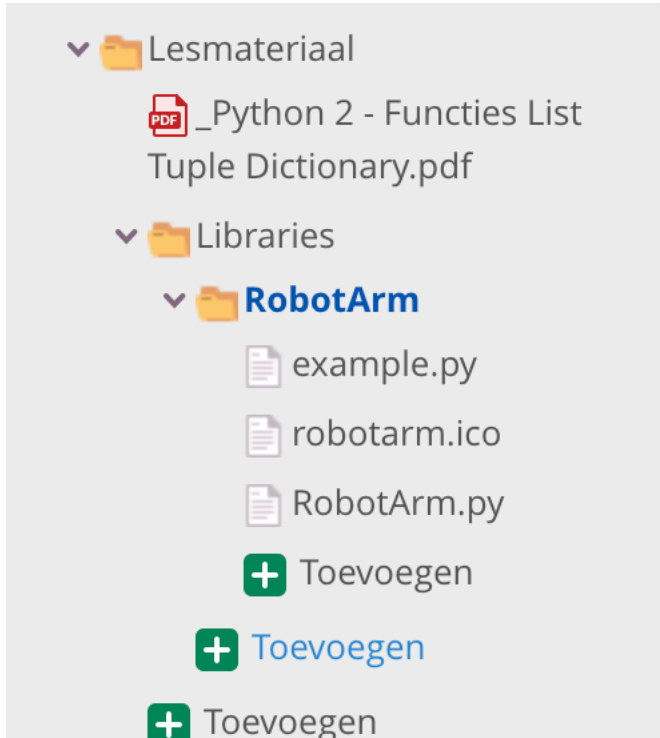


2 Pip, Class en RobotArm

In dit hoofdstuk gaan we ons bezig houden met een robotarm.

In It's learning staan drie bestanden die we eerst moeten downloaden voordat we de robot arm kunnen gebruiken.



Nadat je de bestanden hebt gedownload is het aan te raden om per hoofdstuk een map op je computer aan te maken met de opdrachten die we gaan behandelen. Op deze manier houd je je computerbestanden overzichtelijk.

Open het bestand example.py in Thonny. Dit bestand ziet er als volgt uit:

```
1 from RobotArm import RobotArm
2
3 robotArm = RobotArm('exercise 1')
4
5 # Jouw python instructies zet je vanaf hier:
6 |
7 # Na jouw code wachten tot het sluiten van de window:
8 robotArm.wait()
```

Het bestand RobotArm wordt op regel 1 geïmporteerd in je programma. Hierin staat een class met dezelfde naam RobotArm.

Op regel 3 wordt de class geïnitieerd. Een class is een nieuw datatype dat van alles kan omvatten. We gaan deze class alleen nog gebruiken, en niet zelf maken. We maken een instantie aan van code dat

allemaal te maken heeft met de robotArm. Een instantie betekend eigenlijk dat we het aan een variabele koppelen zodat we er gebruik van kunnen maken.

De variabele robotArm kan allerlei operaties (dingen) uitvoeren zoals op regel 8 wachten. Als we de code uitvoeren dan zien we waarschijnlijk de onderstaande foutmelding.

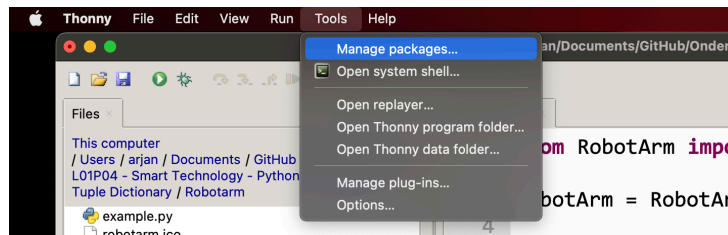
```
>>> %Run example.py
Traceback (most recent call last):
  File "/Users/arjan/Documents/GitHub/Onderwijs/L01P04 - Smart Technology - Python 2 - Functies List Tuple Dictionary/Robotarm/example.py", line 1, in <module>
    from RobotArm import RobotArm
  File "/Users/arjan/Documents/GitHub/Onderwijs/L01P04 - Smart Technology - Python 2 - Functies List Tuple Dictionary/Robotarm/RobotArm.py", line 1, in <module>
    import pygame # install in terminal with: pip install pygame
ModuleNotFoundError: No module named 'pygame'
```

Deze foutmelding geeft aan dat pygame niet gevonden kan worden. Pygame is een stuk programmatuur waarmee je iets makkelijk op het scherm kunt tekenen.

