

Opdrachten les 1, voorbereiding voor les 2

1. GIT intro

- a. Doorloop de “GIT intro” handout en beantwoord de vragen aan het einde.

2. CSD repository maken

- a. Maak op github of gitlab een repository aan met de naam *CSD2*. **Deze ga je de rest van het jaar gebruiken voor al je huiswerk.** Alle opdrachten voor CSD2 breng je in deze repository onder. In de GIT introductie staat de folder-structuur beschreven.
- b. Voeg een README.md file toe aan deze repository en beschrijf hierin beknopt de inhoud van de *CSD2* repository.
- c. Stuur een mail aan Marc en Ciska (ciska.vriezenga@hku.nl en daan.schrier@student.hku.nl) met daarin de link naar je repository. Deze vind je in github onder ‘Clone or download’. Stuur ons de SSH-variant.

Vooraf aan opdracht 3

- (Alleen voor mac gebruikers) Homebrew installeren, zie <https://brew.sh/>
- Git installeren en CSD repository aangemaakt (zie ook “2. repository” hierboven)
- Python 3 (niet 2.7) installeren.

Voor Mac: installeer Python vanuit Homebrew!

- windows & mac: <https://www.python.org/downloads/>
- Linux: <http://docs.python-guide.org/en/latest/starting/install3/linux/>

3. Python “Hello World”

- a. Maak in je lokale *CSD2* repository een map aan genaamd *python_basics*
- b. Maak hierbinnen een Python file genaamd “helloworld.py”
- c. Schrijf een python script dat de tekst “Hello World” op de commandline print
- d. Voeg dit script toe aan je lokale git repository
- e. Push je repository naar je online repository

4. Python “Hello World” uitbreiden

- a. Breid je Hello World script zo uit dat het aan de gebruiker zijn/haar naam vraagt en deze naam vervolgens op de command line gebruikt in een begroeting.

Voorbeeld:

De gebruiker vult “Peter” in, het python print vervolgens “Hello Peter” (of iets dergelijks) op de command line.

Tips:

- `username = input("...")`
- `print("...", username)`
- Voorzie de gebruiker van duidelijke instructies in de command line bij het opvragen van zijn/haar naam, zodat hij/zij weet wat er gevraagd wordt.

- b. Voeg het script weer toe aan je git repository, zowel lokaal als online.

5. Python sample player

- a. Maak binnen *python_basics* een Python file *playsound.py*.
- b. Schrijf een auditief python script. Bij het runnen van dit script wordt er een sample afgespeeld.

Tips:

- installeer pip (zoek online hoe)
- installeer de game library pygame (simpleaudio kan ook, voorbeelden van eerder jaar werken hiermee, maar die wordt niet ondersteund op windows).
`pip3 install pygame`
OP I.I.G. MAC WERKT DIT NIET MEER → volg de instructie die volgt, direct mooie oefening in het lezen en volgen van feedback bij het runnen console applicatie.
Zie ook: <https://www.pygame.org/>
- zoek naar eenvoudige voorbeelden van het gebruik van je gekozen library, deze zijn doorgaans te vinden op de site van de library zelf.

- c. Voeg je script toe aan je git repositories.

6. Python sample player

Breid je script uit met:

- aan de gebruiker vragen hoe vaak het geluid afgespeeld moet worden.

- een zelfgeschreven functie aanroepen waaraan je het door de gebruiker ingevoerde aantal keren doorgeeft. Deze functie speelt vervolgens meerdere keren (ingevoerde aantal keren) het geluid af.