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

Vincent Claes

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

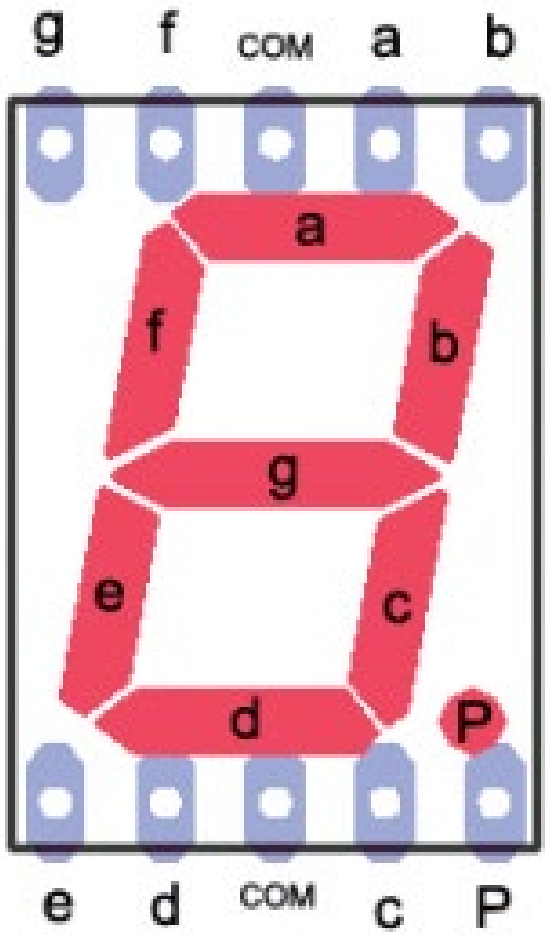
- Veel gebruikt:
  - Pandas
    - https://pandas.pydata.org/
    - Bibliotheek voor data analyse
  - Numpy
    - http://www.numpy.org/
    - Bibliotheek voor “scientific computing”; lineaire algebra,…
  - Matplotlib
    - https://matplotlib.org/
    - Bibliotheek voor grafieken te maken
  - Sklearn
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

