

Workshop: Introductie tot Python

Vincent Claes





Introductie

- Python Syntax
- Communiceren met Hardware
- Webserver programmatie in Python
- Game Programming in Python
- Fun experiments
- <https://github.com/pxltech/Python-Workshop/>

Python: Introductie

- Python is a clear and powerful object-oriented programming language, comparable to Perl, Ruby, Scheme, or Java.
- Some of Python's notable features:
 - Uses an elegant syntax, making the programs you write easier to read.
 - Is an easy-to-use language that makes it simple to get your program working. This makes Python ideal for prototype development and other ad-hoc programming tasks, without compromising maintainability.
 - Comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, reading and modifying files.
 - Python's interactive mode makes it easy to test short snippets of code. There's also a bundled development environment called IDLE.
 - Is easily extended by adding new modules implemented in a compiled language such as C or C++.
 - Can also be embedded into an application to provide a programmable interface.
 - Runs anywhere, including [Mac OS X](#), [Windows](#), [Linux](#), and [Unix](#), with unofficial builds also available for [Android](#) and iOS.
 - Is free software in two senses. It doesn't cost anything to download or use Python, or to include it in your application. Python can also be freely modified and re-distributed, because while the language is copyrighted it's available under [an open source license](#).

Python: Introductie

- Some programming-language features of Python are:
 - A variety of basic data types are available: numbers (floating point, complex, and unlimited-length long integers), strings (both ASCII and Unicode), lists, and dictionaries.
 - Python supports object-oriented programming with classes and multiple inheritance.
 - Code can be grouped into modules and packages.
 - The language supports raising and catching exceptions, resulting in cleaner error handling.
 - Data types are strongly and dynamically typed. Mixing incompatible types (e.g. attempting to add a string and a number) causes an exception to be raised, so errors are caught sooner.
 - Python contains advanced programming features such as generators and list comprehensions.
 - Python's automatic memory management frees you from having to manually allocate and free memory in your code.

Python: Introductie

- Data Types in Python
 - Numbers (int, float, complex,...)
 - Strings (str)
 - Lists
 - Dictionaries
 - Booleans
 - Tuples
 - Sets
- Uitprinten
- Interactief werken via: `Introductie_tot_Python_Syntax.ipynb`

Oefeningen in Python

- Via `input()` kan je invoer aan je gebruiker vragen in Python, schrijf een stukje code waar de gebruiker zijn naam invoert en print deze naam af in hoofdletters
- Schrijf een stukje code waar je de gebruiker achter een cijfer vraagt, en print alle cijfers uit tot dit getal
 - Via `int(getal)` kan je de invoer string converteren naar een getalwaarde (integer)
- Schrijf een functie waar je 2 getallen meegeeft en die het grootste getal terugstuurt

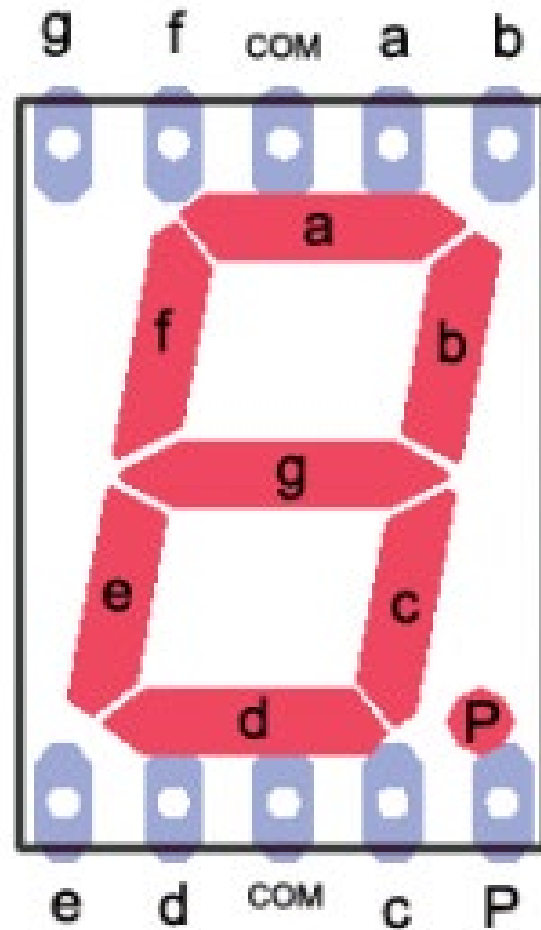
Bibliotheken in Python

- Veel gebruikt:
 - Pandas
 - <https://pandas.pydata.org/>
 - Bibliotheek voor data analyse
 - Numpy
 - <http://www.numpy.org/>
 - Bibliotheek voor “scientific computing”; lineaire algebra,...
 - Matplotlib
 - <https://matplotlib.org/>
 - Bibliotheek voor grafieken te maken
 - Sklearn
 - <https://scikit-learn.org>
 - Machine Learning library
 - Seaborn
 - <https://seaborn.pydata.org/>
 - Statistische data visualisatie bibliotheek
 - OpenCV
 - <https://opencv.org/>
 - Computer Vision bibliotheek

Python: Hardware Communicatie

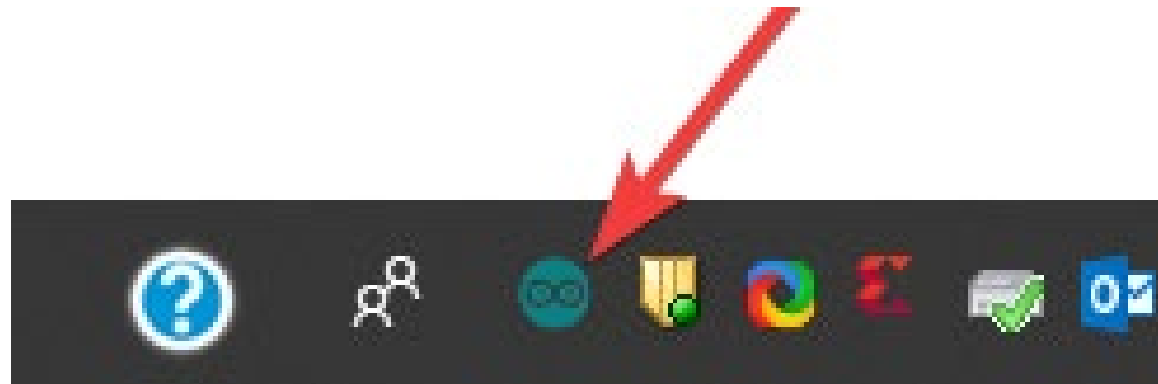
- Hardware: Arduino met 7-Segment Display
- Seriële Communicatie PC met Python \Leftrightarrow Arduino
- Aansturen van 7-segment display vanuit Python PC Applicatie
- Oefening Teller op 7-segment display
- Oefening Random Value op 7-segment display

