Deep Learning

Übung WS 23/24

Judith Nester (nester@uni-koeln.de)

26-10-2023

Reference solutions

- Submission of the exercise until 23:59 pm on Tuesday
- Reference solution will be uploaded in the course of Wednesday
- Discussion of the reference solution on Thursday in case there are questions

Recap

- Version control, git
- Git vs. GitHub vs. GitLab
- Commits are central concept in git
  - Represents a set of changes in multiple files
  - Knows its predecessor
- Staging area
  - Area in which changes are assembled
- GUIs

Today's ToDo

- Python
  - Getting started
  - Syntax
  - Dynamic typing and data types
  - Standard library
- Exercise 2

Section 1

Python

Python

- ›New‹ programming language (1.0: 1991)
- General-purpose, high-level
- Popular for machine learning and natural language processing
- Latest version 3.12.0
  - Versions 3.8 and 3.9 are probably more bug-free

Python

- ›New‹ programming language (1.0: 1991)
- General-purpose, high-level
- Popular for machine learning and natural language processing
- Latest version 3.12.0
  - Versions 3.8 and 3.9 are probably more bug-free
- Differences to Java/C++
  - Dynamic typing
  - Indentation part of syntax
  - Interpreted language

Documentation

- `https://docs.python.org/3/`
  - Library reference: `https://docs.python.org/3/library/index.html`

Documentation

- `https://docs.python.org/3/`
  - Library reference: `https://docs.python.org/3/library/index.html`
- Tutorial
  - Al Sweigart: `https://automatetheboringstuff.com`
  - Also available as printed book and YouTube series

Installation and Getting Started

- Install Python from `https://www.python.org/downloads/`
- Install a code editor
  - Mac: TextMate `https://macromates.com`
  - Windows: Notepad++ `https://notepad-plus-plus.org`
- Get used to start python script via terminal

Installation and Getting Started

- Install Python from `https://www.python.org/downloads/`
- Install a code editor
  - Mac: TextMate `https://macromates.com`
  - Windows: Notepad++ `https://notepad-plus-plus.org`
- Get used to start python script via terminal
- Python can be used interactively in the interpreter

Installation and Getting Started

- Install Python from `https://www.python.org/downloads/`
- Install a code editor
  - Mac: TextMate `https://macromates.com`
  - Windows: Notepad++ `https://notepad-plus-plus.org`
- Get used to start python script via terminal
- Python can be used interactively in the interpreter

Integrated Development Environments (IDEs)

- There are Python IDEs:
  - PyCharm `https://www.jetbrains.com/pycharm/`
  - Thonny `https://thonny.org`
  - ...

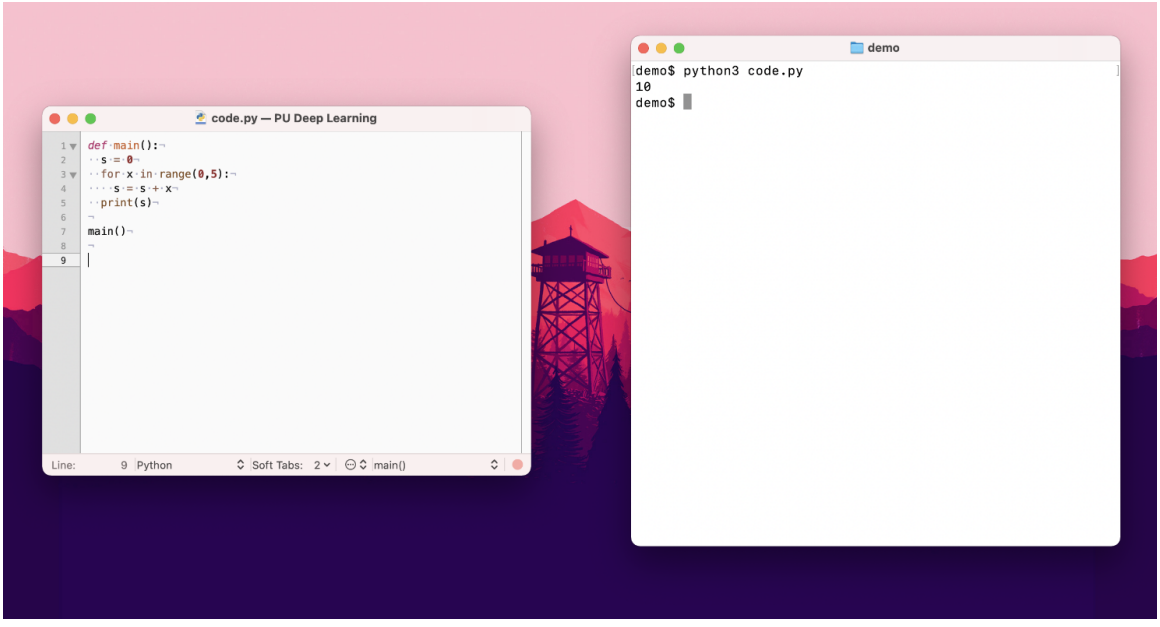
Installation and Getting Started

- Install Python from `https://www.python.org/downloads/`
- Install a code editor
  - Mac: TextMate `https://macromates.com`
  - Windows: Notepad++ `https://notepad-plus-plus.org`
- Get used to start python script via terminal
- Python can be used interactively in the interpreter

Integrated Development Environments (IDEs)

- There are Python IDEs:
  - PyCharm `https://www.jetbrains.com/pycharm/`
  - Thonny `https://thonny.org`
  - ...

