Python OOP

1/ Ons klaarmaken voor Python

Kristof Michiels

Het doel van deze les?

- Het hoe en waarom van Python meegeven
- De tools oplijsten die we gaan gebruiken om onze programma's te schrijven

Inhoud

- Wat is Python? Waarom Python?
- Je programmeeromgeving opzetten
  - Python installeren
  - Onze development tools installeren/klaarmaken
- Je eerste programma in Python schrijven

Wat is Python? Waarom Python?

Wat is Python?

- Python is een programmeertaal die begin jaren 90 ontworpen en ontwikkeld werd door Guido van Rossum
- Het is open source software en de Python Software Foundation leidt actief de verdere ontwikkeling
- Python is ontwikkeld met het oog op leesbare code
  - Zo wordt structuur aangebracht door regelinspringing (i.p.v. bvb. accolades zoals in C-achtige talen)
  - Statements worden simpelweg beëindigd door het eind van de regel
- Vandaag is het één van de meest gebruikte talen: zie bvb Tiobe, PYPL

Waarom Python?

- Python is een geweldige taal om te leren. We gaan ons amuseren, en jij zal er 100% voor willen gaan 😃
- Je staat als ontwikkelaar op de schouders van reuzen en je kan beroep doen op gigantisch veel functionaliteit om te gebruiken in je programma's (Python Standard Library + van andere 3rd party ontwikkelaars). Dit is zeker het geval voor IoT, AI én CSC-ontwikkeling!
- Python is een efficiënte taal. Je programma's zullen meer doen in minder regels code, vergeleken met andere programmeertalen
- De syntax is eenvoudig en consistent en zal je helpen heldere en elegante code te schrijven. Deze code zal leesbaar zijn, eenvoudig te debuggen, gemakkelijk om uit te breiden
- Python heeft een zeer actieve gemeenschap van gebruikers die je met open armen zullen verwelkomen!

De Zen van Python

- De Zen of Python: een verborgen grap/easter egg met 20 richtlijnen voor het design van Python
- De tekst verschijnt wanneer je aan je code "import this" toevoegt
- "*Beautiful is better than ugly. Explicit is better than implicit. Simple is better than complex. Complex is better than complicated...*" Hier lees je een interpretatie
- Het vertelt je iets over de filosofie achter Python
- Veel Pythonista's (zo noemen de Python-adepten zichzelf) hangen deze aan, maar het is geen verplichting
- *Import* betekent hier: gebruik een module zodat ik ze in mijn code kan gebruiken
- De richtlijnen zijn nu voor jou misschien nog weinig sprekend, maar dit zal hopelijk anders zijn tegen het einde van de cursus 😃

Je programmeeromgeving opzetten

Python installeren

- De huidige versie van Python is 3.11
- Windows komt niet met Python aanwezig dus wellicht moet je het toch zelf installeren
- Je kan eerst nakijken of je toch een recente versie van Python hebt
- Je opent hiervoor een command venster door *command* in te typen in het Start menu. Je klikt dan op de Command Prompt app. Je krijgt dan een terminal venster
- Daarin typ je *python* + enter. Als je de Python prompt krijgt (>>>) dan is Python geïnstalleerd op je systeem (typ exit() + enter om deze prompt te verlaten)

Python installeren

- Sluit de Microsoft store indien deze zou openen. Beter is een officiële installer te downloaden via https://python.org
- Ga naar https://www.python.org/downloads/. Je vindt er een knop om de laatste versie van Python te downloaden
- Als dit gebeurd is voer je de installer uit. Belangrijk: kies voor de "Advanced installation"
- Kies op het vervolgscherm voor de optie "Add Python to PATH". Voor de rest volg je de standaard keuzes
- Je kan na afloop je installatie testen door de Python prompt op te roepen zoals in de vorige slide staat beschreven