Python: Hardware Communicatie



Arduino Software

- Online Arduino IDE: <https://create.arduino.cc>
- Install Arduino Plugin: <https://create.arduino.cc/getting-started/plugin>



Arduino Software 1

← → ↻ https://create.arduino.cc/editor/cteq_eu/94cf8a27-aa53-4885-a0a8-2fbf82d1adcf

⋮

> EDITOR

Sketchbook

Examples

Libraries

Monitor

Help

Preferences

NEW SKETCH

SEARCH SKETCHBOOK

ORDERING BY LAST MODIFIED

sketch_may4a

sketch_jan29a

sketch_may4a

✓ → Arduino/Genuino Uno at C... SHARE

sketch_may4a.ino ReadMe.adoc

```
1 // Application by Vincent Claes
2 // vincent@cteq.eu
3 // 14/11/2018
4
5 void setup() {
6   // Setting up Serial Communication with 115200 baudrate
7   Serial.begin(115200);
8 }
9 void loop() {
10  // Sending a string over serial communication
11  Serial.println("Hello From the Hardware world");
12  // 1 Second Delay
13  delay(1000);
14 }
```

Arduino Software 1

The screenshot displays the Arduino IDE web editor interface. The left sidebar contains navigation links: EDITOR, Sketchbook, Examples, Libraries, Monitor, Help, and Preferences. The main editor area shows a sketch named 'sketch_may4a' with a dropdown menu set to 'Arduino/Genuino Uno at C...'. A red arrow points to this dropdown. Below the dropdown, the sketch code is visible, starting with a comment: '// Application by Vincent Claes'. The code includes a setup function for serial communication and a loop function that prints 'Hello From the Hardware world' every 1000ms. At the bottom, a green status bar displays the message 'Success: Done uploading sketch_may4a', with a red arrow pointing to it. The bottom right corner of the image features the text 'Vincent Claes'.

https://create.arduino.cc/editor/cteq_eu/94cf8a27-aa53-4885-a0a8-2fbf82d1adcf

EDITOR

NEW SKETCH

Sketchbook

SEARCH SKETCHBOOK

ORDERING BY LAST MODIFIED

sketch_may4a

sketch_jan29a

sketch_may4a

ReadMe.adoc

```
1 // Application by Vincent Claes
2 // vincent@cteq.eu
3 // 14/11/2018
4
5 void setup() {
6   // Setting up Serial Communication with 115200 baudrate
7   Serial.begin(115200);
8 }
9 void loop() {
10  // Sending a string over serial communication
11  Serial.println("Hello From the Hardware world");
12  // 1 Second Delay
13  delay(1000);
14 }
```

Success: Done uploading sketch_may4a

Vincent Claes

Python: Hardware Communicatie

- Python: Installatie **PySerial**

A screenshot of the Anaconda Navigator application interface. On the left is a sidebar with navigation options: Home, Environments, Learning, and Community. The 'Environments' section is active, showing a list of environments: 'base (root)' and 'PXL_SS'. The 'PXL_SS' environment is selected. The main panel displays the 'Channels' tab, showing a list of installed and available packages. A search bar at the top right contains the text 'pyserial'. A red arrow points to the 'pyserial' entry in the search results. The entry shows a checkbox, the name 'pyserial', and a description 'Python serial port access library'.

Name	Description
<input type="checkbox"/> pyserial	Python serial port access library

Python: Hardware Communicatie

- Workshop_Serial1.ipynb

Reading Serial Port from Python

Developed by Vincent Claes

```
In [ ]: import serial
```

```
In [ ]: arduino = serial.Serial('COM5', 115200, timeout=.1)
```

```
In [ ]: while True:
        data = arduino.readline()[:-2]
        if data:
            print(data)
```

Python: Hardware Communicatie

- Workshop_Serial1.ipynb



Reading Serial Port from Python

Developed by Vincent Claes

```
In [1]: import serial
```

```
In [2]: arduino = serial.Serial('COM5', 115200, timeout=.1)
```

```
In [3]: while True:
        data = arduino.readline()[:-2]
        if data:
            print(data)
```

```
b'Hello From the Hardware world'
b'Hello From the Hardware world'
b'Hello From the Hardware world'
b'Hello From the Hardware world'
b'Hello From the Hardware world'
b'Hello From the Hardware world'
```

Arduino Software 2

```
// Application by Vincent Claes
// vincent@cteq.eu
// 14/11/2018

void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
    // Setting up Serial Communication with 115200 baudrate
    Serial.begin(115200);
}

// the loop function runs over and over again forever
void loop() {
    if(Serial.available()){
        // Parse Received value as an integer
        int state = Serial.parseInt();
        // Reading in Delimiter
        Serial.read();
        if(state==1)
            digitalWrite(LED_BUILTIN, HIGH);
        if(state==0)
            digitalWrite(LED_BUILTIN, LOW);
    }
}
```


Python: Hardware Communicatie

- Workshop_Serial2.ipynb

Sending Serial Data from Python

Developed by Vincent Claes

```
In [13]: import serial
import time
```

```
In [14]: arduino = serial.Serial('COM5', 115200, timeout=.1)
time.sleep(1)
```

```
In [19]: arduino.write(b'1')
```

```
Out[19]: 1
```

```
In [20]: arduino.write(b'0')
```

```
Out[20]: 1
```



