

A collection of abstract geometric shapes in yellow, blue, red, and pink, including triangles, squares, circles, and polygons, scattered around the text.

WEBTIGERPYTHON

EDU-I-DAY

CLEMENS BACHMANN

WHO AM I

- Clemens Bachmann
- Doctoral / Teachers Diploma Student
- clemens.bachmann@inf.ethz.ch
- Algorithms and **Didactics** Group
- Working in the Group of Dennis Komm



Illustration by Anna Staub

WEBTIGERPYTHON



PROGRAMMING IN THE BROWSER

- New Features
- Hands-on Examples
- Code skeletons
- <https://linktr.ee/webtp>



WebTigerPython

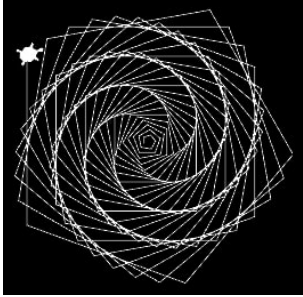
TigerJython for the Web



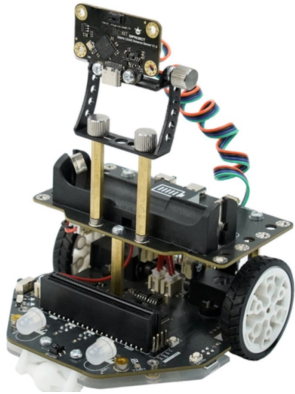
```
1 # Koch.py
2
3 from turtle import *
4
5 def koch(s, n):
6     if n == 0:
7         forward(s)
8         return
9     koch(s / 3, n - 1)
10    left(45)
11    koch(s / 3, n - 1)
12    right(90)
13    koch(s / 3, n - 1)
14    left(45)
15    koch(s / 3, n - 1)
16
17 setPenColor("blue")
18 speed(-1)
19 setPos( -100, 100)
20 for i in range(4):
21     right(90)
22     koch(100, 4)
23
```



FEATURES OF WTP

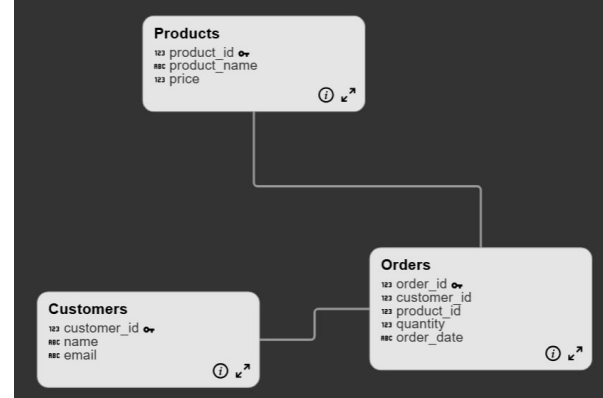


Visual Computing (turtle, gpanel, pygame)

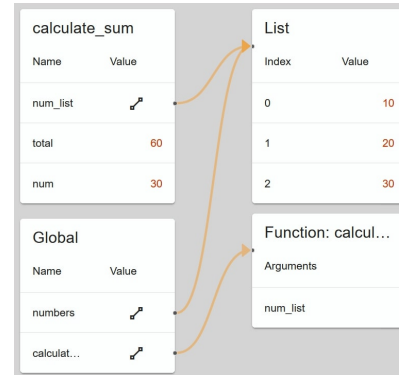


Robotics (WebUSB, simulation)

python-online.ch/



Databases (sqlite3, database1)



Debugger



- Popular library for programming games
- Sounds, images, animations
- Event handling (keyboard, mouse)
- Can be used in the browser*

