook

Ipython console

- Code highlighting + autocorrection
- Fast and easy
- Perfect to test code snippets

Sublime (+ terminal)

- Code highlighting + autocorrection
- Very powerful text editor (also LaTeX)
- Code execution in Sublime is possible
- Several cursors handling

Jupyter notebook

- Highlighting + autocorrection + docu
- Integration of graphics, text (HTML…)
- Magic functions
- Data analysis

PyCharm

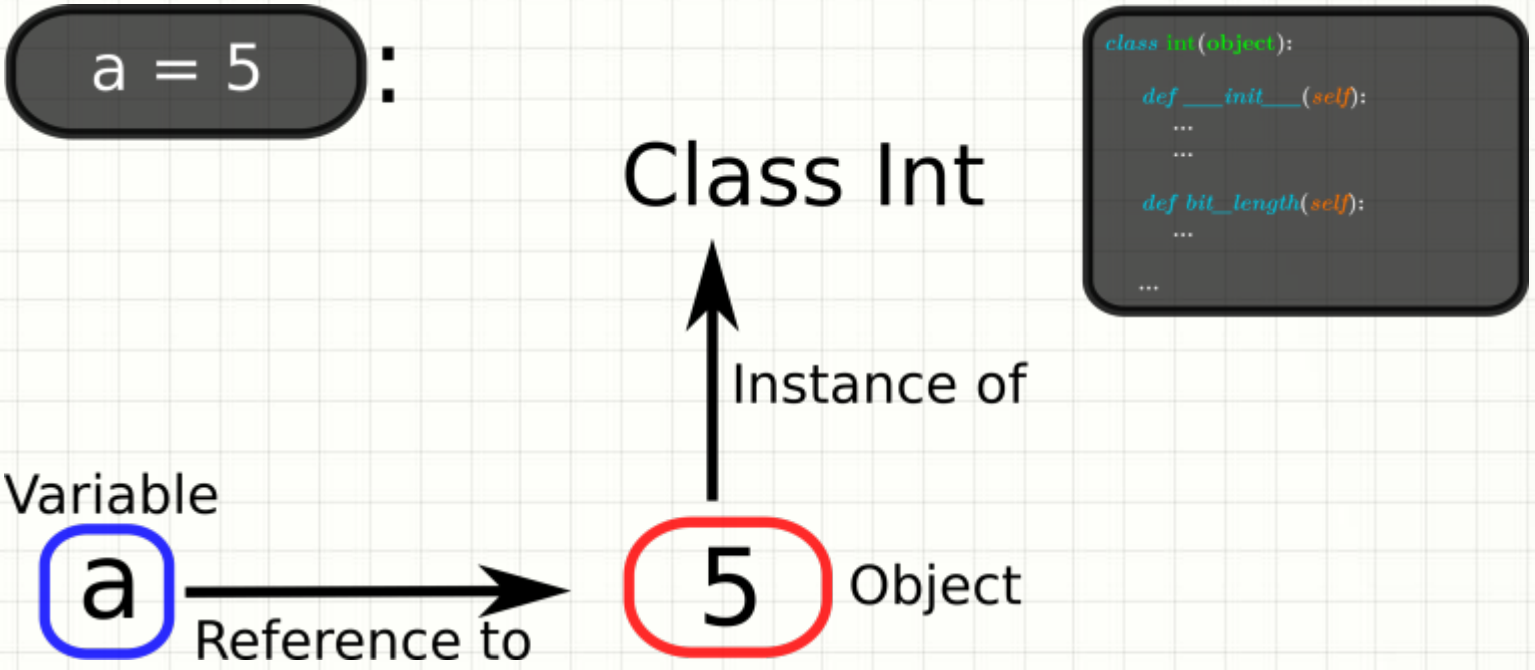
- Code highlighting + autocorrection
- Git integration
- Live code analysis
  (syntax, logic, PEP-8, …)
- Debug mode

Homework

Present and explain your solution

Classes

- **Define your own object**
- **Functions of a class are called methods**
- **You can create instances of a class (object)**