Arduino Code 3

```
// Application by Vincent Claes
// vincent@cteq.eu
// 14/11/2018

#define a 2
#define b 3
#define c 5
#define d 6
#define e 7
#define f 8
#define g 9
#define p 4 // point
#define btn 10 // button on pcb

char Digits[10] = {B01111111,B0000110,B1011011,B1001111,B1100110,B1101101, B1111101,B0000111, B1111111,B1101111};

void Display_Value(char value);

void setup()
{
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  pinMode(p, OUTPUT);
  pinMode(btn, INPUT);
  Serial.begin(115200);
}

void loop()
{
  if(Serial.available()){
    // Parse Received value as an integer
    int state = Serial.parseInt();
    // Reading in Delimiter
    Serial.read();
    Display_Value(Digits[char(state)]);
  }
}
```

```
void Display_Value(char value)
{
  if(bitRead(value,0) == 1)
    digitalWrite(a,LOW);
  else
    digitalWrite(a,HIGH);
  if(bitRead(value,1) == 1)
    digitalWrite(b,LOW);
  else
    digitalWrite(b,HIGH);
  if(bitRead(value,2) == 1)
    digitalWrite(c,LOW);
  else
    digitalWrite(c,HIGH);
  if(bitRead(value,3) == 1)
    digitalWrite(d,LOW);
  else
    digitalWrite(d,HIGH);
  if(bitRead(value,4) == 1)
    digitalWrite(e,LOW);
  else
    digitalWrite(e,HIGH);
  if(bitRead(value,5) == 1)
    digitalWrite(f,LOW);
  else
    digitalWrite(f,HIGH);
  if(bitRead(value,6) == 1)
    digitalWrite(g,LOW);
  else
    digitalWrite(g,HIGH);
}
```

Python: Hardware Communicatie

- Workshop_7Segment.ipynb

Controlling a 7 Segment over Serial Connection from Python

Developed by Vincent Claes

```
In [ ]: 1 import serial
        2 import time
        3 import random
```

```
In [ ]: 1 arduino = serial.Serial('COM5', 115200, timeout=.1)
        2 time.sleep(1)
```

Sending Numbers in a Loop

```
In [ ]: 1 i = 0
        2 while i < 10:
        3     print(i)
        4     arduino.write(str(i).encode())
        5     time.sleep(1)
        6     i += 1
```

Using input value

```
In [ ]: 1 number = input()
        2 arduino.write(str(number).encode())
```

Random Value

```
In [ ]: 1 x = random.randint(0,9)
        2 arduino.write(str(x).encode())
        3 print(x)
```

```
In [ ]: 1 arduino.close()
```

Arduino Experiments

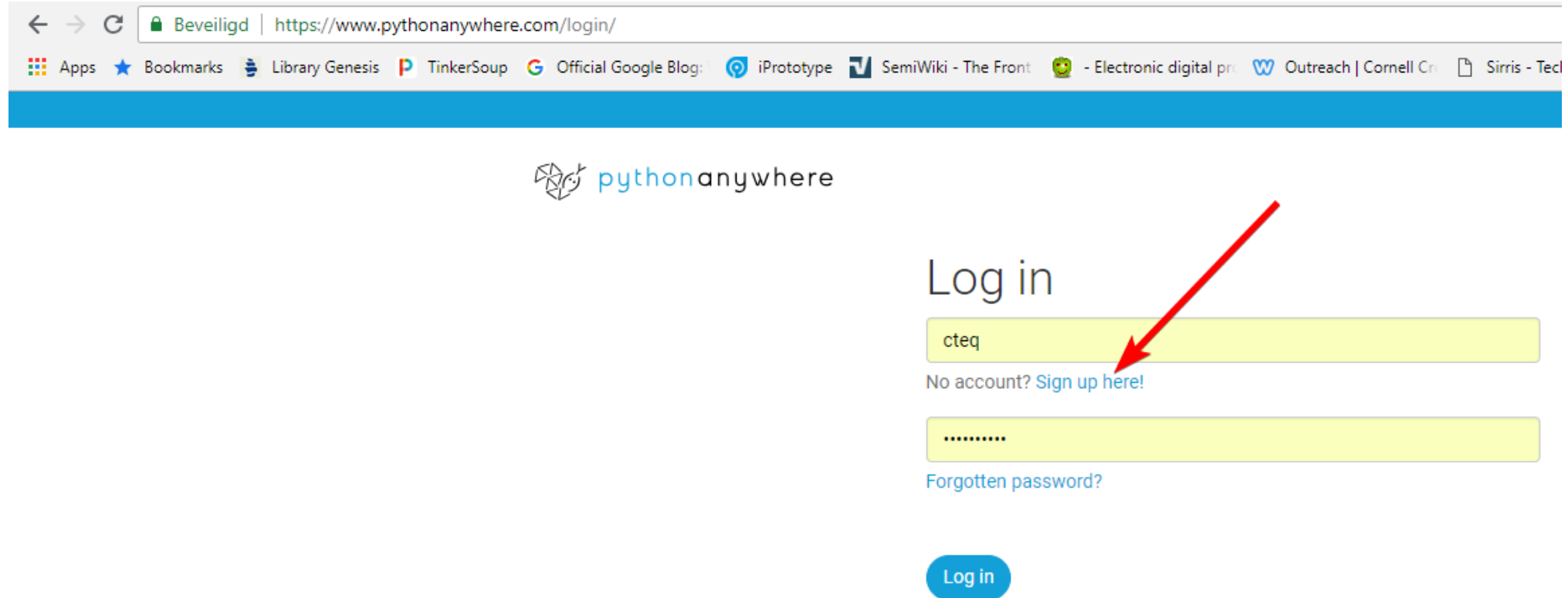
- Schrijf Python code waar je de 7-segment display aanstuurt maar waar je de decoder logica in Python schrijft; toon al de resultaten in een loop op de 7-segment display
- Schrijf Python code waar je de 7-segment display aanstuurt maar waar je de decoder logic in Python schrijft; vraag de gebruiker achter een getal.
- Schrijf Python code waar je de schakelaar inleest van de arduino.

Python: Webserver Programming

- Pythonanywhere
 - <https://www.pythonanywhere.com>
- Flask Python Library
 - <http://flask.pocoo.org>



Log in or Sign up @ pythonanywhere



A screenshot of a web browser showing the login page for pythonanywhere. The browser's address bar displays the URL <https://www.pythonanywhere.com/login/>. Below the browser window, the pythonanywhere logo is centered. To the right, the 'Log in' section contains two yellow input fields. The first field, for the username, contains the text 'cteq'. A red arrow points from the top right towards this field. Below the first field is a link that says 'No account? [Sign up here!](#)'. The second field, for the password, contains a series of dots. Below it is a link that says 'Forgotten password?'. At the bottom of the login section is a blue 'Log in' button.

pythonanywhere

Log in

cteq

No account? [Sign up here!](#)

.....