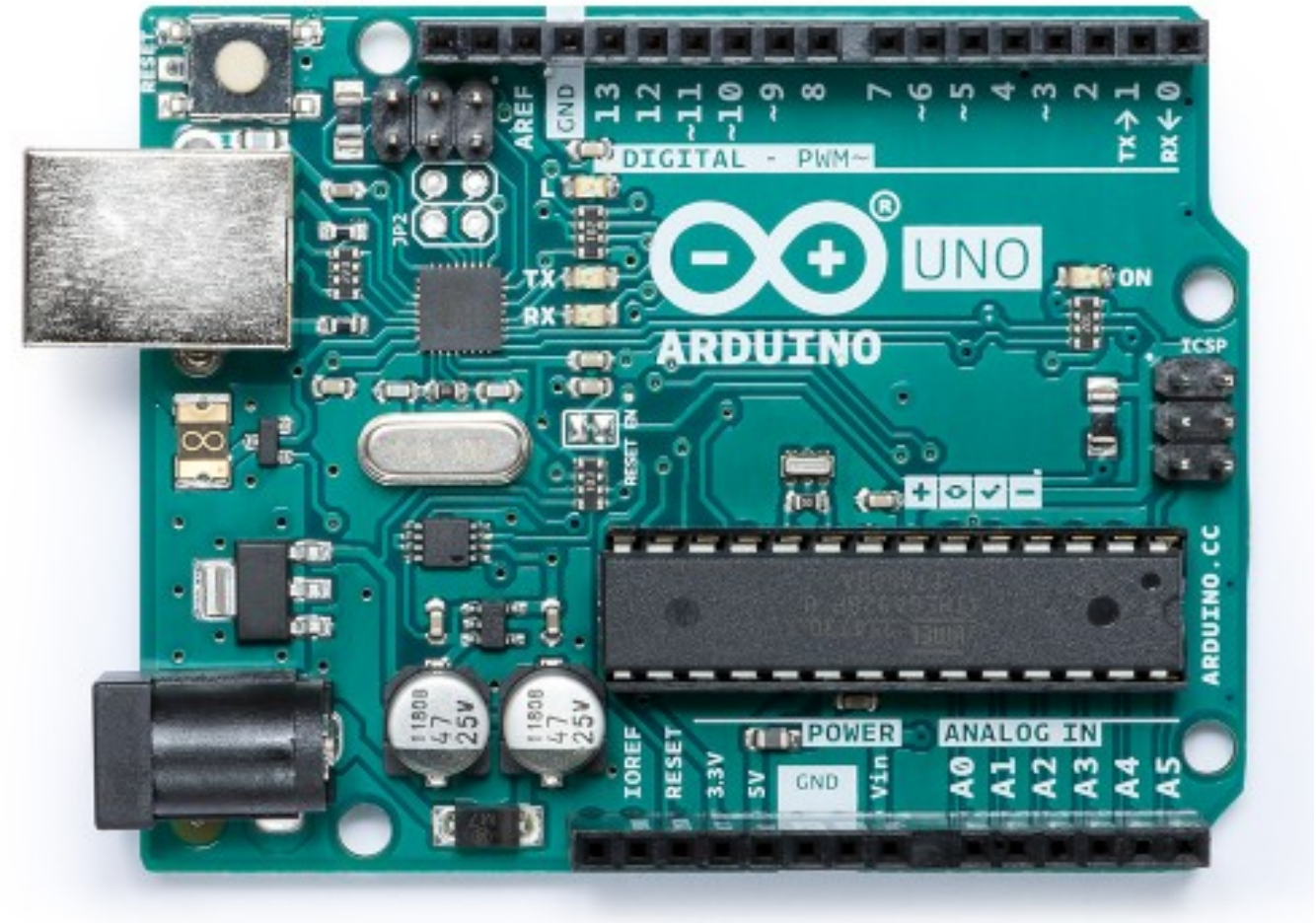
Python: Hardware Communicatie

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

Vincent Claes

Python: Hardware Communicatie

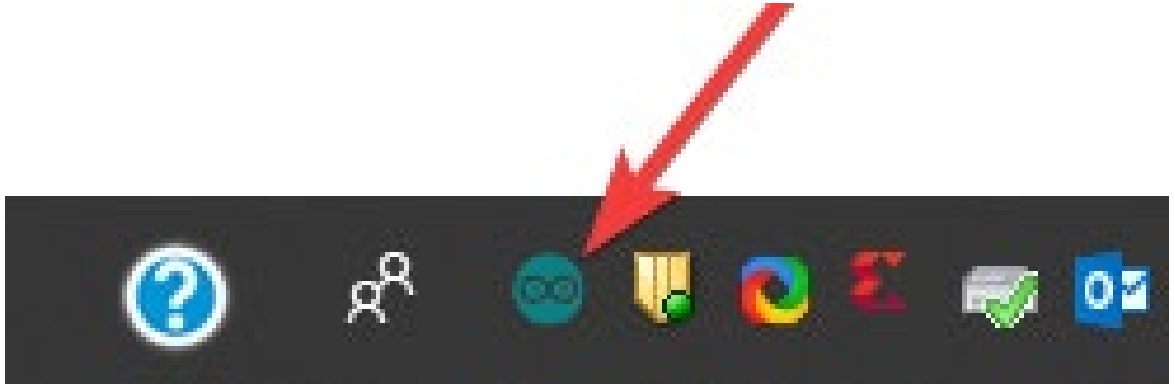




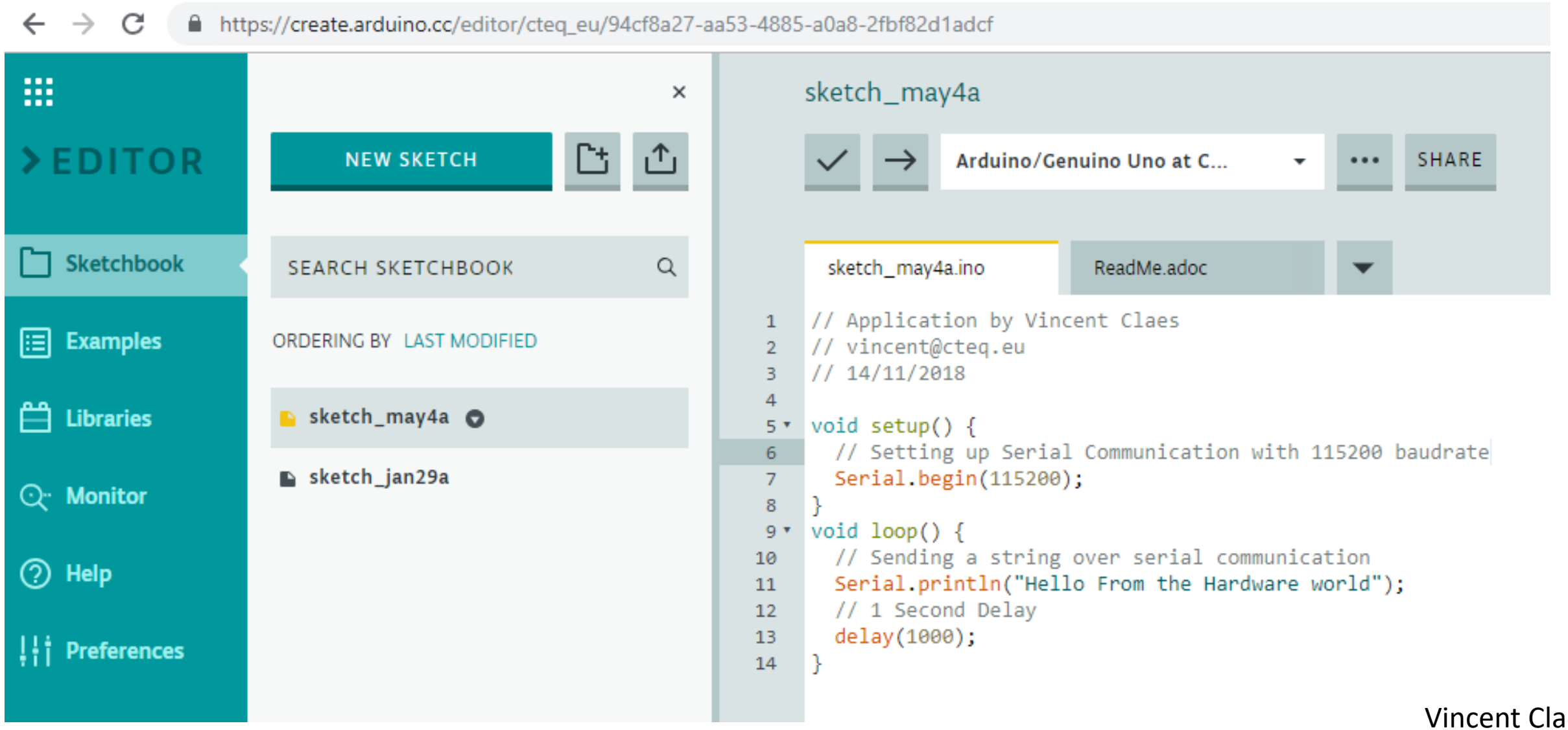
Arduino Software

- Online Arduino IDE: https://create.arduino.cc

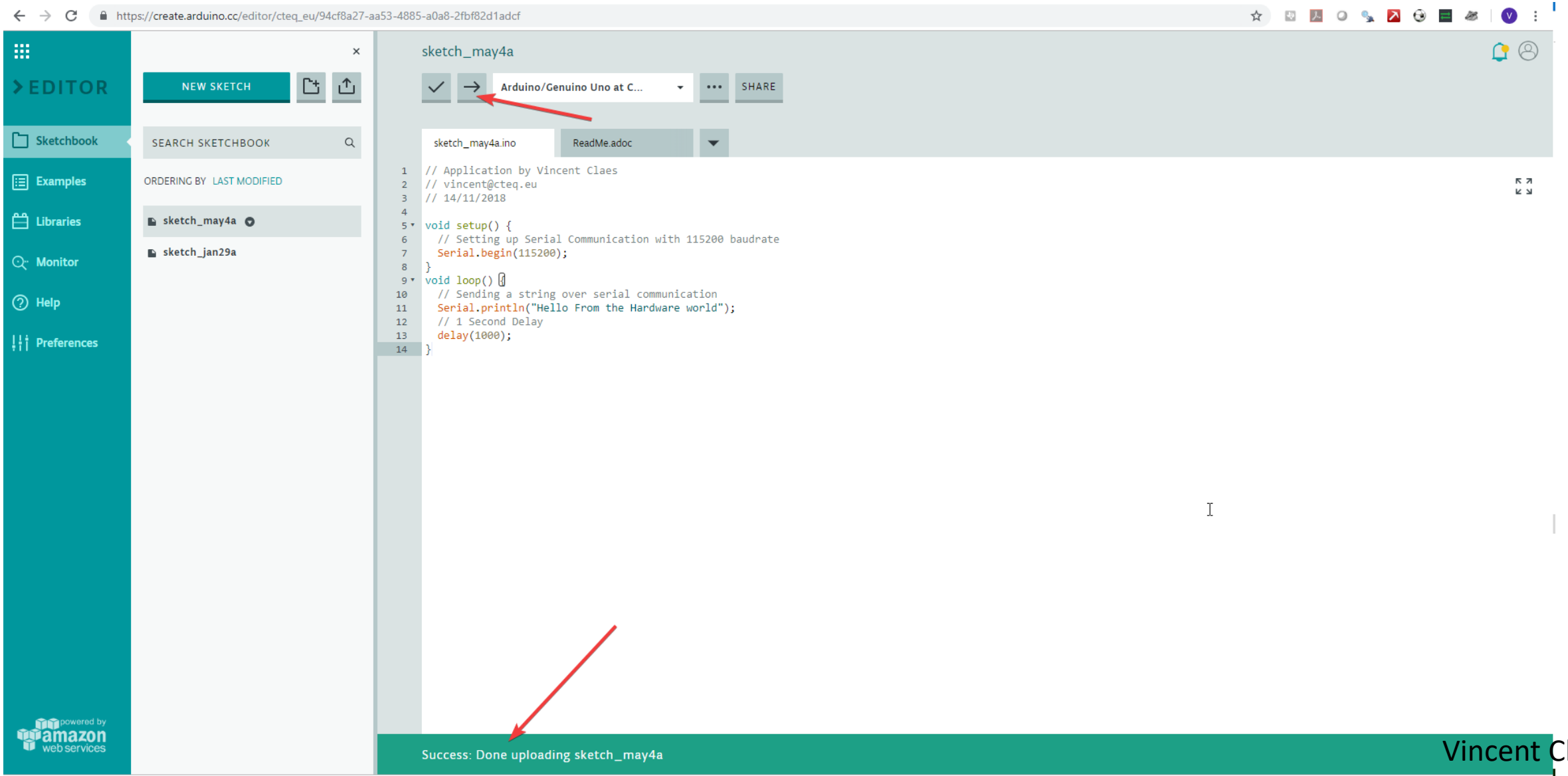
- Install Arduino Plugin: https://create.arduino.cc/getting-started/plugin



Arduino Software 1

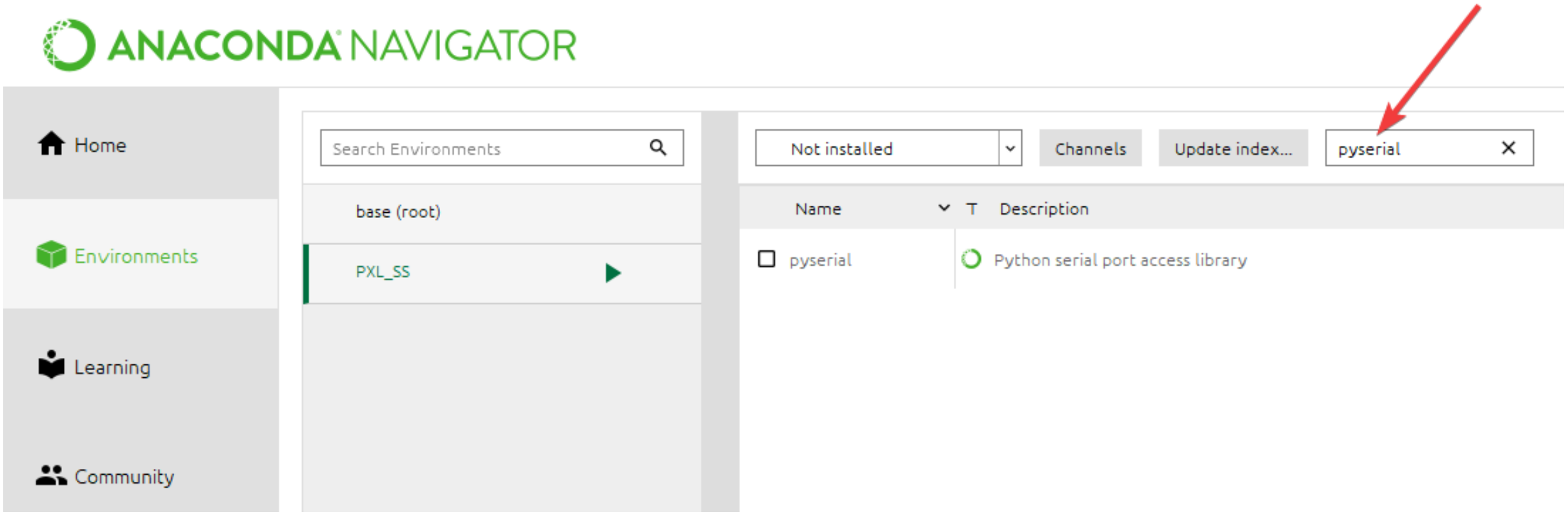


Arduino Software 1



Python: Hardware Communicatie

- Python: Installatie ***PySerial***



Python: Hardware Communicatie

- Workshop_Serial1.ipynb

Reading Serial Port from Python

Developed by Vincent Claes

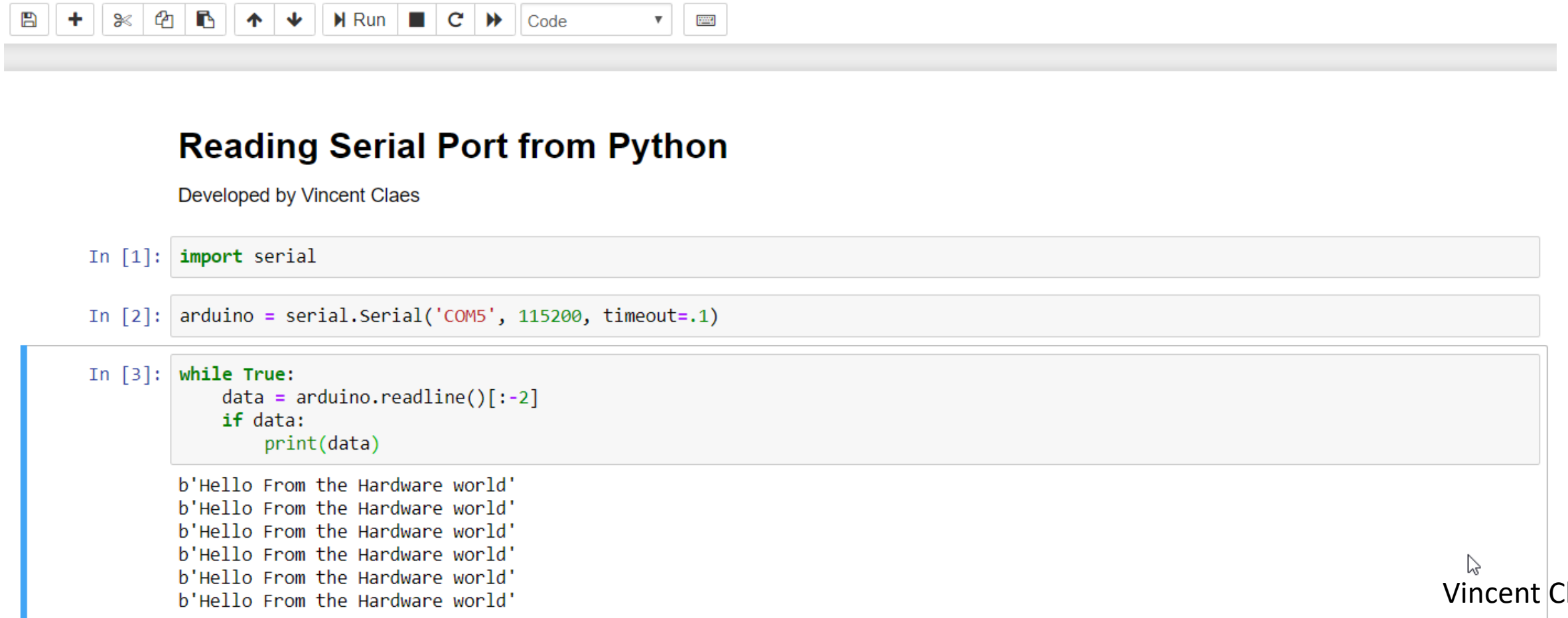
In [ ]:
```
import serial
```

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



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] = {B0111111,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 [ ]:
```
import serial
import time
import random
```