Definition

- Definition of function with `class` statement
- Specify a name
- Define the __init__() method

```
class MyClassName:
      def __init__(self):
          self.value = 50

      def my_class_method(self):
          return self.value
```

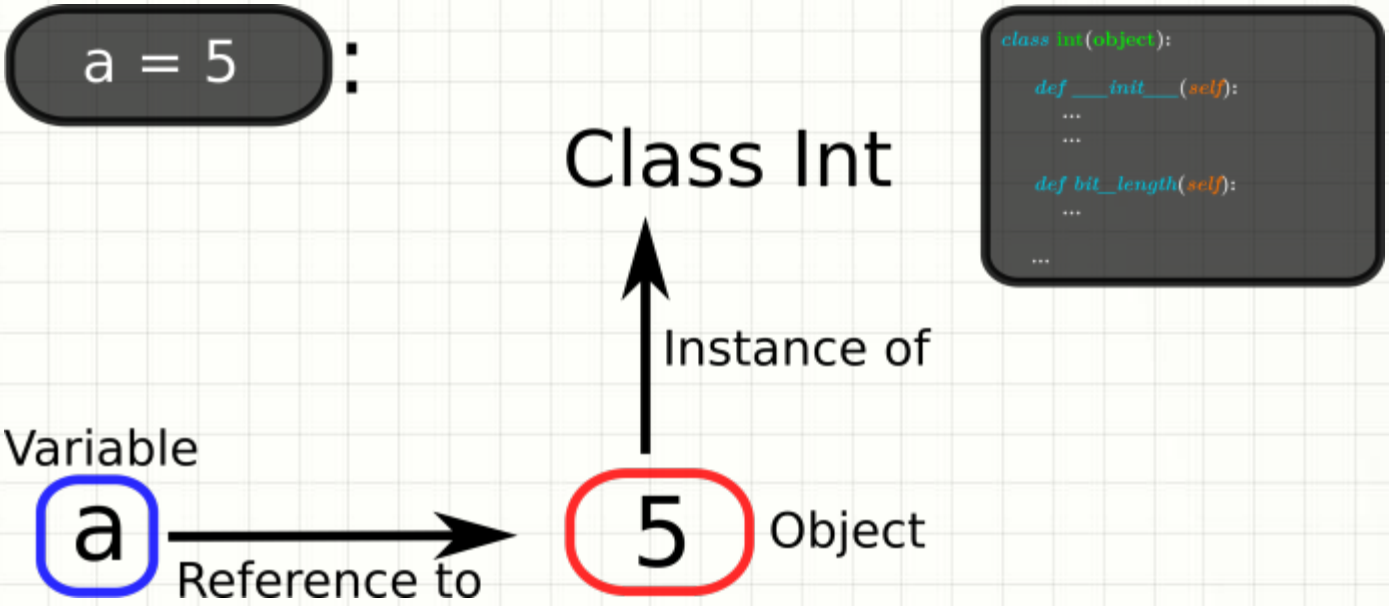
Example

```
class Student:
    def __init__(self, name, age):
        self.name = name
        self.age = age
        self.knowledge = 0

    def study(self):
        self.age += 1
        self.knowledge += 1

    def write_thesis(self):
        if self.knowledge >= 5:
            return ‘Good work!’
        else:
            return ‘Keep studying’
```

Create an object

```
jens = student('Jens Hahn', 20)
```

- **`jens`** is the **reference** to the **object** which is an **instance** of the **class `Student`**



Default values

- Set a value to the usual value

```
class ClassName:
      def __init__(self, value=5):
          self.value = value

      def class_method(self):
          ……
```

Doc strings

- Same as for functions
- Also use it for the class methods!!
- String is displayed when you call the **help()** function

```
class ClassName:
    """class description"""
    def __init__(self):
        """my constructor description"""
        self.value = 50

    def my_class_method(self):
        """my method description"""
```

HOMEWORK

Rock, paper, scissors class

- Take the class structure
- Fill the methods and add owns if you like
- Create an object of this class
- Is it mutable ot immutable?