[Forgotten password?](#)

Log in

Open Web Apps



Dashboard [Consoles](#) [Files](#) [Web](#) [Tasks](#) [Databases](#)

Dashboard

Welcome back, [cteq](#)

CPU Usage: 1% used – 1.61s of 100s. Resets in 5 hours, 36 minutes [More Info](#)

[Upgrade Account](#)

File storage: 0% full – 128.0 KB of your 512.0 MB quota

Recent Consoles

+ 5 -

 [Python3.7 console 10173805](#)

[View all](#)

New console:

You can have up to 2 consoles. To get more, [upgrade your account!](#)

Recent Files

+ 5 -

[/home/cteq/mysite/flask_app.py](#)

[/home/cteq/.pythonstartup.py](#)

[/home/cteq/.bashrc](#)

[/home/cteq/README.txt](#)

[+ Open another file](#)

[Browse files](#)

Recent Notebooks

+ 5 -

Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!

All Web apps

You don't have any web apps.

[Open Web tab](#)



Add a new web app



+ Add a new web app



Create a new web app

Create new web app

Your web app's domain name

Your account doesn't support custom domain names, so your PythonAnywhere web app will live at `cteq.pythonanywhere.com`.

Want to change that? [Upgrade now!](#)

Otherwise, just click "Next" to continue.

Cancel

« Back


Next »

Flask Python Web Framework

Create new web app ✕

Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » Django
- » web2py
- » **Flask** 
- » Bottle
- » **Manual configuration** (including virtualenvs)


What other frameworks should we have here? Send us some feedback using the link at the top of the page!

Cancel« BackNext »

Python 3.7 (Flask 1.0.2)

Create new web app ✕

Select a Python version

- » Python 2.7 (Flask 1.0.2)
- » Python 3.4 (Flask 1.0.2)
- » Python 3.5 (Flask 1.0.2)
- » Python 3.6 (Flask 1.0.2)
- » Python 3.7 (Flask 1.0.2) 

Note: If you'd like to use a different version of Flask to the default version, you can use a virtualenv for your web app. There are [instructions here](#).

Cancel « Back Next »

Select path for python file – Flask app


Create new web app

Quickstart new Flask project

Enter a path for a Python file you wish to use to hold your Flask app. If this file already exists, its contents will be overwritten with the new app.

Path

/home/cteq/mysite/flask_app.py



Cancel

« Back

Next »

Overview Web Tab



[Dashboard](#) [Consoles](#) [Files](#) **Web** [Tasks](#) [Databases](#)

All done! Your web app is now set up. Details below. ×

cteq.pythonanywhere.com

[+ Add a new web app](#)

Configuration for cteq.pythonanywhere.com

Reload:

[↻ Reload cteq.pythonanywhere.com](#)

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Wednesday 12 December 2018**

[Run until 3 months from today](#)

[Paying users'](#) sites stay up forever without any need to log in to keep them running.

Traffic:

How busy is your site?

This month (previous month)	71	(0)
Today (yesterday)	3	(1)
Hour (previous hour)	1	(2)

Want some more data? [Paying accounts](#) get pretty charts ;)

Files Tab



Dashboard Consoles **Files** Web Tasks Databases

All done! Your web app is now set up. Details below. ×

cteq.pythonanywhere.com

[+ Add a new web app](#)

Configuration for cteq.pythonanywhere.com

Reload:

[↻ Reload cteq.pythonanywhere.com](#)

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Wednesday 12 December 2018**

[Run until 3 months from today](#)

[Paying users'](#) sites stay up forever without any need to log in to keep them running.

Open mysite directory



/home/ cteq

Directories

Enter new directory name

New directory

[.local/](#)



[.virtualenvs/](#)



[mysite/](#)



[Dashboard](#) [Consoles](#) **[Files](#)** [Web](#) [Tasks](#) [Databases](#)

[Open Bash console here](#)

0% full – 128.0 KB of your 512.0 MB quota

Files

Enter new file name, eg hello.py

New file

[.bashrc](#)



2018-09-06 12:31 559 bytes

[.gitconfig](#)



2018-09-06 12:31 266 bytes

[.profile](#)



2018-09-06 12:31 79 bytes

[.pythonstartup.py](#)



2018-09-06 12:31 77 bytes

[.vimrc](#)



2018-09-06 12:31 4.6 KB

[README.txt](#)



2018-09-06 12:31 232 bytes

Upload a file

100MiB maximum size



Open flask_app.py



/home/cteq/ mysite

Directories

[New directory](#)

[__pycache__/](#)

[Dashboard](#) [Consoles](#) **[Files](#)** [Web](#) [Tasks](#) [Databases](#)

[Open Bash console here](#) **0% full** – 128.0 KB of your 512.0 MB quota

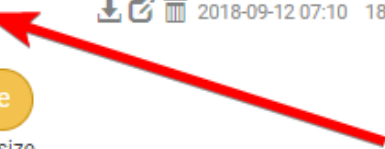
Files

[New file](#)

[flask_app.py](#) 2018-09-12 07:10 186 bytes

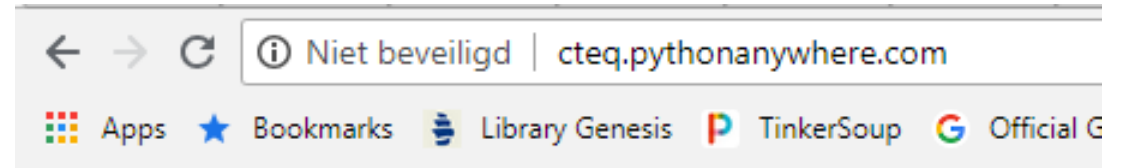
[Upload a file](#)

100MiB maximum size



<https://cteq.pythonanywhere.com>

```
1
2 # A very simple Flask Hello World app for you to get started with...
3
4 from flask import Flask
5
6 app = Flask(__name__)
7
8 @app.route('/')
9 def hello_world():
10     return 'Hello from Flask!'
11
```



Hello from Flask!

Example Flask App



/home/cteq/mysite/flask_app.py

Keyboard shortcuts:

Normal ▼

Share

Save

Save as...