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

Sending Numbers in a Loop

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

Using input value

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

Random Value

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

In [ ]:
```
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.

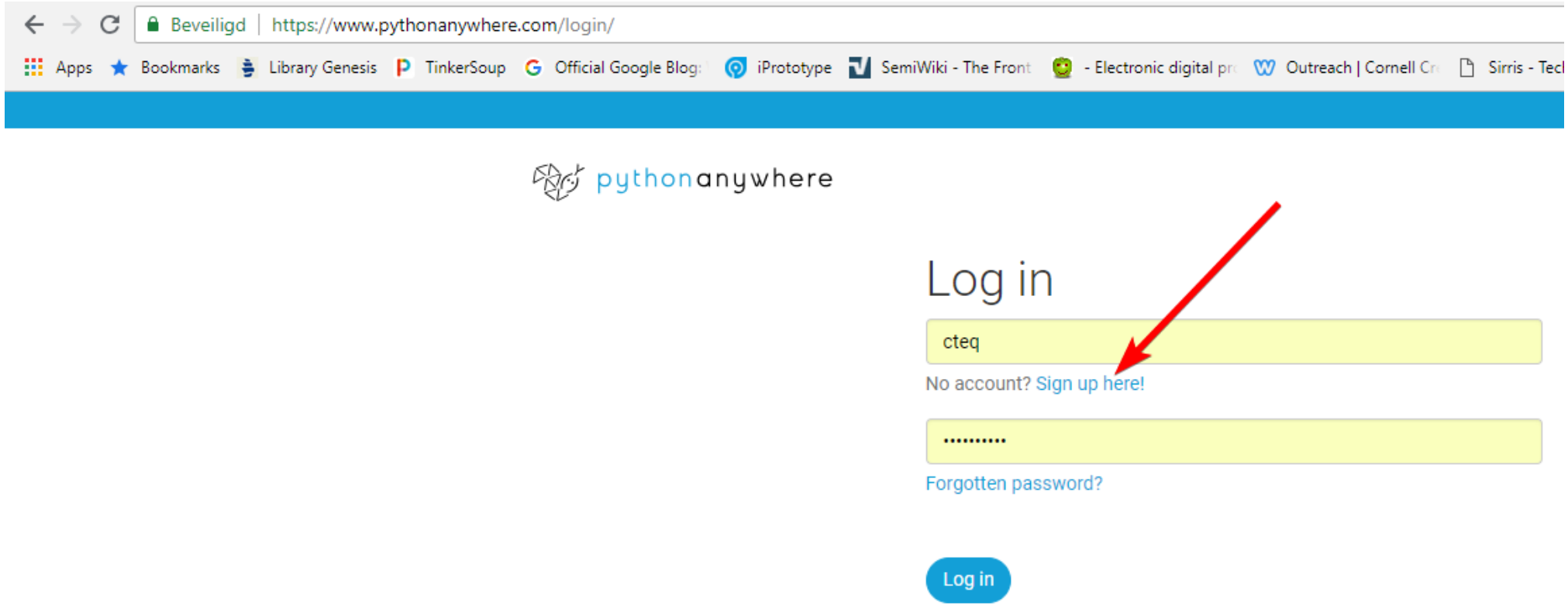
Vincent Claes

Python: Webserver Programming

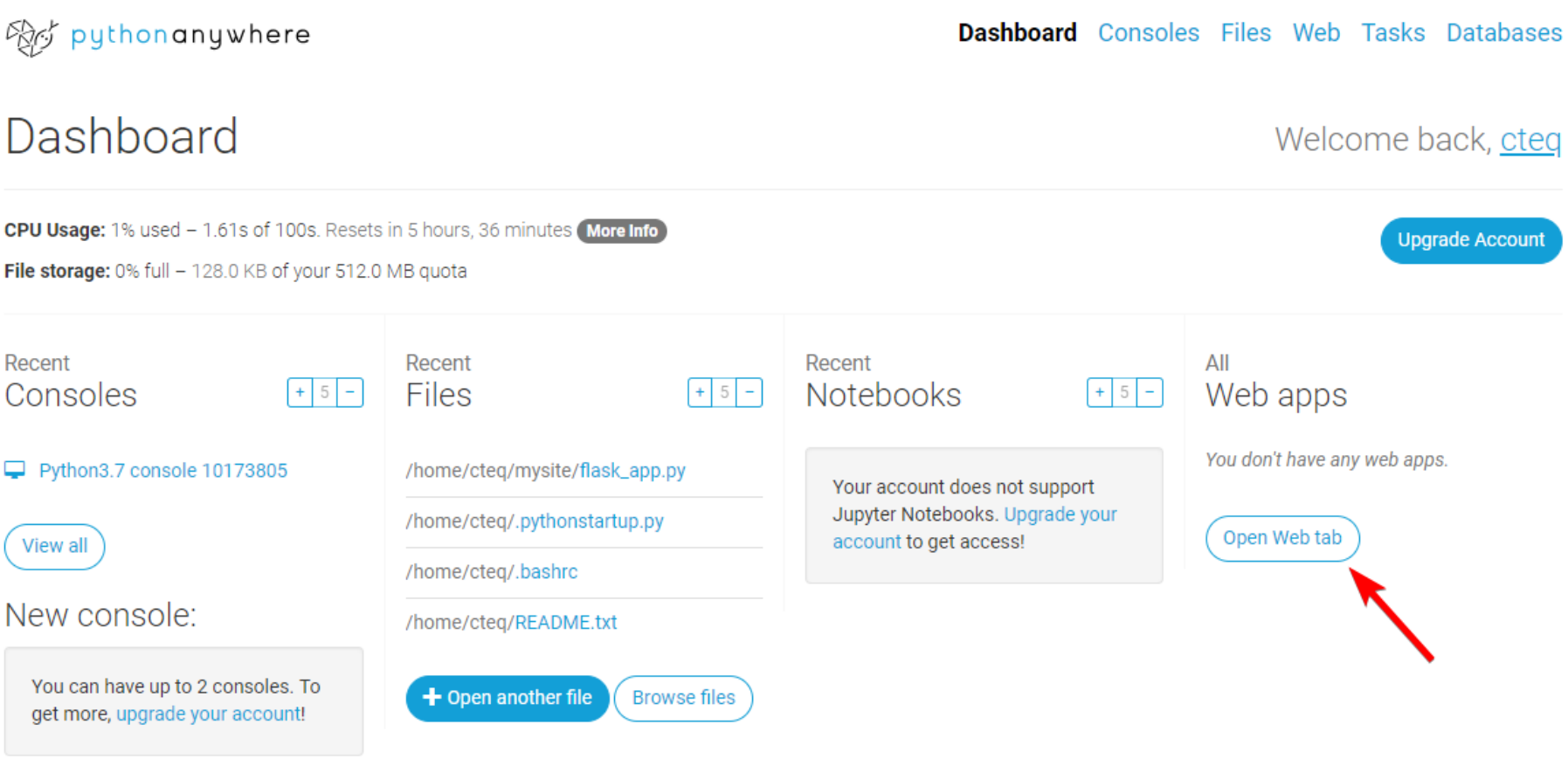
- Pythonanywhere
  - https://www.pythonanywhere.com

- Flask Python Library
  - http://flask.pocoo.org

Log in or Sign up @ pythonanywhere

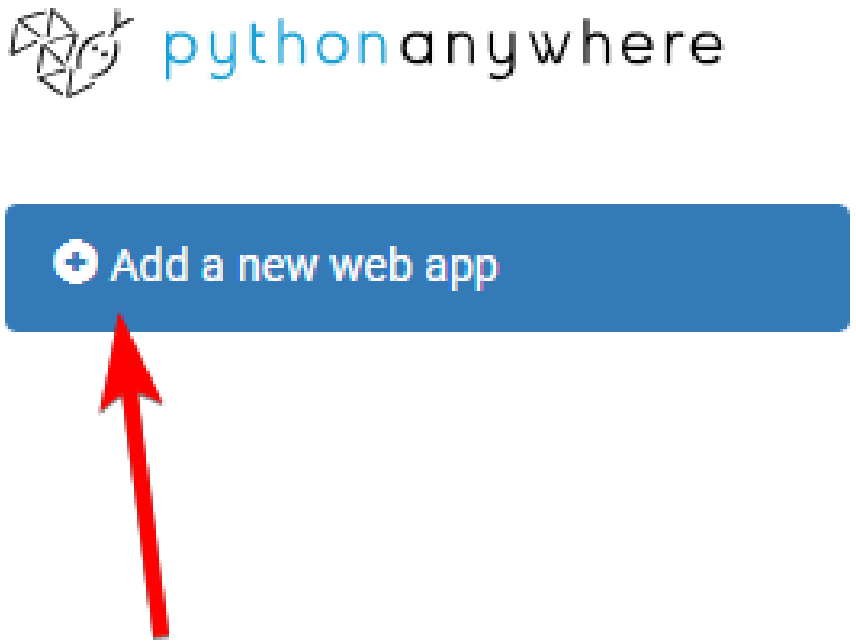


Open Web Apps

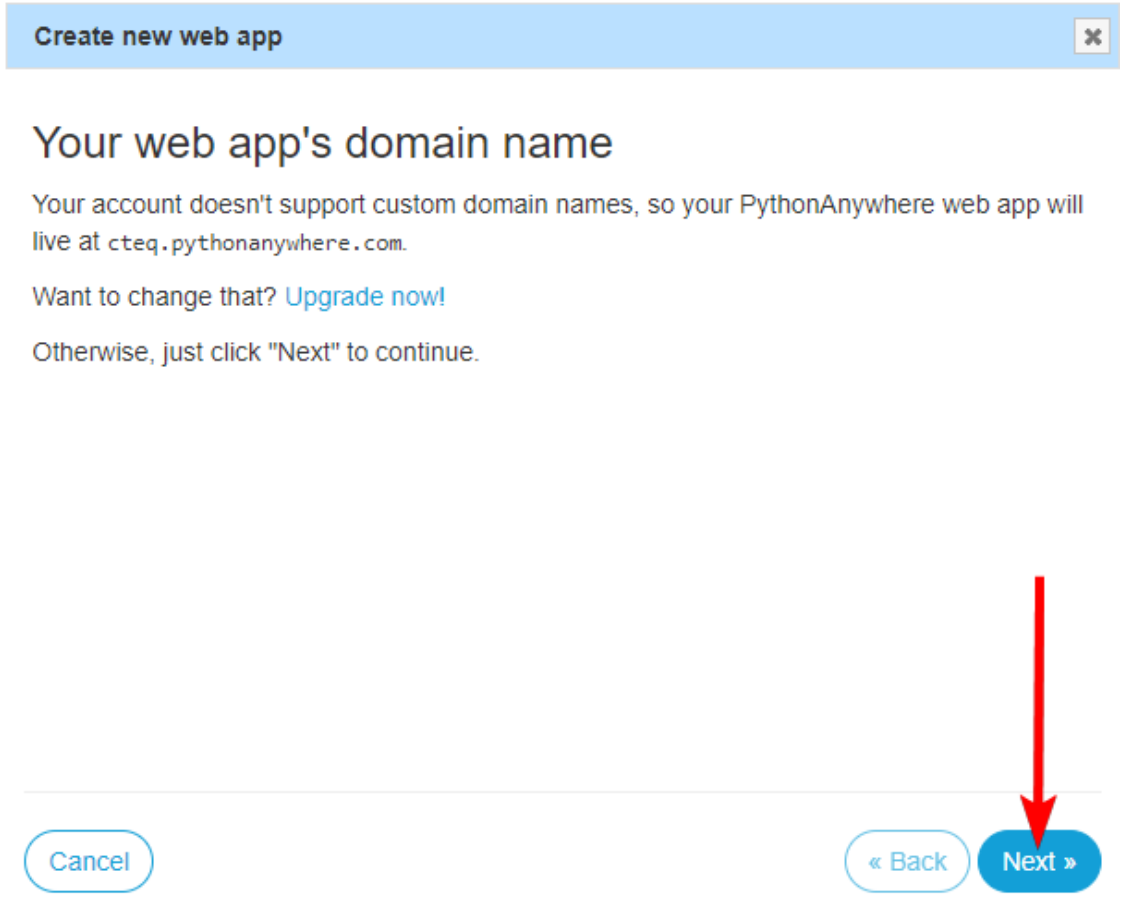