*asynchronity issue

 WebTigerPython
TigerJython for the Web



1 `import pygame`

2

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

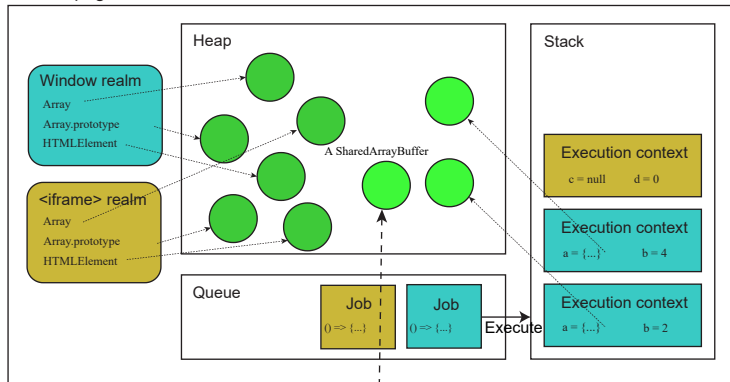
4



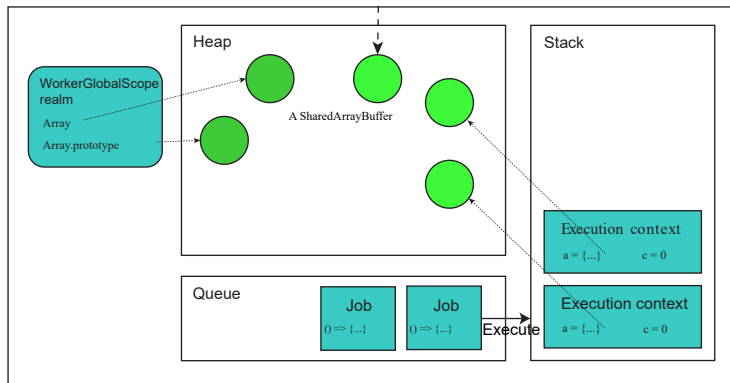
ASYNCHRONITY PROBLEM

Agent cluster

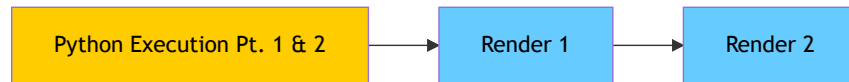
HTML page



Worker



SYNCHRONOUS EXECUTION IN BROWSER



DESIRED EXECUTION FLOW



ASYNCHRONITY EXAMPLE

```
from pygame import *
import asyncio

async def render():
    # ...
    display.flip()
    await asyncio.sleep(0)(0)

async def updateActors():
    # ...
    await render()

async def main():
    while True:
        # ...
        await updateActors()

await main()
```

PYGAME - HANDS ON

- Check out the code example
- Try to implement bullet movement
- Implement simple collision detection
 - use `x.colliderect(y)` to check if there is a collision
- Implement alien movement
- Have fun!
- linktr.ee/webtp



```
1 import sys, pygame, time
2
3 W,H = 400, 600
4 pygame.init()
5 s = pygame.display.set_mode((W,H))
6
7 # simple objects as rects
8 player = pygame.Rect(W//2-15, H-50, 30,
9 aliens = [pygame.Rect(50 + i*60, 50, 32
10 bullets = []
11
12 # gameLoop
13 while True:
14     for e in pygame.event.get():
15         if e.type == pygame.QUIT:
16             pygame.quit(); sys.exit()
17         if e.type == pygame.KEYDOWN:
18             if e.key == pygame.K_SPACE:
19                 bullets.append(pygame.R
20
21 # keyboard interaction
22 keys = pygame.key.get_pressed()
23 if keys[pygame.K_LEFT]:
24     player.x -= 4
25 if keys[pygame.K_RIGHT]:
26     player.x += 4
27 player.x = max(0, min(W-player.w, p
28
29 # TODO move bullets and simple coll
30 # use x.colliderect(y) to check if
31
32
33 # draw
```