>>> Run



```
1 from flask import Flask, url_for, request, json, Response, jsonify
2 from functools import wraps
3
4 app = Flask(__name__)
5
6 # test with curl -i https://xxxxx.herokuapp.com/hi
7 @app.route('/hi', methods = ['GET'])
8 def api_hi():
9     data = {
10         'hello': 'world',
11         'number': 456
12     }
13     js = json.dumps(data)
14
15     resp = Response(js, status=200, mimetype='application/json')
16     resp.headers['Link'] = 'http://www.cteq.eu'
17     return resp
18
```

Web Tab => reload



Dashboard Consoles Files **Web** Tasks Databases

cteq.pythonanywhere.com

+ Add a new web app

Configuration for cteq.pythonanywhere.com

Reload:

Reload cteq.pythonanywhere.com

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Wednesday 12 December 2018**

Run until 3 months from today

[Paying users'](#) sites stay up forever without any need to log in to keep them running.

Webserver2.py

- webserver2.py
- map templates aanmaken

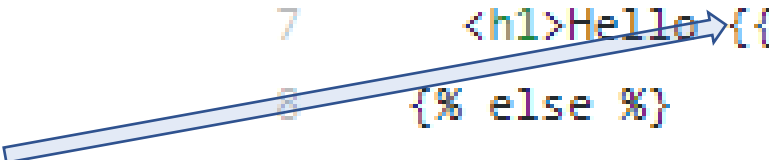
```
1  # Developed by Vincent Claes
2  # vincent@cteq.eu
3  # 14/11/2018
4
5  from flask import Flask, json, Response, render_template
6
7  app = Flask(__name__)
8
9  # test with curl -i https://xxxxxx.pythonanywhere.com/hi
10 @app.route('/hi', methods = ['GET'])
11 def api_hi():
12     data = {
13         'hello': 'world',
14         'number': 456
15     }
16     js = json.dumps(data)
17     resp = Response(js, status=200, mimetype='application/json')
18     resp.headers['Link']= 'http://www.cteq.eu'
19     return resp
20
21 # test with curl -i https://xxxxxx.pythonanywhere.com/hello/name
22 # test with curl -i https://xxxxxx.pythonanywhere.com/hello/
23
24 @app.route('/hello/')
25
26 @app.route('/hello/<name>')
27 def hello(name=None):
28     return render_template('hello.html', name=name)
```

Webserver2.py – hello.html

- webserver2.py
- map templates aanmaken
 - hello.html

```
24
25 @app.route('/hello/<name>')
26 def hello(name=None):
27     return render_template('hello.html', name=name)
```

```
1  <!-- Developed by Vincent Claes -->
2  <!-- vincent@cteq.eu -->
3  <!-- 14/11/2018 -->
4  <!doctype html>
5  <title>Introduction to Python Workshop</title>
6  {% if name %}
7      <h1>Hello{{ name }}!</h1>
8  {% else %}
9      <h1>Elektronica-ICT @Hogeschool PXL!</h1>
10 {% endif %}
```



Webserver2.py – hello.html



ⓘ Niet beveiligd | cteq.pythonanywhere.com/hello/



Elektronica-ICT @Hogeschool PXL!



ⓘ Niet beveiligd | cteq.pythonanywhere.com/hello/jos



Hello jos!

Oefeningen Webserver

- Schrijf code voor een webserver die een parameter verwacht “tekst”, print deze tekst in hoofdletters op je website
- Voeg CCS / HTML toe aan je website zodoende de site er mooier uit ziet
 - <https://www.w3schools.com/>
- Schrijf code voor een server die als parameter een e-mail adres verwacht en deze dan opsplitst in naam – voornaam en bedrijf, toon deze informatie op je website
- Bouw een eenvoudig webserver rekenmachine
- Maak een toepassing waar je een for loop gebruikt in de html template
 - Meer info :
<http://interactivepython.org/runestone/static/webfundamentalsITUBIL103E2015Fall/Frameworks/templates.html>

Python: Game Programming

- Gebruik van pygame library
 - <https://www.pygame.org>



Python: Game Programming



The screenshot shows the Anaconda Navigator application interface. On the left is a sidebar with navigation options: Home, Environments (selected), Learning, and Community. The main area displays a list of environments. The 'PXL_SS' environment is selected, and a context menu is open over it, showing options: 'Open Terminal' (highlighted with a red arrow), 'Open with Python', 'Open with IPython', and 'Open with Jupyter Notebook'. The background shows a table of installed packages with columns for Name, T, and Description.

Name	T	Description
Python asn.1 library		
Specifications for ce		
Easy, whitelist-base		
bokeh	✓	Statistical and nove

Python: Game Programming

https://anaconda.org/cogsci/pygame

ANACONDA CLOUD

CogSci / pygame

Python library based on SDL for making 2D games

Conda

License: LGPL

41448 total downloads

Last upload: 2 years and 6 months ago

Installers

conda install ?

linux-64	v1.9.2e0
win-32	v1.9.2e0
osx-64	v1.9.2e0
win-64	v1.9.2e0

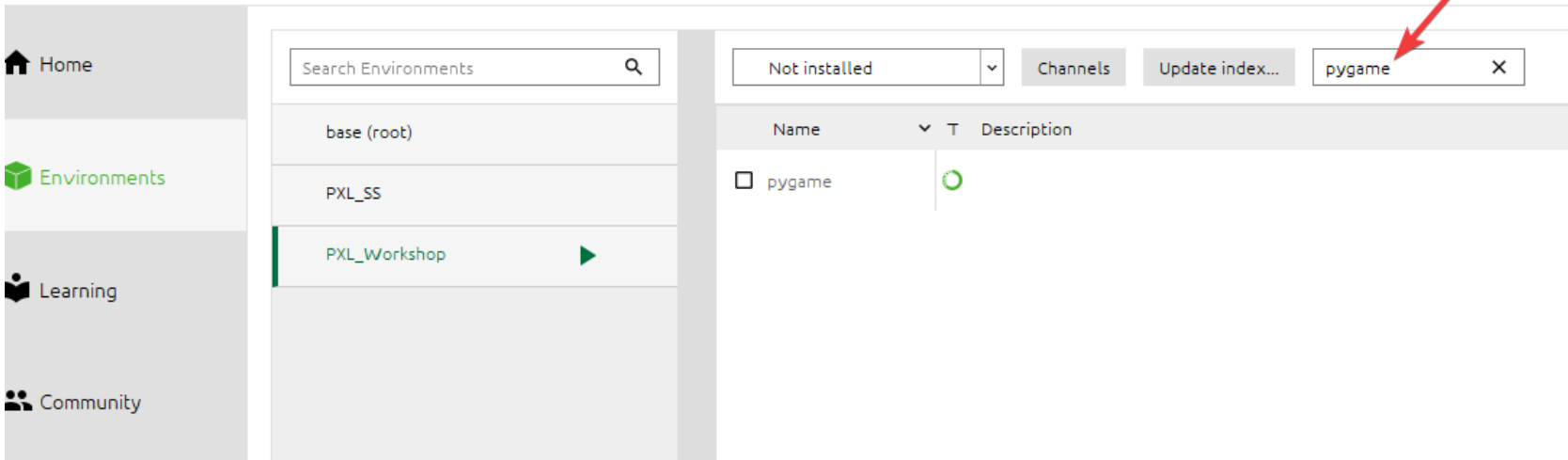
To install this package with conda run:

```
conda install -c cogsci pygame
```