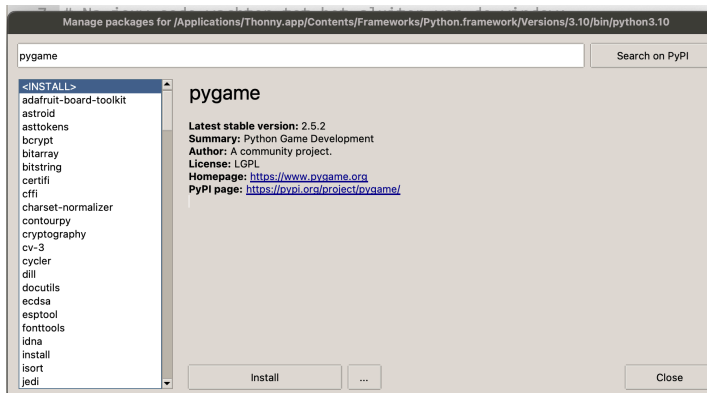
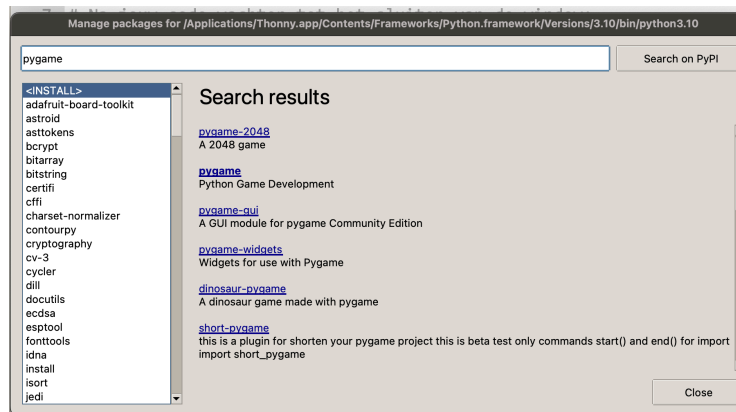
We hebben al eerder met time en random gezien dat we extra functionaliteit kunnen binnenhalen in ons programma. Random en time stonden alleen al standaard op je computer. Maar er zijn nog heel veel dingen die gemaakt zijn om te gebruiken maar die je nog moet binnenhalen naar je computer.

Dit gaan we als volgt doen.

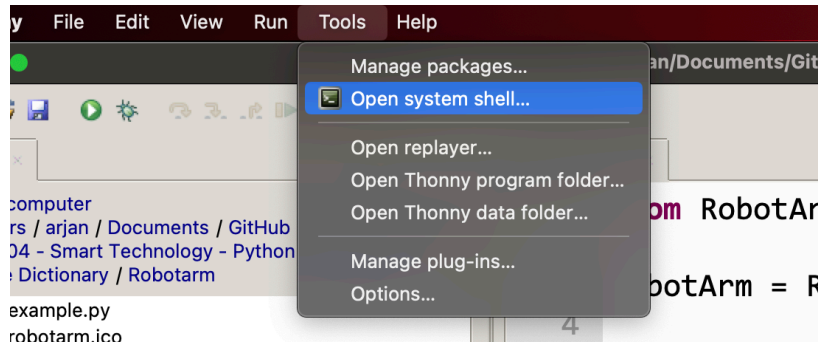
2.1 Package downloaden: Methode 1



Als we Manage packages kiezen in het menu dan kunnen we zoeken naar pygame. We selecteren dan pygame die aan de rechterkant verschijnt. Druk vervolgens op install.



2.2 Package downloaden: Methode 2



We openen de **system shell** en type dan in: **pip install pygame** zoals hieronder staat weergegeven. Nu wordt direct het de bestanden op je computer geïnstalleerd.

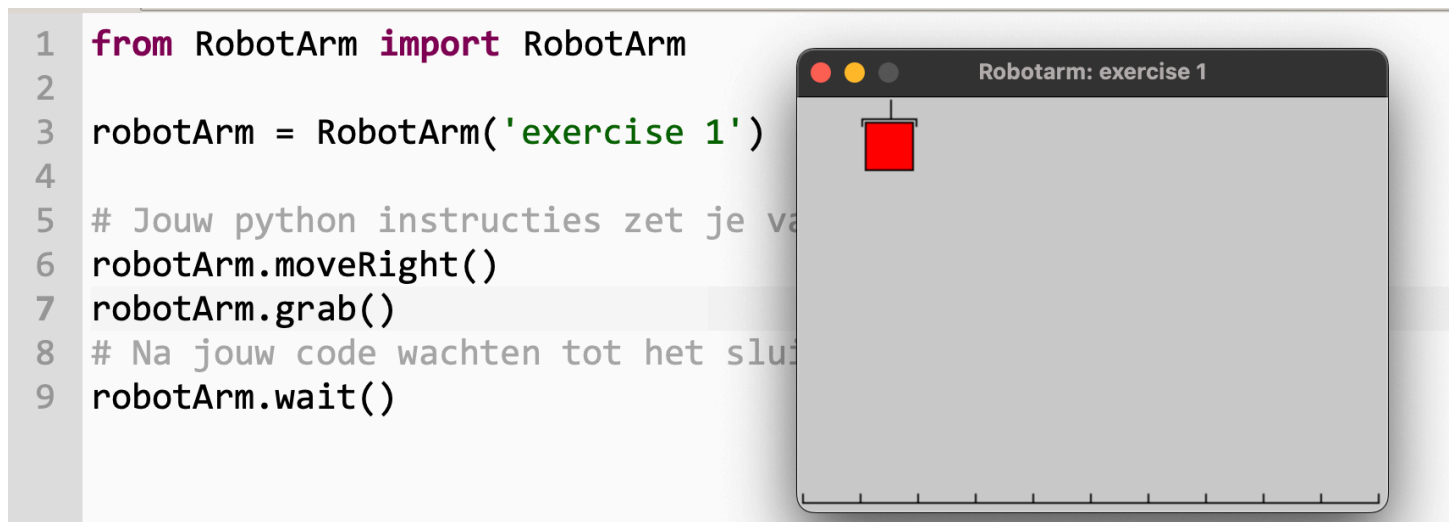
```
arjan@Arjans-MacBook-Pro Robotarm % pip install pygame
Defaulting to user installation because normal site-packages is not writeable
Collecting pygame
  Downloading pygame-2.5.2-cp310-cp310-macosx_10_9_x86_64.whl (12.9 MB)
    12.9/12.9 MB 3.5 MB/s eta 0:00:00
Installing collected packages: pygame
Successfully installed pygame-2.5.2

[notice] A new release of pip is available: 23.0 -> 23.3.1
[notice] To update, run: python3.10 -m pip install --upgrade pip
arjan@Arjans-MacBook-Pro Robotarm %
```