PYGAME ALIENS EXAMPLE



WebTigerPython

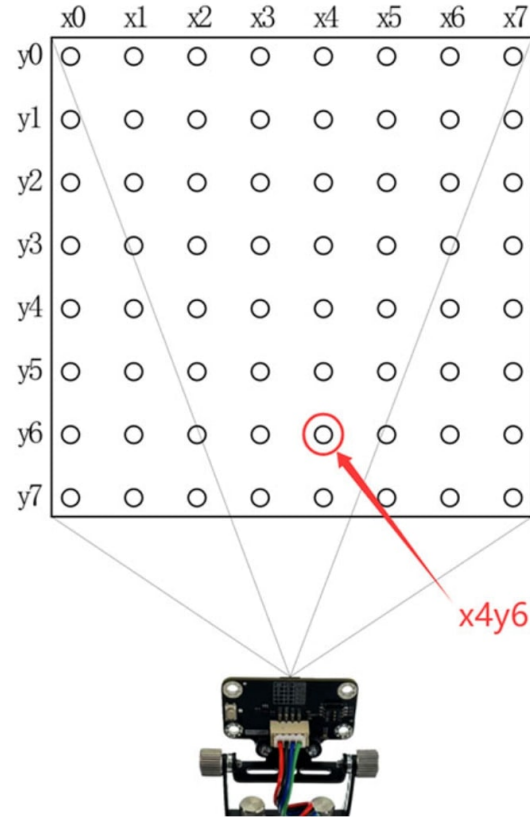
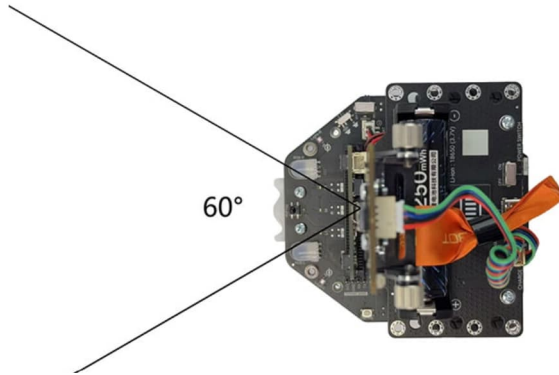
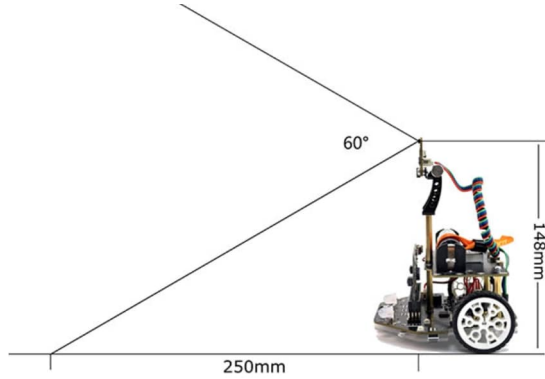
TigerJython for the Web



```
1 import pygame
2 pygame.examples.aliens.main()
3
```



MAQUEEN PLUS V3 LIDAR

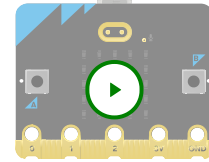


MAQUEEN PLUS V3 LIDAR - HANDS ON

- Get a micro:bit Maqueen Plus V3 Robot
- Get a micro:bit
- Screw the lidar sensor on top
- Exercise: Follow the Wall
 - `getDistanceGrid()` - gets the distance grid (slow)
 - `getDistanceAt(x,y)` - gets the distance at a specific coordinate (fast)
 - `forward()`, `left()`, `right()`,
`leftArc(radius)`, `rightArc(radius)`
- Solution



```
1 from mrobot_plusV3 import *
2
3 setSpeed(100)
4
5 repeat:
6     # Very fast
7     f1 = getDistanceAt(0,0)
8     # Rather slow
9     grid = getDistanceGrid()
10    print(f1,grid)
11    delay(1000)
12
```



ROBOTICS WITH MICRO:BIT - INTERSECTIONS

- Does not work in the simulator
- Code skeleton follows the line
- Exercise 1: Turn on type 2 intersection
- Exercise 2: Go left on four way intersection



```
1 from mrobot_plusV3 import *
2
3 setSpeed(100)
4 repeat:
5     l = not irLeft.read_digital()
6     r = not irRight.read_digital()
7     m = not irM.read_digital()
8     if m:
9         forward()
10    elif l:
11        print("right")
12        leftArc(0.1)
13    elif r:
14        rightArc(0.1)
15    else:
16        left()
17    x = intersectionDetecting()
18    ## Exercise 1:
19    # Turn on type 2
20    # four Way: 1
21    # T: 2
22    # | -: 3
23    # -| : 4
24    # Exercise 2:
25    # go Left on four way intersection
26
```

WEBTIGERPYTHON - THE FUTURE

- Improve simulator
- Multifile projects
- Git / OneDrive integration
- More libraries
- Reach out to me!
- Thank you