Vughtse Programmeerclub

Basiscursus programmeren met Python

Wat is Python

Vughtse Programmeerclub

Python is een krachtige, veelzijdige en gebruiksvriendelijke programmeertaal die in veel verschillende toepassingen wordt gebruikt, van webontwikkeling en data-analyse tot kunstmatige intelligentie en machine learning.

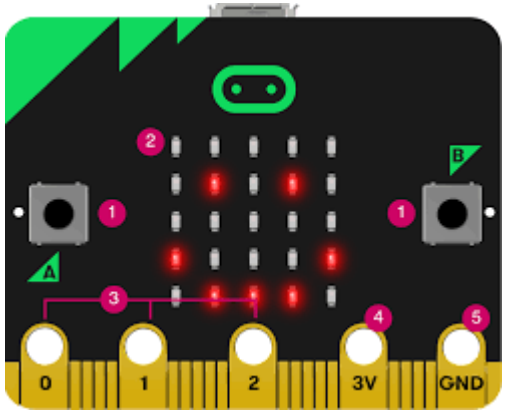
Dankzij de eenvoudige syntax is Python ideaal voor beginners, maar het biedt ook geavanceerde mogelijkheden voor ervaren programmeurs.

In deze training gaan we de basisprincipes van Python verkennen en leren hoe je met deze taal je eigen programma's kunt schrijven.

Wat hebben we nodig

Vughtse Programmeerclub

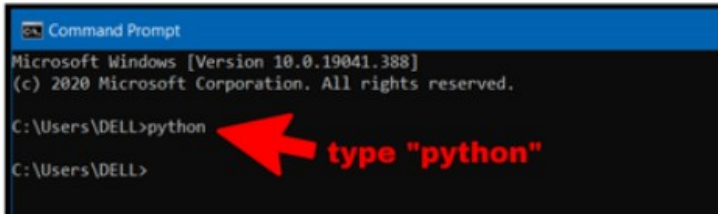
- Een laptop met Windows, macOS of Linux
- Python (https://www.python.org/)
- Thonny editor (https://thonny.org/)
- Een Micro:Bit (cursusavond 3)



Hello World!

