ICTA_Python_Osnovni

Osnovne informacije

Trajanje: 10 tednov, 1x na tedn

Izpit: Zadnji teden
Predavatelja:

- **Gregor Balkovec** (gregor.balkovec@fe1.uni-lj.si)
- **Anže Glušič** (anze.glusic@fe1.uni-lj.si)

1 termin poteka 4 ure, 16:15 - 20:00. Vmes bomo imel nekaj pavzic, po potrebi.

Termini

Zaporedje	Datum	Dan	Tip
1.	19.4.2023	Sre	Predavanje 1
2.	25.4.2023	Tor	Predavanje 2
3.	4.5.2023	Čet	Predavanje 3
4.	10.5.2023	Sre	Predavanje 4
5.	17.5.2023	Sre	Predavanje 5
6.	24.5.2023	Sre	Predavanje 6
7.	31.5.2023	Sre	Predavanje 7
8.	7.6.2023	Sre	Predavanje 8
9.	14.6.2023	Sre	Predavanje 9
10.	21.6.2023	Sre	Izpit

Domače naloge

Vsak teden bomo imeli domače naloge iz tematik katere smo obravnavali na tečaju.

Namen domačih nalog je, da sami doma poizkušate rešiti problem, nato jih pošljete in jih mi pogledamo in vam povemo kje točno delate napake. Domače naloge so tudi nam indikator, kje se še kaj zatika in kaj bi bilo za ponoviti oziroma dodatno razložiti.

Naloge oddate preko email-a, kjer priložite datoteko z rešeno nalogo.

Zadeva email-a: [DN-Python] <Termin>_Ime_Priimek

> Primer: [DN-Python] Termin_01_Luka_Novak

Email pošljete na gregor.balkovec@fe1.uni-lj.si in anze.glusic@fe1.uni-lj.si.

Izpit

Izpit bo potekal zadnji termin. Časa boste imeli vse 4 ure, vendar povprečno participanti potrebujejo približno 2 uri.

Na izpitu vam je dovoljeno vse - internet, lastni zapiski, naši zapiski, itd. Le komunikacija (pogovarjanje, ...) ni odvoljena.

Za uspešno opravljen izpit boste potrebovali vsaj 80% vseh točk. Po koncu tečaja dobite certifikat.

Teorija in Zapiski predavanj

Zapiski predavanj bodo dostopni na GitHub-u, dodatna teorija pa je dostopna na potralu NetAcad.

GitHub

Notri boste našli naše zapiske in razlage snovi, navodila za vaje in rešitve, domača naloga itd.

NetAcad

Dodatna gradiva lahko najdete na NetAcad.

Vsi boste na emaile dobili povabilo k tečju. Predno lahko dostopate do gradiv morate izpolniti njihov "terms and services" in formo.

Setting up Programming Environment

Installing Python

Da se prične s programiranjem moramo imeti inštaliran **Python Interpreter** oziroma lahko uporabimo Online Python Interpreter.

Uporabljali bomo Python3.x verzijo.

Windows

Preverimo, če imamo že inštaliran Python:

- Odpremo CMD
- vpišemo `$ python --version`

    - Če piše `python is not recognized as an internal or external command` ... Potem nimamo inštaliranega Python internpreterja

Inštalacija:

- Python Webpage
- Najdemo za željeni operacijski sistem. Zdownloadamo najnovejšo različico 3.x verzije
    - Embedded zip file - to je, da ti extractaš v svojo datoteko in je to to
    - Executable - da ti inštalira in nrdi path itd..
- Obkljukamo **ADD Python to PATH!**

Problemi:

- Če je operacijski sistem že imel inštalirano neko drugo verzijo pythona, potem imamo sedaj lahko dve različni verziji.
- Da preverimo katera verzija python-a bo uporabljena vpišemo v CMD: `$ python --version` . Če ne dobimo željene verzije lahko vpišemo `$ python3 --version` oziroma `$ python3.10.8 --version`

Linux

Večina Linux sistemov pride z že inštaliranim python-om. Preverimo, če imamo dovolj visoko verzijo (Python3 +) - `$ python --version` .

V primeru, da je nimamo jo inštaliramo. Če nam zgornji ukaz še vedno ne vrača pravilne verzije pythona lahko poizkusimo ukaz `$ python3 --version` ali pa `$ python3.10.8 --version` .

MacOS

While current versions of macOS (previously known as “Mac OS X”) include a version of Python 2, it is likely out of date by a few months. Also, this tutorial series uses Python 3, so let’s get you upgraded to that.

The best way we found to install Python 3 on macOS is through the Homebrew package manager. This approach is also recommended by community guides like The Hitchhiker’s Guide to Python.