⚠ Distracting and vendor-specific functionality


code.py — PU Deep Learning
def main():
s = 0
for x in range(0,5):
s = s + x
print(s)
main()
Line: 9 Python Soft Tabs: 2 main()
demo
demo$ python3 code.py
10
demo$

Subsection 1

Syntax

Code blocks

» Many programming languages use {} to mark code blocks

Listing: code.java

```java
public static void main(String[] args) {
    int s = 0;
    for (int i = 0; i < 5; i++) {
        s = s + i;
    }
    System.out.println(s);
}
```

Listing: code.py

```python
def main():
    s = 0
    for x in range(0,5):
        s = s + i
    print(s)
```

Indentation

- Python can use indentation by SPACE or TAB
- Spaces are preferred
- Mixing space or tab is not allowed (in Python 3)

Various Things

» # is used for comments

```
1 x = 15 # I am a comment
```

Various Things

» `#` is used for comments

```
1 x = 15 # I am a comment
```

» `def func():` declares a function

```
1 def func(x):
2     return x + 5
3 func(6) # returns 11
```

Data types

» Python is strongly, but dynamically typed

Data types

- Python is strongly, but dynamically typed
- Strong typing
  - No implicit conversion between data types

Data types

- Python is strongly, but dynamically typed
- Strong typing
  - No implicit conversion between data types
- Dynamic typing
  - Type is associated with values at runtime, not with variables at compile time

Data types

- Python is strongly, but dynamically typed
- Strong typing
  - No implicit conversion between data types
- Dynamic typing
  - Type is associated with values at runtime, not with variables at compile time

Listing: Python

```python
# This is ok:
x = "Hallo!" # a string value
x = 15       # an int value

# This is not ok (type error):
print("Das Ergebnis ist " + x)
```

Listing: Java

```java
// This does not compile:
String x = "Hallo!"
int x = 15
```

Dynamic typing

- Separation between variable name and object
- Variable name references an object
  - Several variable names can also reference the same object at the same time.
- Object knows its own data type
- Automatic memory management
  - Object that has no references to it gets deleted by garbage collector

Python Data Types

Name	`type(...)`	Casting	Values
Boolean	`<class 'bool'>`	`bool()`	{ `True, False` }
Integer	`<class 'int'>`	`int()`	natural numbers
Float	`<class 'float'>`	`float()`	decimal numbers
String	`<class 'str'>`	`str()`	character sequence
List	`<class 'list'>`	`list()`	arrays
Dictionaries	`<class 'dict'>`	`dict()`	key value store

Table: Builtin types in Python

`https://docs.python.org/3/library/stdtypes.html#built-in-types`

On Truth

» Some contexts require boolean values: `if`, `while`

- Any object can be used in this context:

```
1 x = 5
2 if x:
3     print("Hallo")
```

On Truth

» Some contexts require boolean values: `if`, `while`
  - Any object can be used in this context:

```
1 x = 5
2 if x:
3     print("Hallo")
```

» Non-boolean values can be used in these contexts, because they can be implicitly interpreted as bool
  - Constants defined to be false: `None`, `False`
  - Zero of any numeric type: `0`, `0.0`, ...
  - Empty sequences and collections: `''`, `()`, `[]`, `{}`, ...

» All other values are considered true

Standard Library

» Extensive standard library: `re`, `string`, `datetime`, `math`, `os.path`, `csv`, `os`, `json`, ...
- `https://docs.python.org/3/library/index.html`

Standard Library

» Extensive standard library: `re`, `string`, `datetime`, `math`, `os.path`, `csv`, `os`, `json`, ...
  - `https://docs.python.org/3/library/index.html`

» Using a package requires importing it

```
1 import math # import entire module
2 math.floor(7.5) # call a function from module, using module name
3
4 from math import ceil # import one function
5 ceil(6.9) # call a function without module name
```

Standard Library

» Extensive standard library: `re`, `string`, `datetime`, `math`, `os.path`, `csv`, `os`, `json`, ...
  - `https://docs.python.org/3/library/index.html`

» Using a package requires importing it

```
1 import math # import entire module
2 math.floor(7.5) # call a function from module, using module name
3
4 from math import ceil # import one function
5 ceil(6.9) # call a function without module name
```

» Imported modules and functions can be renamed using `... as ALIAS`

```
1 import math as m
2 m.floor(7.5)
3
4 from math import ceil as c
5 c(6.5)
```

Section 2

Exercise

Exercise 02

» https://github.com/IDH-Cologne-Deep-Learning-Uebung/exercise-02