Vincent Claes

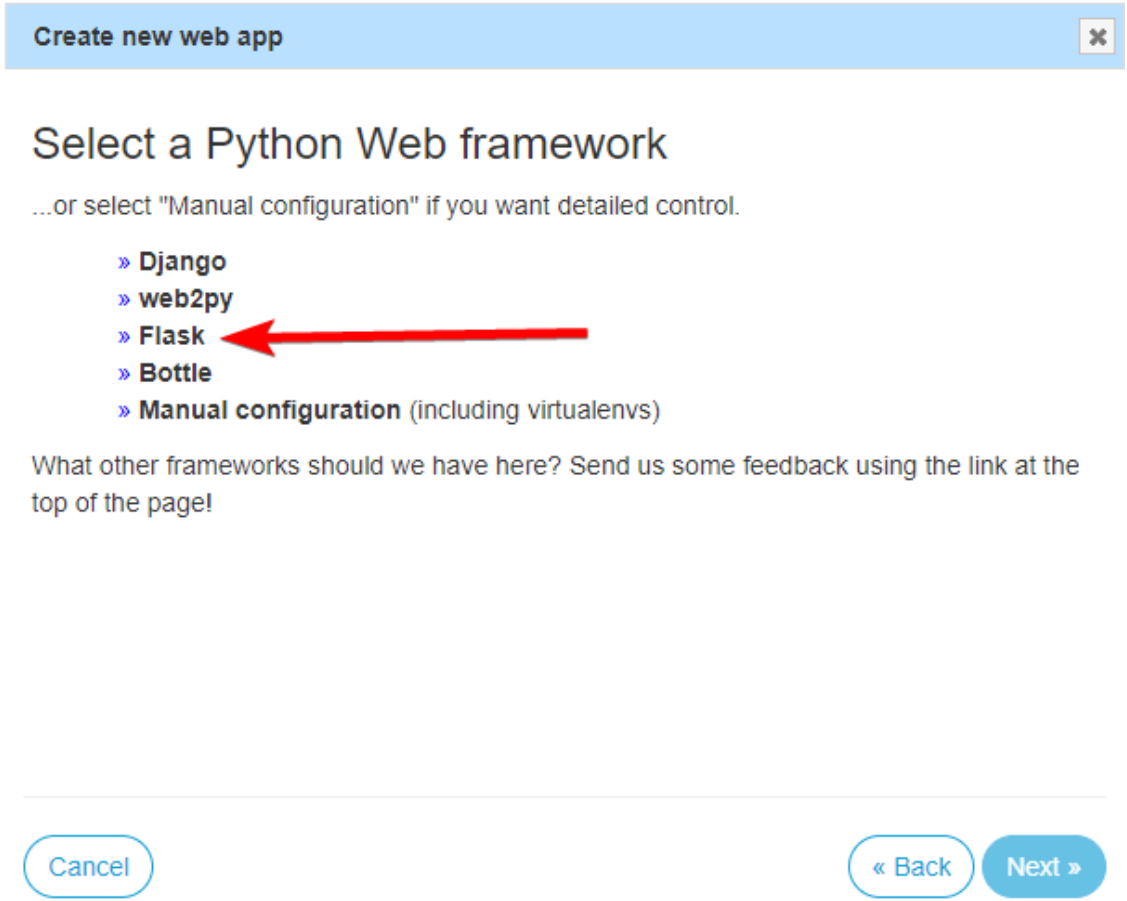
Add a new web app



Create a new web app



Flask Python Web Framework

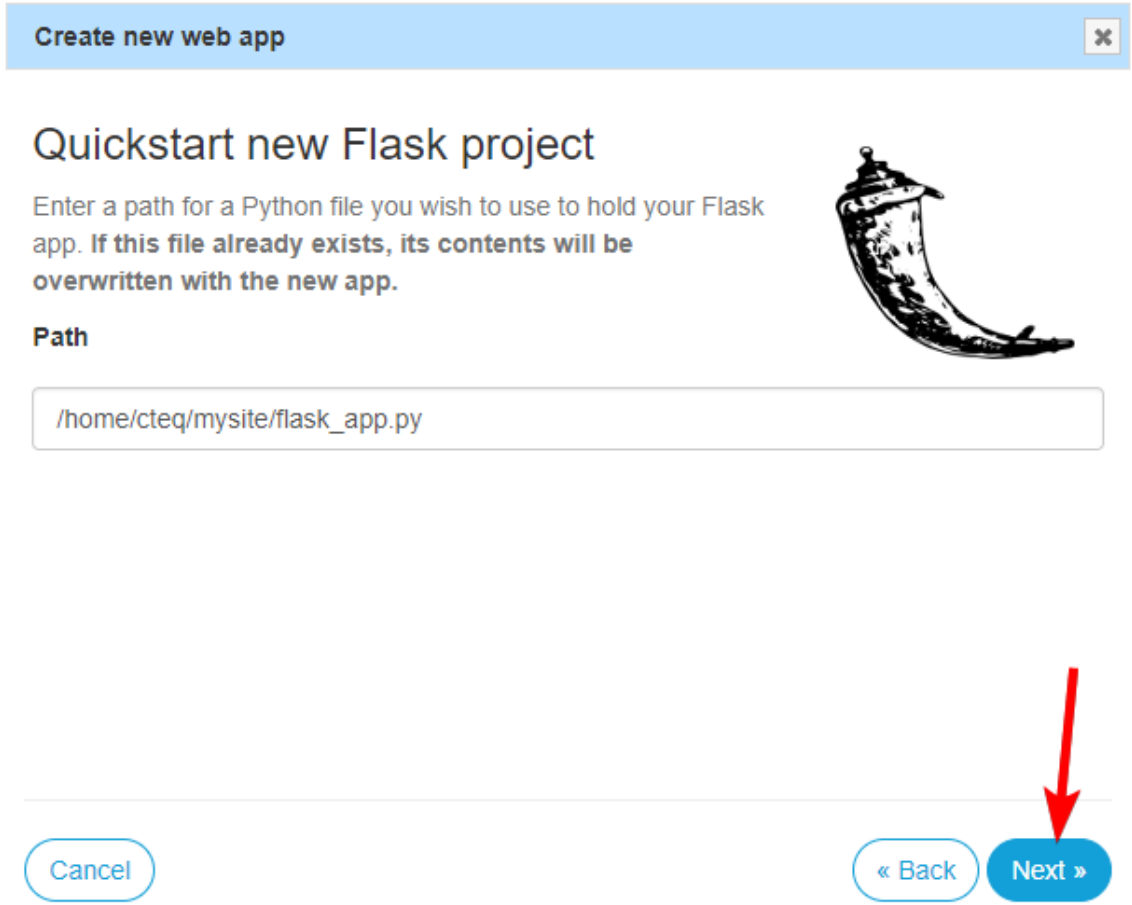


Vincent Claes

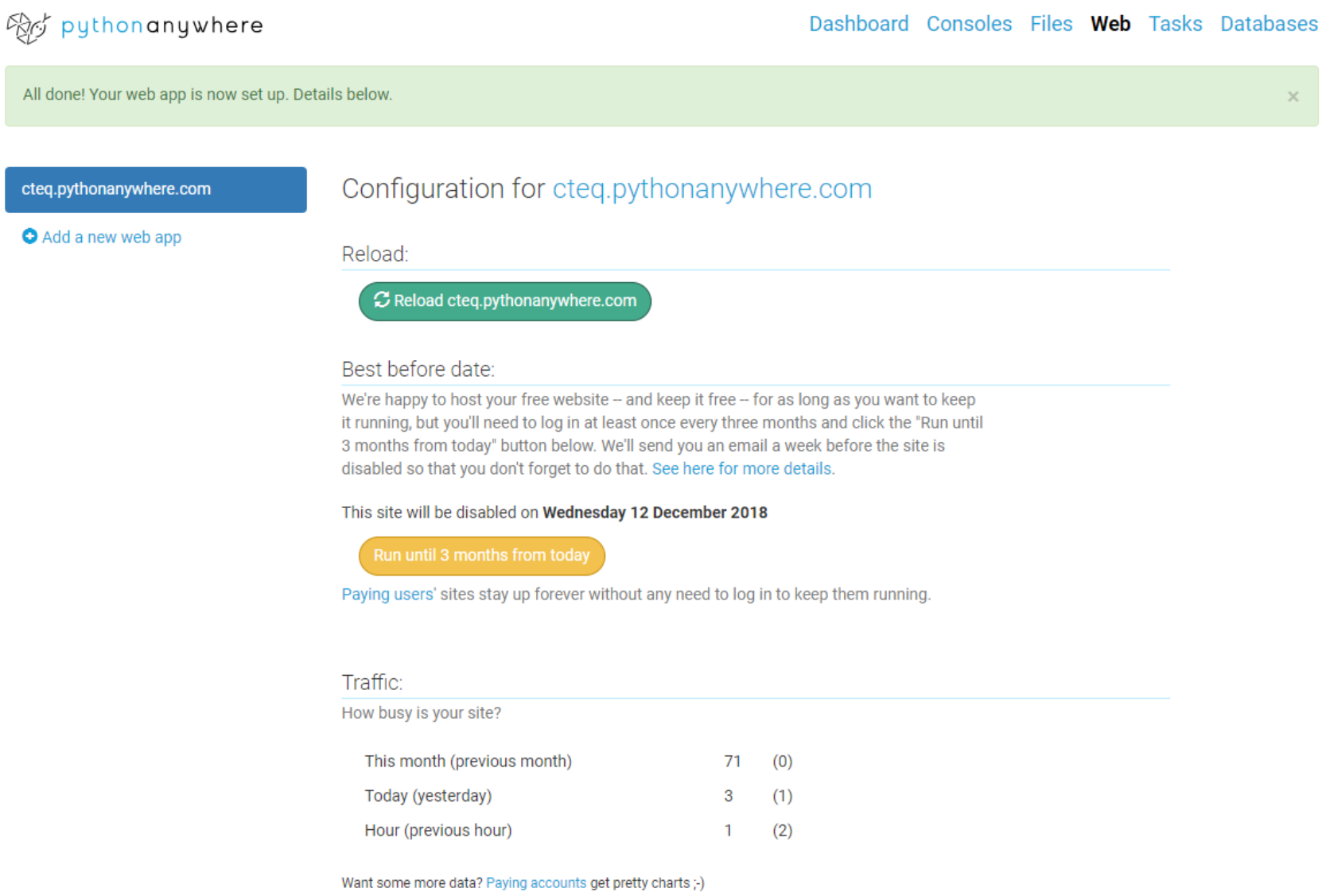
Python 3.7 (Flask 1.0.2)

Vincent Claes

Select path for python file – Flask app

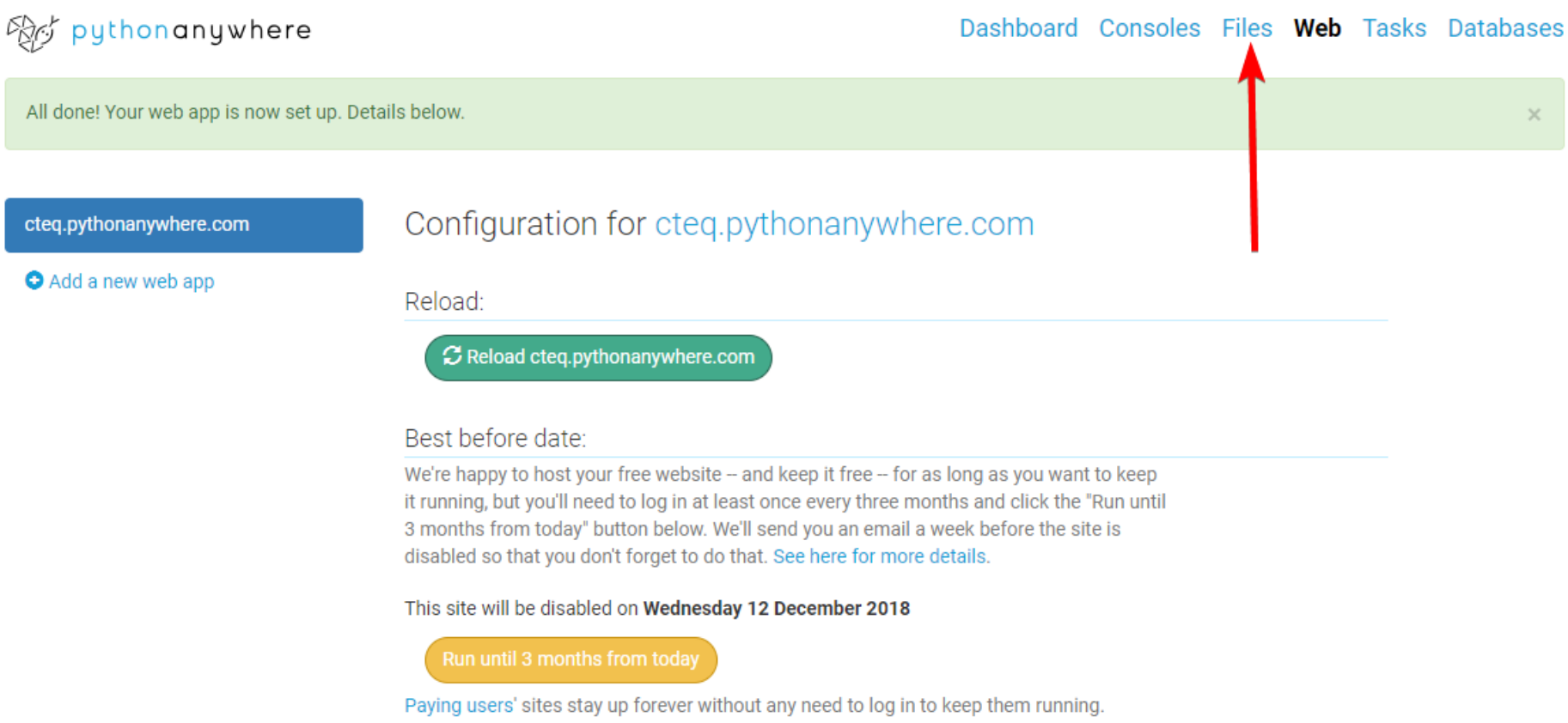


Overview Web Tab

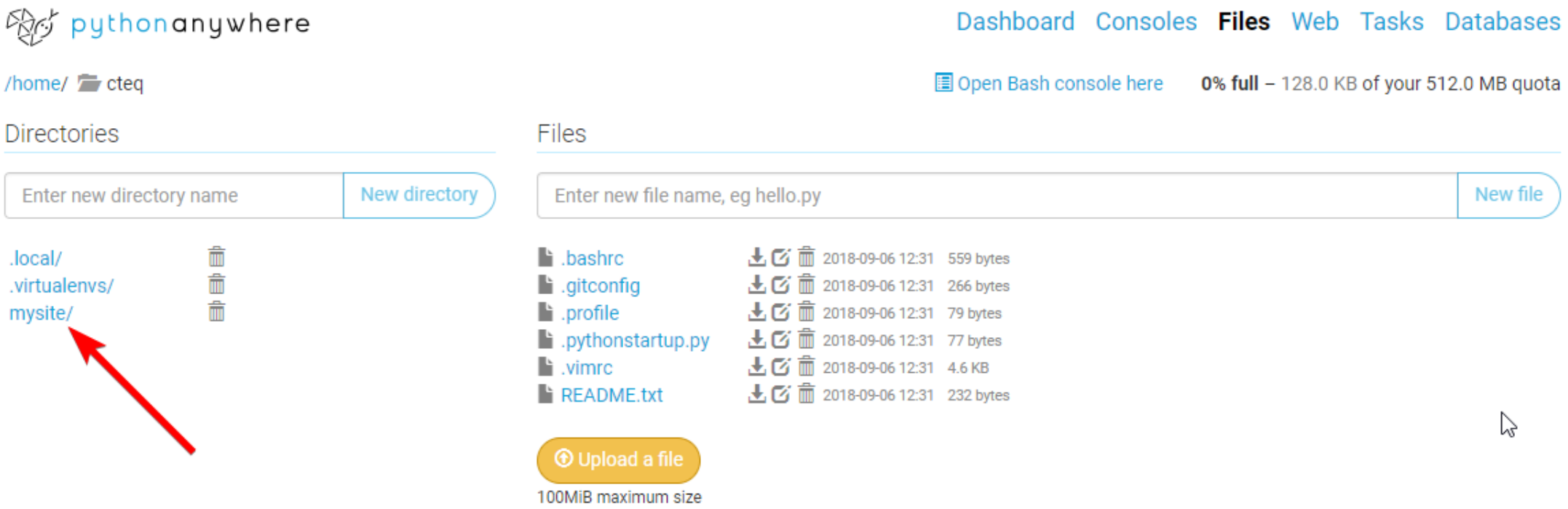


Vincent Claes

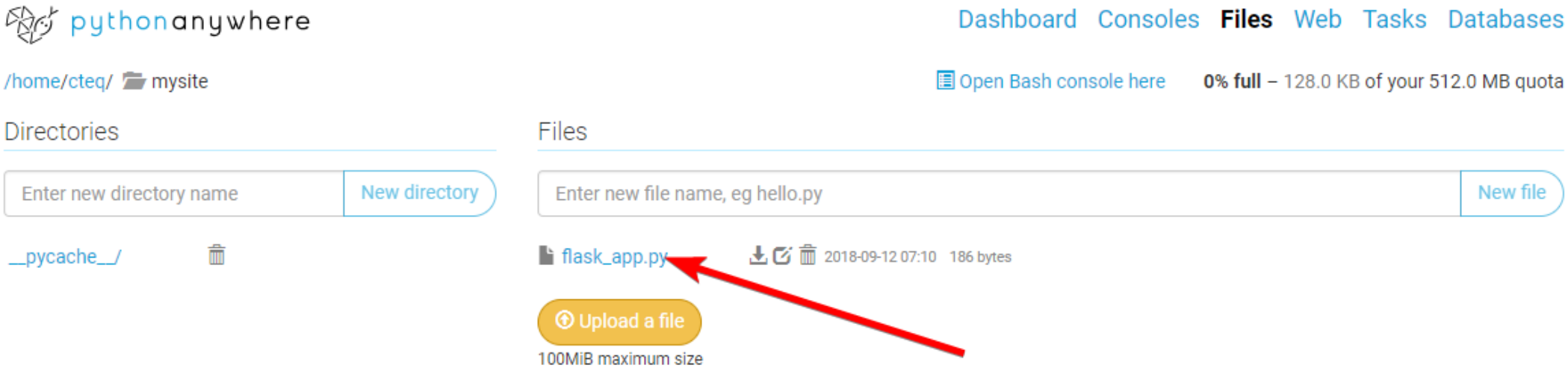
Files Tab


pythonanywhere
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

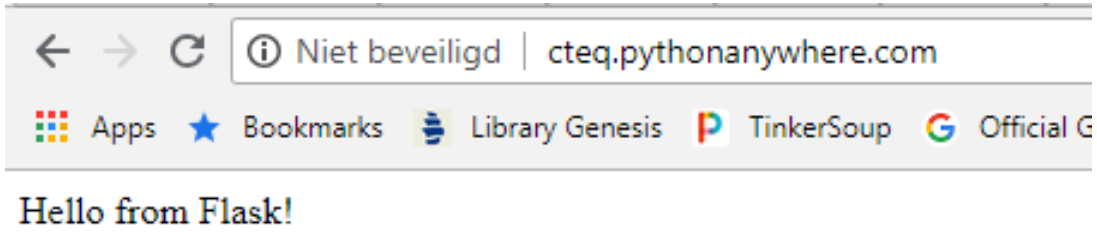


Open flask_app.py

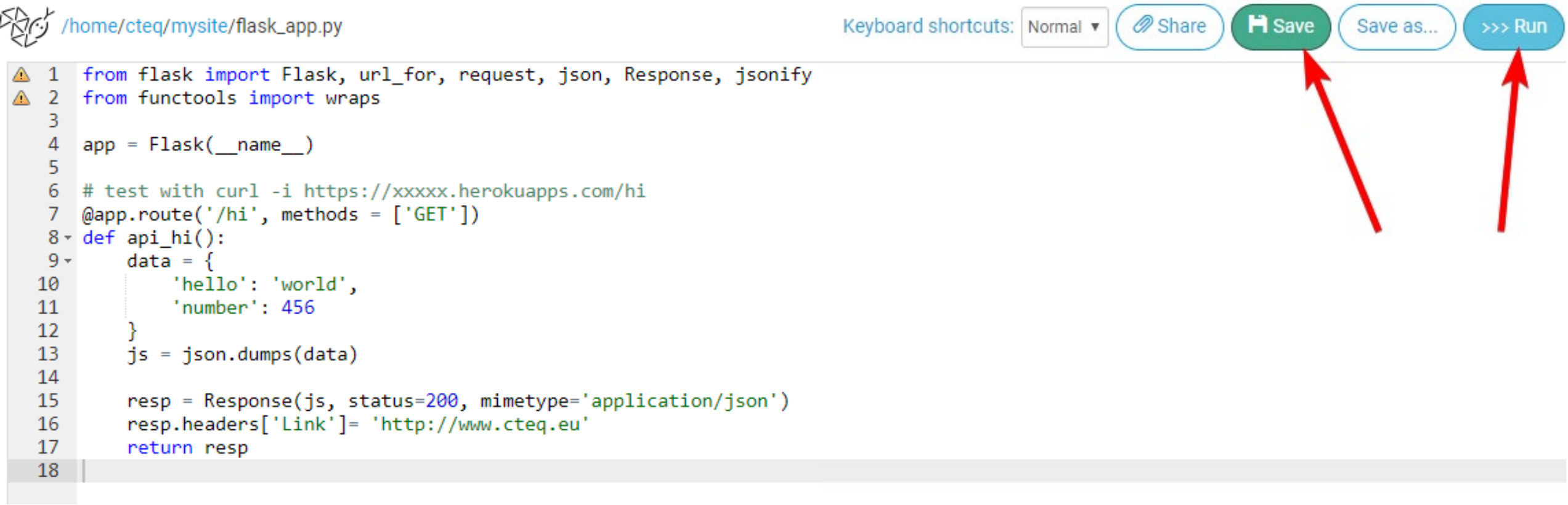


https://cteq.pythonanywhere.com