C:\windows\system32\cmd.exe

```
(PXL_SS) C:\Users\20002890>conda install -c cogsci pygame
```

Python: Game Programming: PyGame installatie – alternatieve methode



```
C:\windows\system32\cmd.exe - conda install -c cogsci pygame

(PXL_SS) C:\Users\20002890>conda install -c cogsci pygame
Solving environment: done

## Package Plan ##

  environment location: C:\Users\20002890\AppData\Local\Continuum\anaconda3\envs\PXL_SS
added / updated specs:
- pygame

The following packages will be downloaded:

package | build | size
-----|-----|-----
pygame-1.9.2a0 | py35_0 | 4.6 MB cogsci

The following NEW packages will be INSTALLED:

pygame: 1.9.2a0-py35_0 cogsci

Proceed ([y]/n)?
```

- `python3 -m pip install -U pygame --user`
- `python3 -m pygame.examples.aliens`

Python: Game Programming

```
import pygame
import pygame.examples.chimp
pygame.examples.chimp.main()
```



Monkey Fever



Pummel The Chimp, And Win \$\$\$



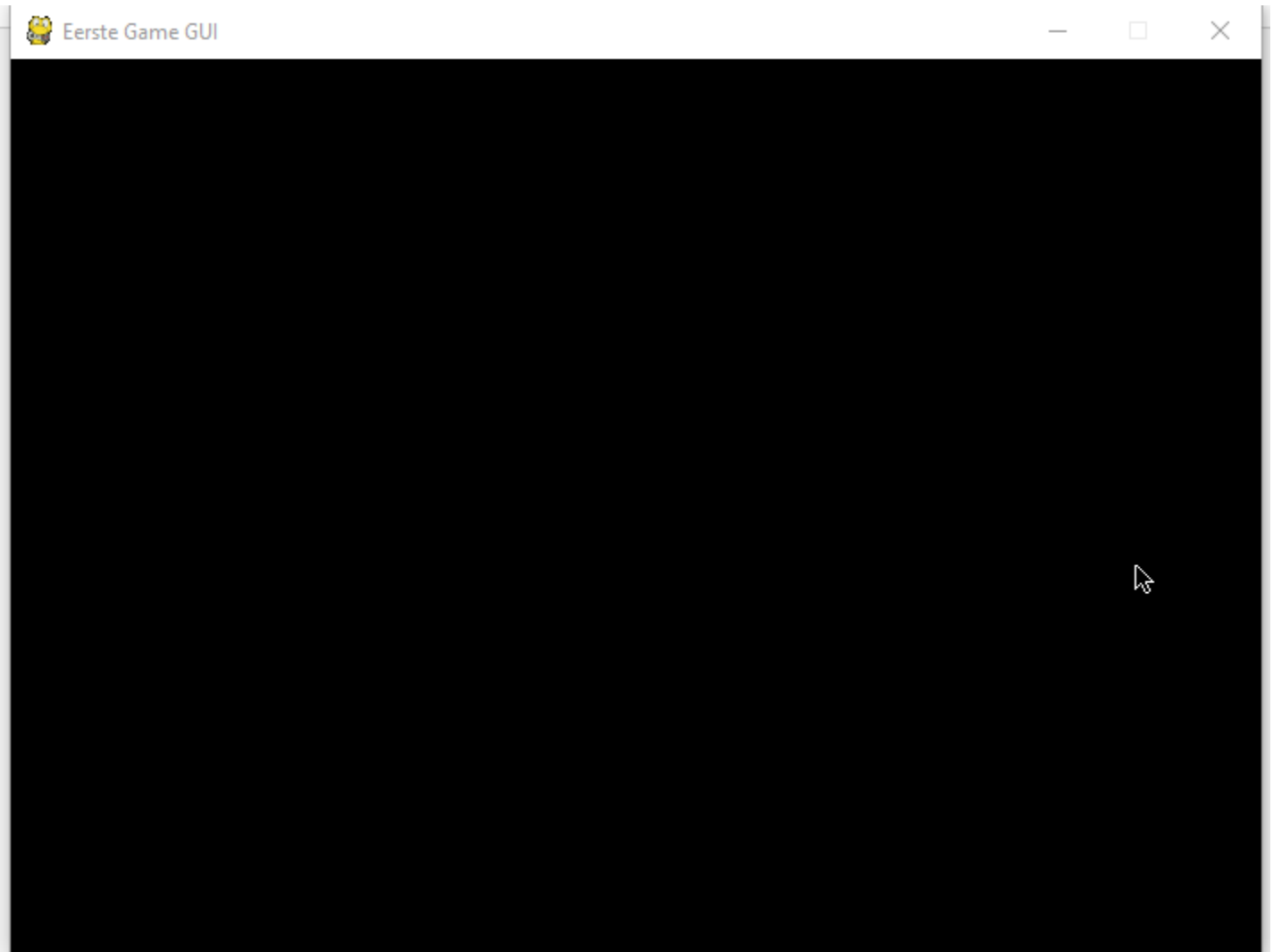
Python: Game Programming

```
# Pygame library importeren
import pygame
# Game Engine Initialiseren voor gebruik
pygame.init()

# Nieuw Scherm aanmaken
afmetingen = (700, 500)
scherm = pygame.display.set_mode(afmetingen)
# Titel van scherm instellen
pygame.display.set_caption("Eerste Game GUI")
# variabele gebruikt voor de game loop
Loop = True

while Loop:
    # --- Main Game Loop
    # Gebruikersinteractie
    for event in pygame.event.get():
        # Indien gebruiker close clicked
        if event.type == pygame.QUIT:
            # Programma beeindigen
            carryOn = False

# Game engine afsluiten
pygame.quit()
```