Vughtse Programmeerclub

Windows: CMD.EXE

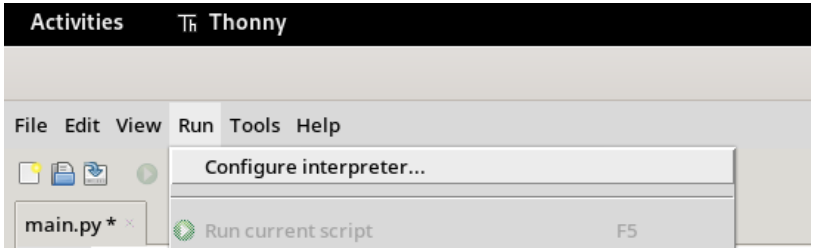


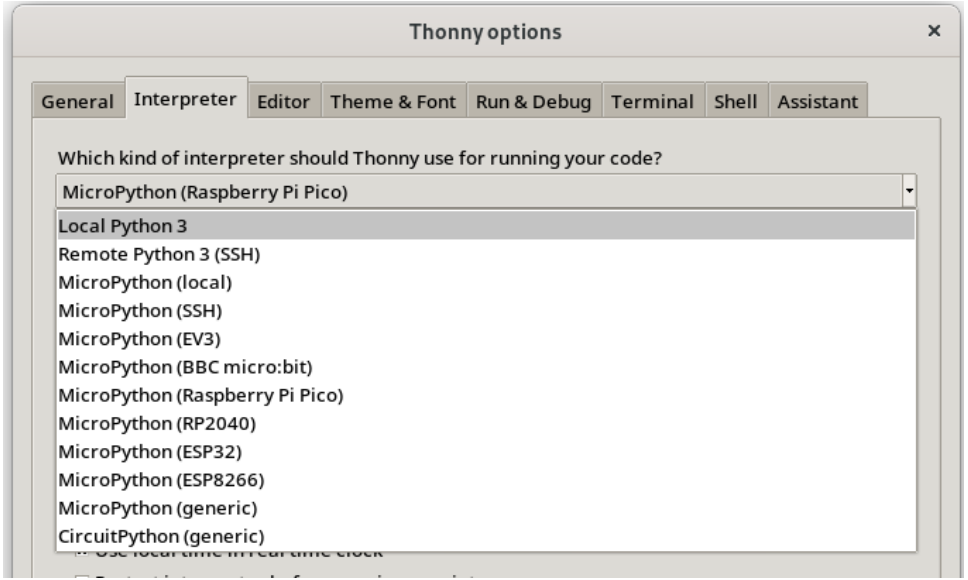
Linux/MacOS: Shell

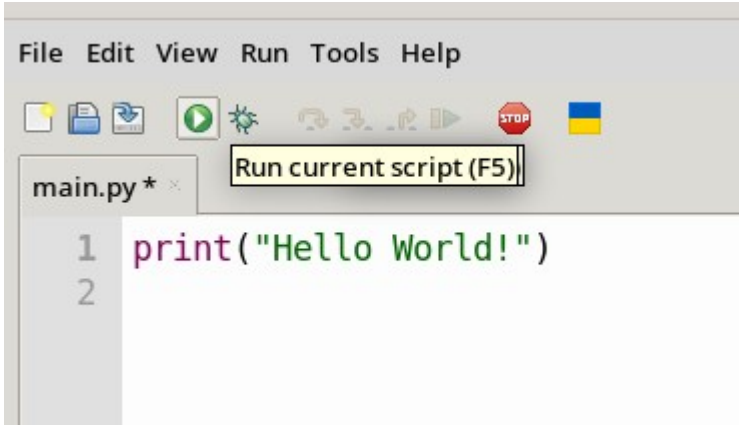
```
rudy@debian-rudy:~$ python
bash: python: command not found
rudy@debian-rudy:~$ python3
Python 3.11.2 (main, Nov 30 2024, 21:22:50) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
>>> print("Hello World!")
Hello World!
>>>
```

Thonny


Activities
Thonny
File Edit View Run Tools Help
Configure interpreter...
main.py *
Run current script F5


Thonny options
General
Interpreter
Editor
Theme & Font
Run & Debug
Terminal
Shell
Assistant
Which kind of interpreter should Thonny use for running your code?
MicroPython (Raspberry Pi Pico)
Local Python 3
Remote Python 3 (SSH)
MicroPython (local)
MicroPython (SSH)
MicroPython (EV3)
MicroPython (BBC micro:bit)
MicroPython (Raspberry Pi Pico)
MicroPython (RP2040)
MicroPython (ESP32)
MicroPython (ESP8266)
MicroPython (generic)
CircuitPython (generic)


File Edit View Run Tools Help
Run current script (F5)
main.py *
1 print("Hello World!")
2

Variabele

Vughtse Programmeerclub

- Integer (int): Geheel getal zonder decimalen.
- Float (float): Getal met decimalen.
- String (str): Tekst, omgeven door aanhalingstekens.
- Boolean (bool): Waarde True of False.
- List (list): Geordende verzameling van waarden.
- Tuple (tuple): Geordende, onveranderbare verzameling van waarden.
- Dictionary (dict): Verzameling van sleutel-waardeparen.
- Set (set): Ongeordende verzameling van unieke waarden.

Operatoren

Vughtse Programmeerclub

Toewijzingsoperator (=): Wijst een waarde toe aan een variabele.

Vergelijkingsoperatoren: Gebruikt om twee waarden te vergelijken.

- == (gelijk aan)
- != (niet gelijk aan)
- > (groter dan)
- < (kleiner dan)
- >= (groter dan of gelijk aan)
- <= (kleiner dan of gelijk aan)

Commando’s

Vughtse Programmeerclub

- print(): Stuurt informatie naar de console.
- input(): Verkrijgt invoer van de gebruiker.
- if / else: Voert code uit op basis van een voorwaarde.
- for / while: Herhaalt code totdat een voorwaarde waar is.
- def: Definieert een functie.
- return: Geeft een waarde terug vanuit een functie.
- import: Laadt een externe module (bibliotheek).
- try / except: Handelt fouten af tijdens de uitvoering van code.