```
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
```

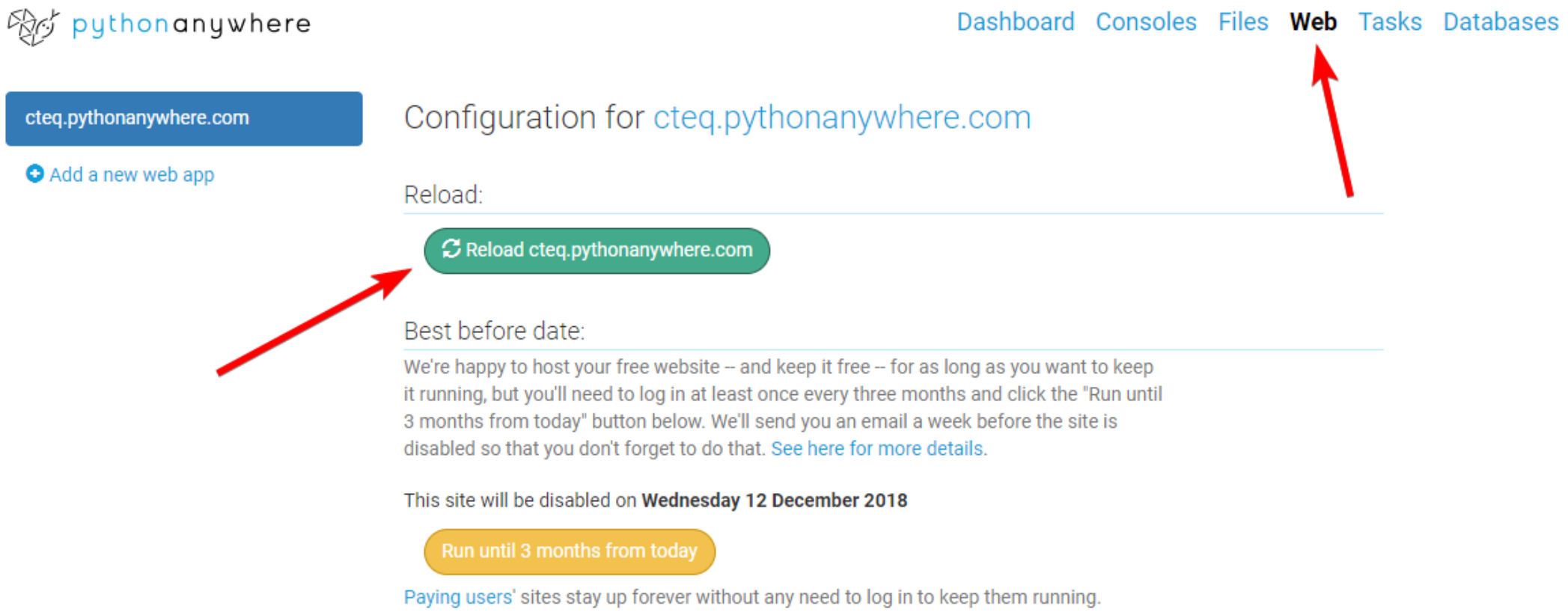


Example Flask App