Python: Game Programming

```
# Pygame library importeren
import pygame
# Game Engine Initialiseren voor gebruik
pygame.init()

# Enkele kleuren definiëren
ZWART = (0,0,0)
WIT = (255,255,255)
GROEN = (0,255,0)
ROOD = (255,0,0)
BLAUW = (0,0,255)

# Nieuw Scherm aanmaken
afmetingen = (700, 500)
scherm = pygame.display.set_mode(afmetingen)
# Titel van scherm instellen
pygame.display.set_caption("Tweede Game GUI")
# variabele gebruikt voor de game loop
Loop = True

#Definiëren van een klok
klok = pygame.time.Clock()

while Loop:
    # --- Main Game Loop
    scherm.fill(BLAUW)
    # Gebruikersinteractie
    for event in pygame.event.get():
        # Indien gebruiker close clicked
        if event.type == pygame.QUIT:
            # Programma beeindigen
            carryOn = False

    # -- Scherm refreshen
    pygame.display.flip()

    # -- Scherm refreshen dient te gebeuren 60 frames/seconde
    klok.tick(60)
# Game engine afsluiten
pygame.quit()
```



Python: Game Programming

```
# Pygame library importeren
import pygame
# Game Engine Initialiseren voor gebruik
pygame.init()

# Enkele kleuren definiëren
ZWART = (0,0,0)
WIT = (255,255,255)
GROEN = (0,255,0)
ROOD = (255,0,0)
BLAUW = (0,0,255)

# Nieuw Scherm aanmaken
afmetingen = (500, 500)
scherm = pygame.display.set_mode(afmetingen)
# Titel van scherm instellen
pygame.display.set_caption("Gebruik van een afbeelding")
# variabele gebruikt voor de game loop
Loop = True

#Definiëren van een klok
klok = pygame.time.Clock()

# Definiëren en inladen van een afbeelding
pxllogo = pygame.image.load("pxl.png")
pxllogo = pygame.transform.scale(pxllogo,afmetingen)

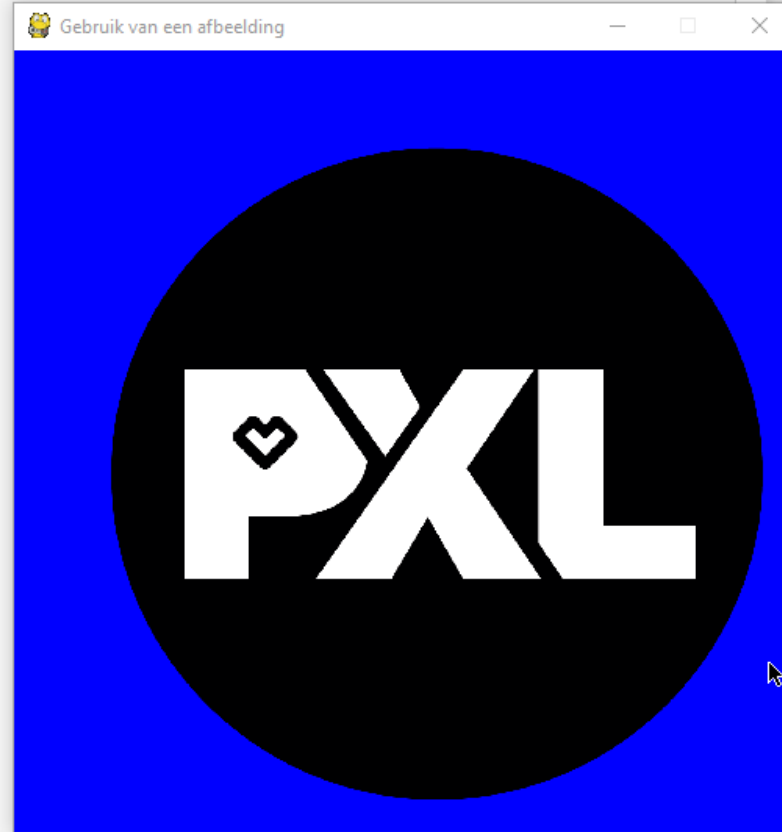
while Loop:
    # --- Main Game Loop
    scherm.fill(BLAUW)

    #Teken Logo op scherm
    scherm.blit(pxllogo, (20,20))

    # Gebruikersinteractie
    for event in pygame.event.get():
        # Indien gebruiker close clicked
        if event.type == pygame.QUIT:
            # Programma beeindigen
            Loop = False

    # -- Scherm refreshen
    pygame.display.flip()

    # -- Scherm refreshen dient te gebeuren 60 frames/seconde
    klok.tick(60)
# Game engine afsluiten
pygame.quit()
```