Visual Studio Code

- We kennen Visual Studio Code uit het vak Web Technology
- Deze teksteditor is behalve voor het schrijven van HTML, CSS en JavaScript ook zeer geschikt voor programmeren in Python
- Mocht je deze tool nog niet hebben, dan kan je ze hier downloaden
- Een extensie met naam "Python" installeert zichzelf als je een .py-bestand aanmaakt. Je kan deze extensie uiteraard ook zelf installeren, samen met de "Pylance" extensie
- Je werkt met workspaces zoals in het vorige semester om je oefeningen te bundelen en te structureren
- Bovendien gaan we ook terminal sessies openen, om onze code te kunnen uitvoeren

Online Python schrijven met Replit

- Een zeer handig instrument om te prototypen in Python is https://replit.com
- Met deze tool kan je Python schrijven en oefenen in je browser
- Je mag Replit vergelijken met https://codepen.io voor html, css en JavaScript
- We noemen Replit een *online integrated development environment* (IDE)
- Ik moedig jullie aan om bij Replit een gratis account aan te maken en het platform te gebruiken bij het leren van Python

Je eerste programma in Python schrijven

Je eerste programma in Python schrijven

```
# Dit is je eerste programma in Python
print("Dit is mijn eerste Python programma")
```

- Een Python-programma wordt geschreven in een tekstbestand met de extensie .py
- Om het uit te voeren roep je de Python-interpreter aan, gevolgd door de naam van het .py-bestand
- Je doet dit in een terminal-venster, dat parallel kan geopend worden vanuit Visual Studio Code
- In de terminal typ je dan "python bestand.py" en je drukt op enter
- Je zal dan de output van het programma in de terminal kunnen zien

Hoe je code structureren in Python?

- Witruimte en inspringing leggen de structuur van blokken code vast ipv haakjes en puntkomma's
- Dit minimalisme is één van de meest karakteristieke aspecten van Python
- De visuele structuur van de code komt overeen met de reële structuur. Dat maakt de code beter leesbaar
- Python code ziet er relatief uniform uit, ook de code die je leest van andere programmeurs

```
vruchten = ["appel", "kiwi", "banaan", "mango"]
for vrucht in vruchten:
    print(vrucht)
    if vrucht == "banaan":
        break
```

Code structureren in Python

- Voor elk insprong-niveau gebruiken we 4 spaties
- Aanbevolen stijl - de zogenaamde PEP-8 die je vaak zult horen noemen: https://www.python.org/dev/peps/pep-0008/
- Gebruik geen tabs en meng geen tabs en spaties: indent-count wordt hierdoor verstoord
- Bij VSCode, settings: "Editor: Insert Spaces" aanvinken
- :-) Tabs versus Spaces (fragment uit de reeks *Silicon Valley*)

Output genereren met de print()-functie

- Om output te genereren in onze terminal maken we gebruik van de print()-functie
- Voor meer informatie over de wondere wereld van print(): https://realpython.com/python-print/

```
print("Welkom nieuwe Pythonista!")
```

Commentaar in je code

- Commentaar laat toe om zelfgeschreven notities toe te voegen aan je programma's
- In Python gebruiken we een hashtag ("#") aan het begin van een lijn commentaar om aan te geven dat we hiermee te maken hebben
- We geven hiermee aan de interpreter mee dat wat volgt niet hoeft geïnterpreteerd te worden
- Python heeft geen notatie voor multi-line commentaar. Je gebruikt een hashtag aan het begin van elke regel commentaar

```
# Een typisch "Hallo wereld"-voorbeeld
boodschap = "Welkom bij Python Development!"
print(boodschap)
```

Coderegels laten doorlopen met \

- Code is leesbaarder wanneer de regels kort wordt gehouden
- Aangeraden max aantal karakters door PEP-8 is 79: https://www.python.org/dev/peps/pep-0008/#maximum-line-length
- Als de regel toch langer wordt kan je gebruik maken van de backslash (= 'continuation character')
- Plaats een \ op het einde van een lijn en Python gedraagt zich alsof je je nog steeds op dezelfde regel bevindt

```
som = 1 + \
      2 \
      + 2
print(som) # 5
```

Python OOP - Ons klaarmaken voor Python

kristof.michiels01@ap.be