```python
from flask import Flask, url_for, request, json, Response, jsonify
from functools import wraps

app = Flask(__name__)

# test with curl -i https://xxxxx.herokuapps.com/hi
@app.route('/hi', methods = ['GET'])
def api_hi():
    data = {
        'hello': 'world',
        'number': 456
    }
    js = json.dumps(data)

    resp = Response(js, status=200, mimetype='application/json')
    resp.headers['Link']= 'http://www.cteq.eu'
    return resp
```

Web Tab => reload



Webserver2.py

- webserver2.py
- map templates aanmaken

```python
# Developed by Vincent Claes
# vincent@cteq.eu
# 14/11/2018

from flask import Flask, json, Response, render_template

app = Flask(__name__)

# test with curl -i https://xxxxx.pythonanywhere.com/hi
@app.route('/hi', methods = ['GET'])
def api_hi():
    data = {
        'hello': 'world',
        'number': 456
    }
    js = json.dumps(data)
    resp = Response(js, status=200, mimetype='application/json')
    resp.headers['Link']= 'http://www.cteq.eu'
    return resp
# test with curl -i https://xxxxx.pythonanywhere.com/hello/name
# test with curl -i https://xxxxx.pythonanywhere.com/hello/

@app.route('/hello/')

@app.route('/hello/<name>')
def hello(name=None):
    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

Oefeningen Webserver

- Schrijf code voor een webserver die een parameter verwacht “tekst”, print deze tekst in hoofdletters op jet website

- Voeg CCS / HTML toe aan je website zodoende de site er mooier uitziet
  - 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

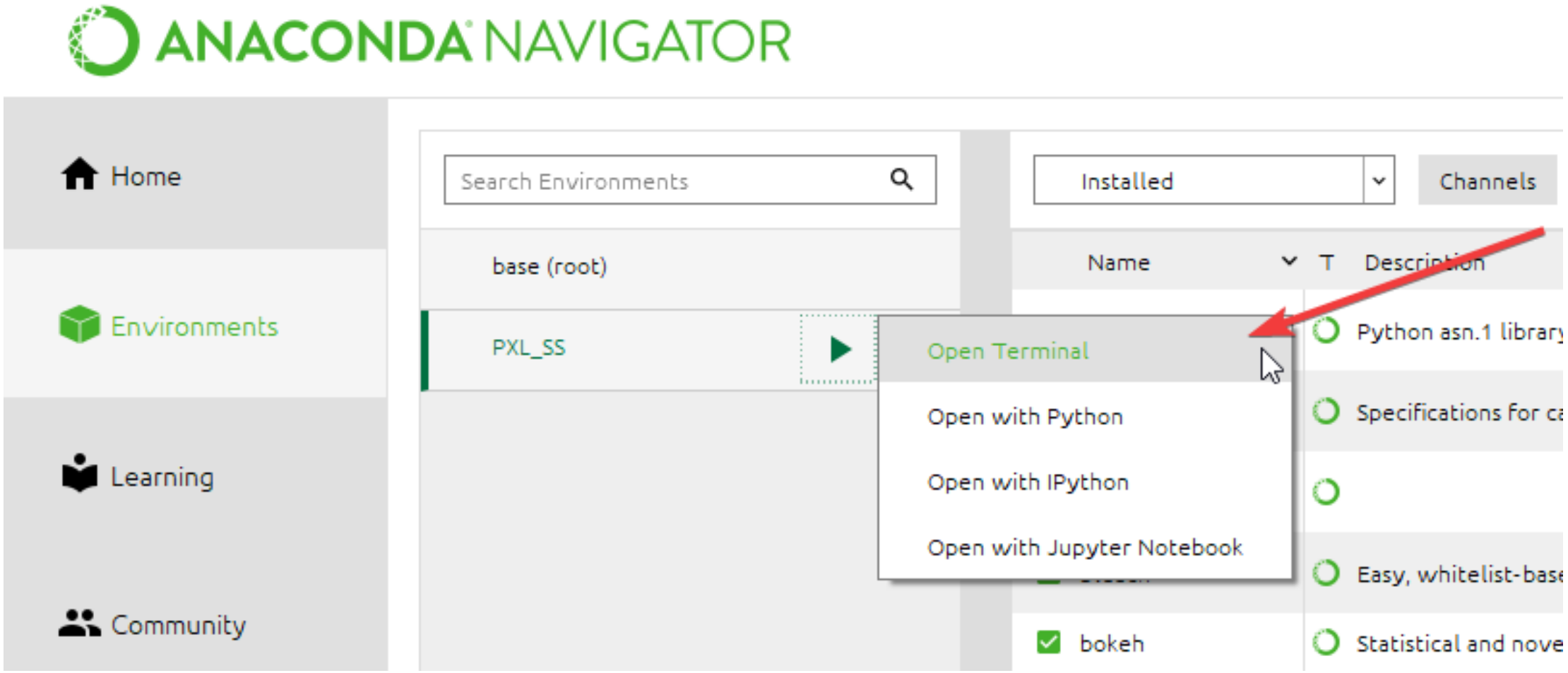
- Bouw een eenvoudig webserver rekenmachine