Python: Game Programming

```
# Pygame library importeren
import pygame
# Game Engine Initialiseren voor gebruik
pygame.init()

# Enkele kleuren definiëren
ZWART = (0,0,0)
WIT = (255,255,255)
GROEN = (0,255,0)
ROOD = (255,0,0)
BLAUW = (0,0,255)

# Nieuw Scherm aanmaken
afmetingen = (500, 500)
scherm = pygame.display.set_mode(afmetingen)
# Titel van scherm instellen
pygame.display.set_caption("Gebruik van een scaled afbeelding")
# variabele gebruikt voor de game Loop
Loop = True

#Definiëren van een klok
klok = pygame.time.Clock()

# Definiëren en inladen van een afbeelding
pxllogo = pygame.image.load("pxl.png")
logo_afmetingen = (50,50)
pxllogo = pygame.transform.scale(pxllogo,logo_afmetingen)

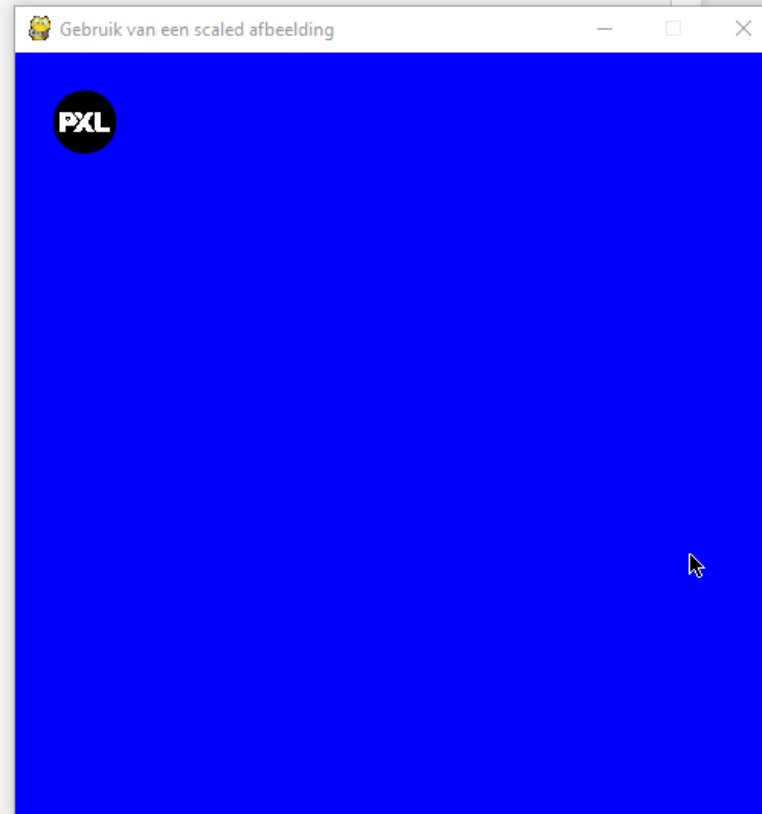
while Loop:
    # --- Main Game Loop
    scherm.fill(BLAUW)

    #Teken Logo op scherm
    scherm.blit(pxllogo, (20,20))

    # Gebruikersinteractie
    for event in pygame.event.get():
        # Indien gebruiker close clicked
        if event.type == pygame.QUIT:
            # Programma beëindigen
            Loop = False

    # -- Scherm refreshen
    pygame.display.flip()

    # -- Scherm refreshen dient te gebeuren 60 frames/seconde
    klok.tick(60)
# Game engine afsluiten
pygame.quit()
```



Python: Game Programming

- Beweging

```
# Pygame Library importeren
import pygame
# Game Engine Initialiseren voor gebruik
pygame.init()

# Enkele kleuren definiëren
ZWART = (0,0,0)
WIT = (255,255,255)
GROEN = (0,255,0)
ROOD = (255,0,0)
BLAUW = (0,0,255)

# Nieuw Scherm aanmaken
afmetingen = (500, 500)
scherm = pygame.display.set_mode(afmetingen)
# Titel van scherm instellen
pygame.display.set_caption("Moving PXL Logo")
# variabele gebruikt voor de game loop
Loop = True

#Definiëren van een klok
klok = pygame.time.Clock()

# Definiëren en inladen van een afbeelding
pxllogo = pygame.image.load("pxl.png")
logo_afmetingen = (50,50)
pxllogo = pygame.transform.scale(pxllogo,logo_afmetingen)

X_val = 0
Y_val = 0

while Loop:
    # --- Main Game Loop
    scherm.fill(BLAUW)

    #Teken Logo op scherm
    scherm.blit(pxllogo, (X_val,Y_val))

    # Gebruikersinteractie
    for event in pygame.event.get():
        # Indien gebruiker close clicked
        if event.type == pygame.QUIT:
            # Programma beëindigen
            Loop = False

    # -- Scherm refreshen
    pygame.display.flip()

    # -- Scherm refreshen dient te gebeuren 60 frames/seconde
    klok.tick(60)
    X_val = X_val + 1
    Y_val = Y_val + 1

    if (X_val > 500):
        X_val = 0
    if (Y_val > 500):
        Y_val = 0

# Game engine afsluiten
pygame.quit()
```

Python: Game Programming

- Keys gebruiken

```
# Ophalen van keys
keys = pygame.key.get_pressed()
if keys[pygame.K_LEFT]:
    X_val = X_val - 1
if keys[pygame.K_RIGHT]:
    X_val = X_val + 1
if keys[pygame.K_UP]:
    Y_val = Y_val - 1
if keys[pygame.K_DOWN]:
    Y_val = Y_val + 1
```

Python: Game Programming

- Geluid
 - <https://freesound.org/browse/>

```
import pygame

pygame.init()

scherm = pygame.display.set_mode((50,50))
klok = pygame.time.Clock()
pygame.mixer.music.load("bell.mp3")

GameLoop = False

while not GameLoop:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            GameLoop = True
        if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE:
            GameLoop = True

    klok.tick(60)
    # GameLoop = True
pygame.mixer.music.play(0)
pygame.time.delay(2000)
pygame.quit()
```

Python: Game Programming

- Tekst

```
import pygame

pygame.init()

scherm = pygame.display.set_mode((1200, 600))
klok = pygame.time.Clock()
GameLoop = False

font = pygame.font.SysFont("comicsansms", 72)

text = font.render("Hogeschool PXL Elektronica-ICT", True, (0, 128, 0))

while not GameLoop:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            GameLoop = True
        if event.type == pygame.KEYDOWN and event.key == pygame.K_ESCAPE:
            GameLoop = True

    scherm.fill((255, 255, 255))
    scherm.blit(text,
                (600 - text.get_width() // 2, 240 - text.get_height() // 2))

    pygame.display.flip()
    klok.tick(60)
pygame.quit()
```

Opdrachten

- Bouw een eigen game
 - Pong
 - Snake
 - Eigen improvisatie
- Informatie
 - <https://www.pygame.org/wiki/GettingStarted>
 - <https://www.pygame.org/docs/>
 - <https://realpython.com/python3-object-oriented-programming/>

Opdrachten

- Schrijf een applicatie die een geluid afspeelt indien iemand de “S” toets gebruikt
- Schrijf een applicatie die “PXL” schrijft indien iemand de “T” toets gebruikt, zorg ervoor dat de tekst terug verdwijnt indien iemand de “O” toets gebruikt
- Schrijf een applicatie waarmee je met de muis tekent op het scherm indien iemand eerst de “T” toets indrukt en dan met de muis over het gamescherm gaat.

Fun Experiments with Python

- Algorithmia API
 - <https://algorithmia.com/>
- Online API's
- Web Scraping
- OpenCV (Vision)
- Machine Learning Experiments

- More info: vincent[at]cteq.eu
- <https://www.linkedin.com/in/vincentclaes/>
- You can contact me for IoT, Embedded Systems (HW, SW, FPGA, ARM,...) and ML/AI projects (R and Python).