Step 1: Install Homebrew (Part 1) To get started, you first want to install Homebrew:

Open a browser and navigate to http://brew.sh/. After the page has finished loading, select the Homebrew bootstrap code under “Install Homebrew”. Then hit Cmd+C to copy it to the clipboard. Make sure you’ve captured the text of the complete command because otherwise the installation will fail. Now you need to open a Terminal.app window, paste the Homebrew bootstrap code, and then hit Enter. This will begin the Homebrew installation. If you’re doing this on a fresh install of macOS, you may get a pop up alert asking you to install Apple’s “command line developer tools”. You’ll need those to continue with the installation, so please confirm the dialog box by clicking on

"Install". At this point, you're likely waiting for the command line developer tools to finish installing, and that's going to take a few minutes. Time to grab a coffee or tea!

Step 2: Install Homebrew (Part 2) You can continue installing Homebrew and then Python after the command line developer tools installation is complete:

Confirm the "The software was installed" dialog from the developer tools installer. Back in the terminal, hit Enter to continue with the Homebrew installation. Homebrew asks you to enter your password so it can finalize the installation. Enter your user account password and hit Enter to continue. Depending on your internet connection, Homebrew will take a few minutes to download its required files. Once the installation is complete, you'll end up back at the command prompt in your terminal window. Whew! Now that the Homebrew package manager is set up, let's continue on with installing Python 3 on your system.

Step 3: Install Python Once Homebrew has finished installing, return to your terminal and run the following command:

$brewinstallpython3Note:Whenyoucopythiscommand,besureyoudon'tincludethe$ character at the beginning. That's just an indicator that this is a console command.

This will download and install the latest version of Python. After the Homebrew brew install command finishes, Python 3 should be installed on your system.

You can make sure everything went correctly by testing if Python can be accessed from the terminal:

Open the terminal by launching Terminal.app. Type pip3 and hit Enter. You should see the help text from Python's "Pip" package manager. If you get an error message running pip3, go through the Python install steps again to make sure you have a working Python installation. Assuming everything went well and you saw the output from Pip in your command prompt window...congratulations! You just installed Python on your system, and you're all set to continue with the next section in this tutorial.

Text-Editor

Visual Studio Code https://code.visualstudio.com/download

> Zdownloadamo Visual Studio Code View -> Terminals - klikneš, da ti pokaže terminal. Da zaženeš se moršs terminalom prestavt do kjer maš datoteko:
>
> - cd .. (da greš nazaj v mapi)
> - cd -ime- (da greš znotraj te mape)
> - ls da vidš kere vse mape so kle notr
>
> ko si znotraj mape k ma tvojo kodo: python -ime_datoteke.py-

Python extension

> Odpremo Visual Studio Code extensions in installiramo **Python**, izdanega s strani Microsoft-a. S tem pridobimo Linter, Code review, Debugger, ...

> Nato še izberemo python interpreter tako, da kliknemo `Ctrl+Shift+P` in napišemo `Python: Select interpreter` in izberemo naš python.

Uporaba linterja in code formaterja

`flake` je Python linter. To pomeni, da nam omogoča hiter pregled kode in izpostavi kje so morebitne napake (syntax errors, pomoč pri styling-u kode po priporočilih PEP8, itd.) Za inštalacijo flake v terminal vpišemo ukaz `pip install flake8`.

- Flake dokumentacija

`black` je Python code formater. Black nam samodejno odstrani prazne vrstice na koncu kode, samodejno naredi razmake med matematičnimi operacijami (po priporočilih PEP8), samodejno naredi razmake med funkcijami, itd. Načeloma nam formatira kodo po PEP8 priporočilih, kar pomeni, da vsi pišemo v istem stilu. To pa naredi kodo bolj berljivo za druge programerje. Za inštalacijo black v terminal vpišemo ukaz `pip install black`.

- Black dokumentacija

Za uporabo v VisualStudio Code:

Da nam VS Code avtomatično uporabi naši novo-inštalirani knjižnjici moramo dodati te nastavitve v naš VS Code.

Znotraj mape kjer se nahaja naš projekt (kjer kliknemo *open with Code*) ustvarimo mapo `.vscode` in znotraj te mape ustvarimo datoteko `settings.json`.

```
new_folder/
├─ .vscode/
│  ├─ settings.json
├─ Termin01/
├─ Termin02/
├─ main.py
```

Znotraj `settings.json` definiramo naše nastavitve:

```
{
    "python.linting.pylintEnabled": false,
    "python.linting.flake8Enabled": true,
    "python.linting.enabled": true,
    "python.linting.flake8Args": [
        "--max-line-length=130",
    ],
    "python.formatting.provider": "black",
    "editor.formatOnSave": true
}
```