- 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

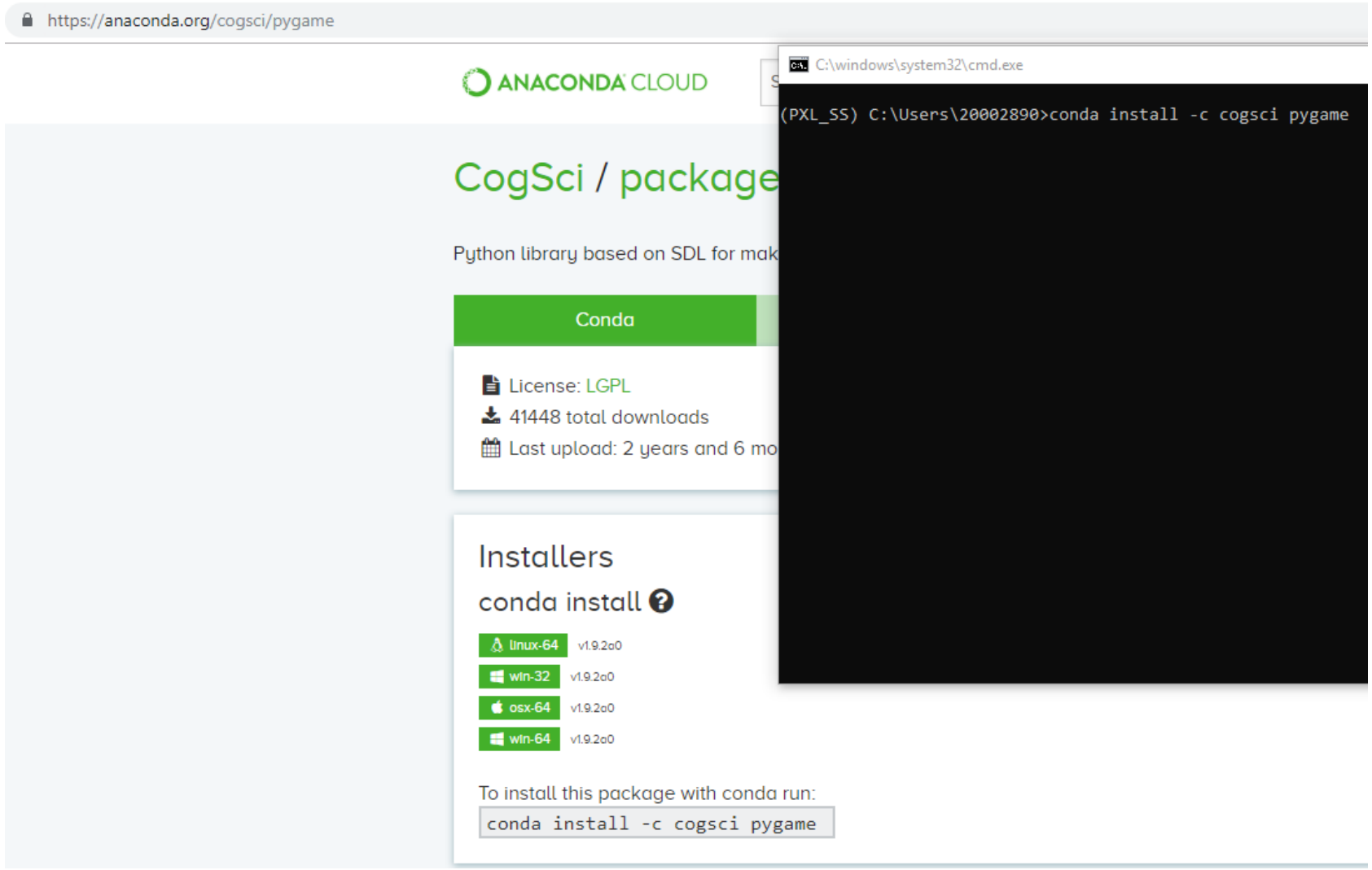
- Gebruik van pygame library
  - https://www.pygame.org