Inspringen

Vughtse Programmeerclub

Een belangrijk kenmerk van Python is dat het inspringen van code onderdeel uitmaakt van de taal, dit in tegenstelling tot vele andere programmeertalen.Een eenvoudig voorbeeld om dit te verduidelijken:

```
for i in range(5):
    x = i * 10
    print(x)

print("Nieuwe opdracht")
```

Programmaverloop if/elif/else

Vughtse Programmeerclub

In deze paragraaf leer je dat je booleans (True en False) vaak zult gebruiken om het programmaverloop te beïnvloeden.

```
if expressie is waar:
    doe iets
else:
    doe iets anders
```

```
x = 20  # Wijs de waarde 20 toe aan de variabele x

if x > 20:
    print("X is groter dan 20!")
else:
    print("X is 20 of kleiner!")
```

```
leeftijd = 83
if leeftijd < 4:
    toegangsprijs = 0
elif leeftijd < 65:
    toegangsprijs = 40
else:
    toegangsprijs = 20

print(f"Je toegangsprijs is €{toegangsprijs}.")
```

While

Vughtse Programmeerclub

Naast if-elif-else is er nog een manier om het programmaverloop te bepalen op basis van expressies die een bool opleveren. Met while voer je iets uit zolang iets waar is.

```
# Eenvoudig voorbeeld
huidig_getal = 1
while huidig_getal <= 10:
    print(huidig_getal)
    huidig_getal += 1
```

```
# User input
instructie = "Vertel me iets, en ik herhaal het voor je. " \
         "Typ 'exit' om te stoppen. "
bericht = ""

while bericht != "exit":
    bericht = input(instructie)
    print(bericht, "\n")
```

Lists [ ]

Vughtse Programmeerclub

Een geordende, wijzigbare verzameling van elementen

- Bevat verschillende datatypes.
- Geordend, met index (start bij 0).
- Kan elementen toevoegen, verwijderen of wijzigen.

Bewerkingen:

- Toevoegen: mijn_lijst.append(5)
- Toegang: mijn_lijst[0]
- Verwijderen: mijn_lijst.remove(3)

```
# Lijsten maken
lege_lijst_1 = []
lege_lijst_2 = list()
lijst_met_items = [1, 2, 3, ]
```

Tuples ( )

Vughtse Programmeerclub

Een geordende, **onveranderbare** verzameling van elementen.

Kenmerken:

- Bevat verschillende datatypes.
- Geordend, met index (start bij 0).
- Kan niet worden gewijzigd na creatie (immutable).
-

```
lege_tuple_1 = ()
lege_tuple_2 = tuple()
tuple_met_items = ("a", "b", 1, 2, )
list_to_tuple = tuple([1, 2, 3, 4, ])
```

Bewerkingen:

- Toegang: mijn_tuple[0]
- Lengte: len(mijn_tuple)
- Kan niet worden gewijzigd (geen `append()`, `remove()`)

Dict (Dictionary) { }

Vughtse Programmeerclub

Een ongeordende verzameling van sleutel-waardeparen.

Kenmerken:

- Sleutels zijn uniek en worden gebruikt om waarden op te halen.
- Waarden kunnen van elk datatype zijn.
- Ongeordend (tot Python 3.7, daarna behouden ze de invoer volgorde).

```
nederlands_engels = {
    "programmeertaal": "programming language",
    "opleiding": "course",
    "leraar": "teacher",
}
```

Bewerkingen:

- Toegang: mijn_dict["sleutel"]
- Toevoegen: mijn_dict["nieuwe_sleutel"] = waarde
- Verwijderen: del mijn_dict["sleutel"]

Functions

Vughtse Programmeerclub

Een functie is een herbruikbare blok code dat een specifieke taak uitvoert.

Kenmerken:

- Gedefinieerd met het def-trefwoord.
- Kan parameters ontvangen en een waarde teruggeven met return.

- Bewerkingen:
- Definiëren: def mijn_functie():
- Aanroepen: mijn_functie()
- Teruggeven: return waarde

```
namen = ["Ali", "Marie", "John", ]


# Definieer de functie `groet`
def groet(naam):
    print(f"Hallo, {naam}!")


# Roep `groet` aan voor elke naam in `namen`
for naam in namen:
    groet(naam=naam)

# Hallo, Ali!
# Hallo, Marie!
# Hallo, John!
```

Uitzonderingen

Vughtse Programmeerclub

Afhandelen van uitzonderingen doe je met try.. except .., ofwel, je probeert iets en als dat niet lukt (er ontstaat een fout), dan doe je wat anders.

```python
try:
    x = 10 / 0  # Dit veroorzaakt een fout (delen door nul)
except ZeroDivisionError:
    print("Kan niet door nul delen!")
```

```python
try:
    x = 10 / 0  # Dit veroorzaakt een fout (delen door nul)
except Exception as e:
    print(f"Er is een fout opgetreden: {e}")
```