Python: Game Programming



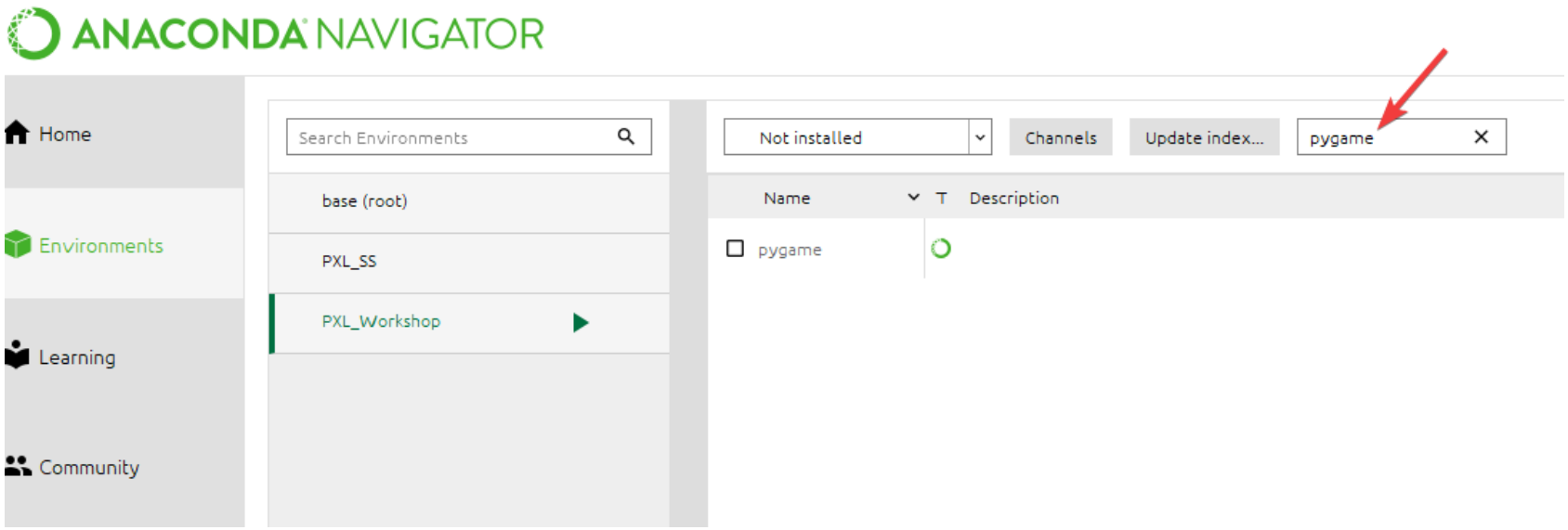
Vincent Claes

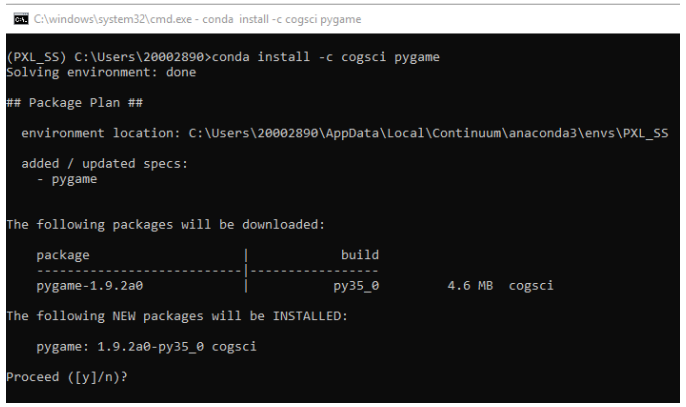
Python: Game Programming



Vincent Claes

Python: Game Programming: PyGame installatie – alternatieve methode





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

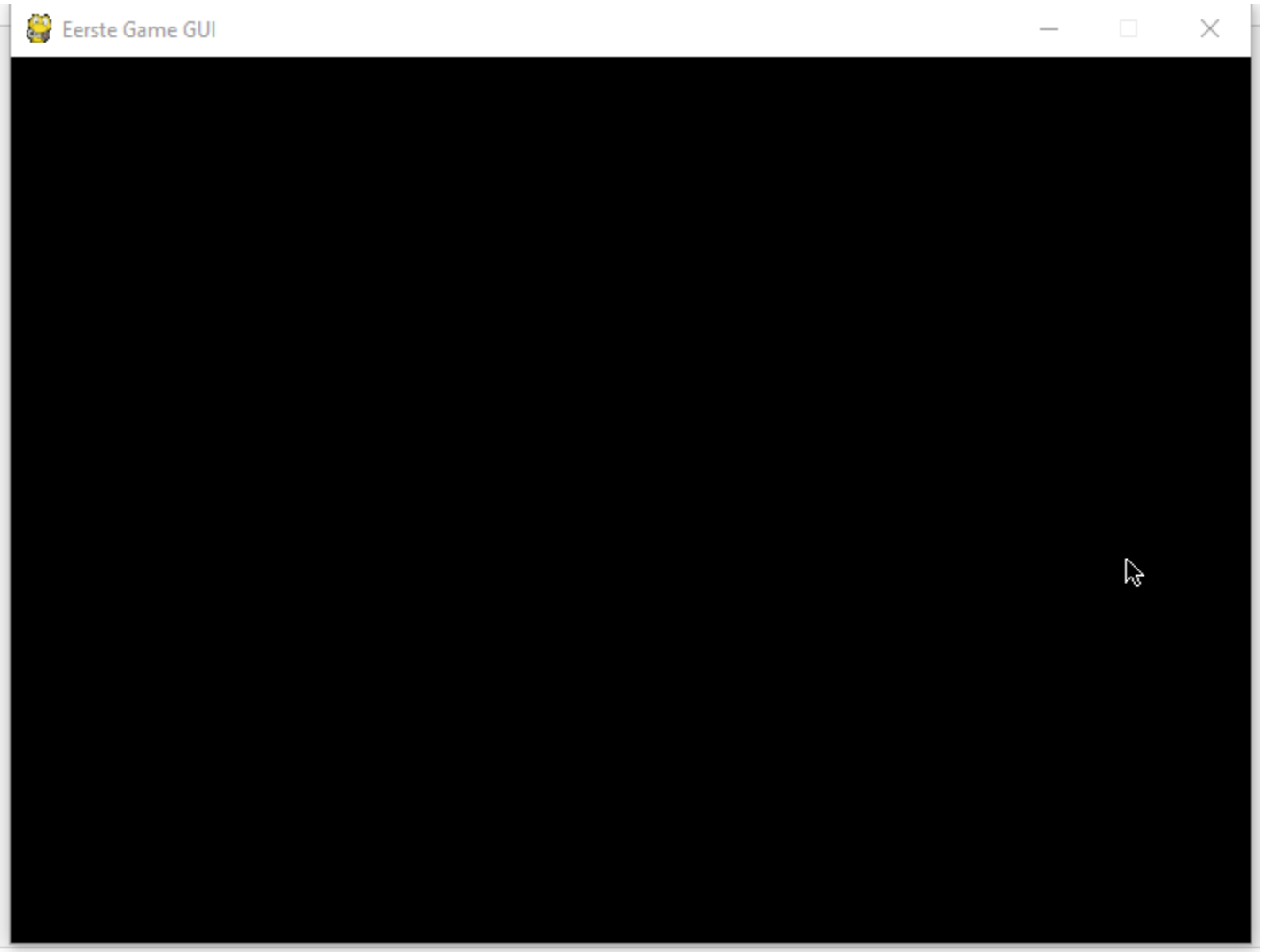
Vincent Claes

Python: Game Programming

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

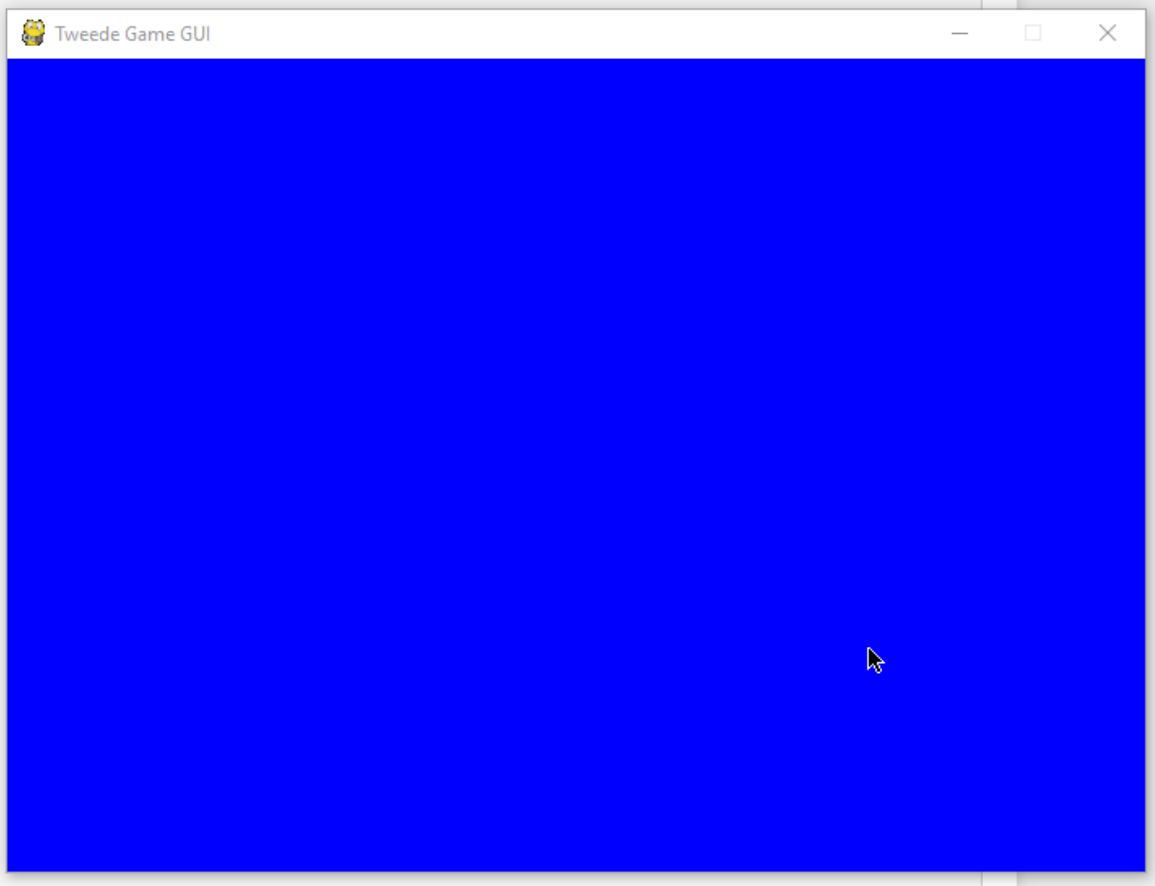
Python: Game Programming

```python
# 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 beëindigen
            carryOn = False

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



Python: Game Programming

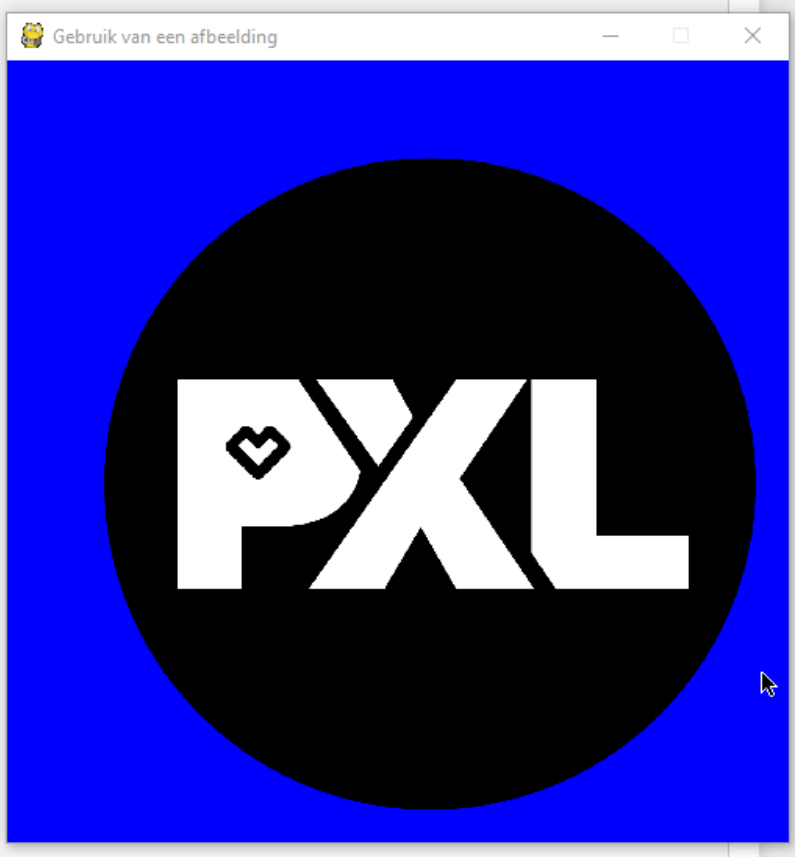
```python
# Pygame library importeren
import pygame
# Game Engine Initialiseren voor gebruik
pygame.init()

# 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 beëindigen
            carryOn = False

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

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



Vincent Claes

Python: Game Programming

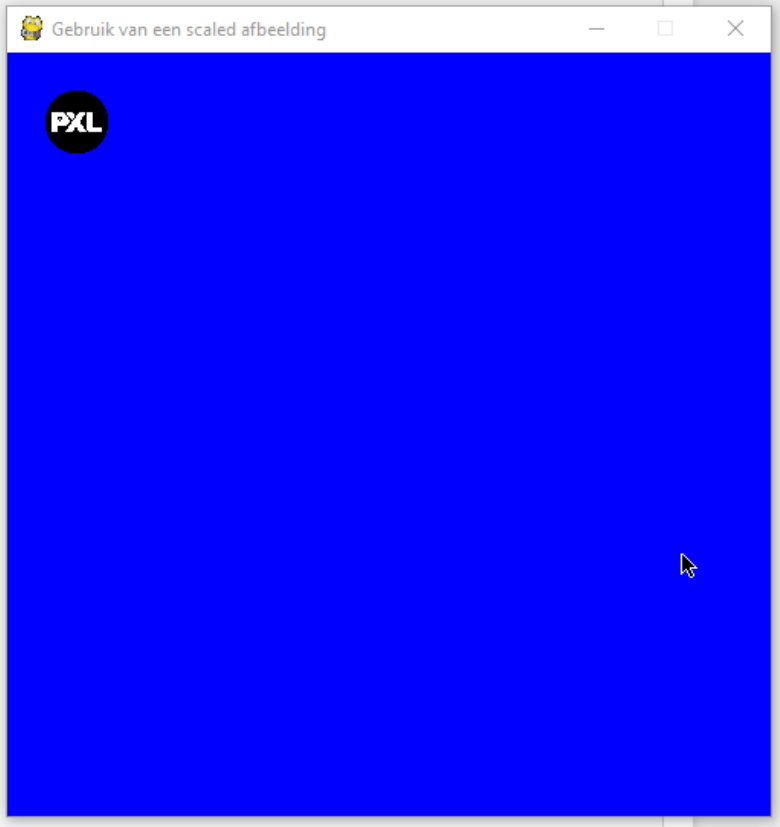
```python
# Pygame library importeren
import pygame
# Game Engine Initialiseren voor gebruik
pygame.init()

# 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 beëindigen
            Loop = False

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

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



Vincent Claes

Python: Game Programming

```python
# 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()
```



Vincent Claes

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()
```

Vincent Claes

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).