Als we de code nu runnen dan zien we een huiskraan met een blokje op het scherm verschijnen.

Op de [wiki pagina](#) staan instructies hoe de robotarm bibliotheek werkt.

Daar staat onder anderen als we de code aanpassen naar het onderstaande de grijparm een naar rechts gaat en het blokje optilt.



Vervolgens ga je proberen de onderstaande oefeningen te maken. Sla ieder bestand met een unieke naam op je computer op. Je kunt hiervoor de optie **save as** gebruiken in het files menu.

De latere opdrachten zijn zo gemaakt dat je een maximaal aantal regels mag gebruiken. Als dit nog niet lukt dan weet je dan je nog van alles kan leren tijdens de komende weken. Lukt dit nog niet, maar ze met zo min mogelijk stappen.

2.3 Opdrachten

2.3.1 Oefening 1

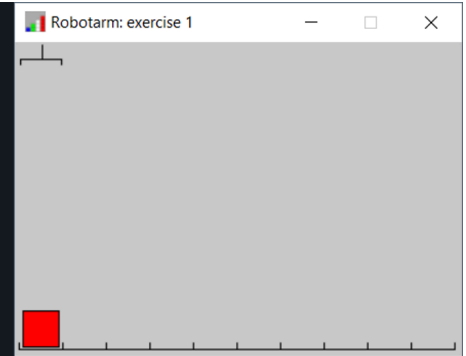
Verplaats het rode blok één plek naar rechts.

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 1')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.2 Oefening 2

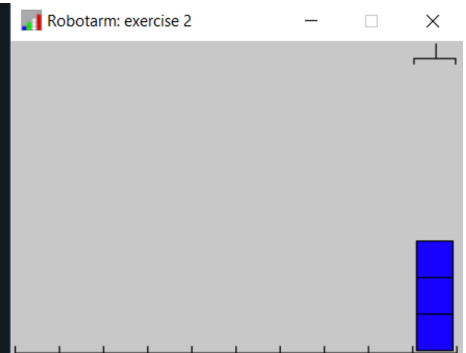
Stapel alle blokken op aan de rechterkant.

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 2')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.3 Oefening 3

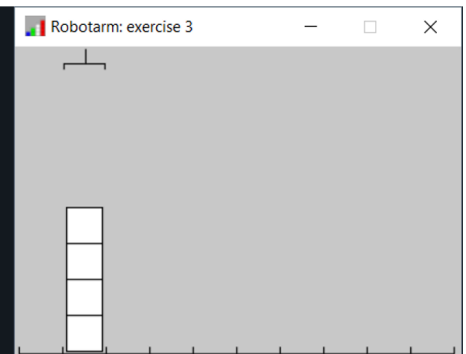
Verplaats de hele stapel blokken één plek naar rechts.

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 3')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.4 Oefening 4

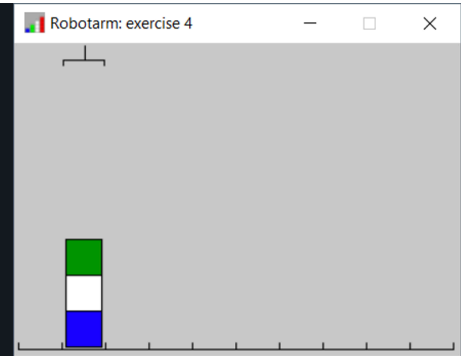
Verplaats de hele stapel blokken één plek naar rechts. Zorg ervoor dat de volgorde van de blokken gelijk blijft.

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 4')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.5 Oefening 6

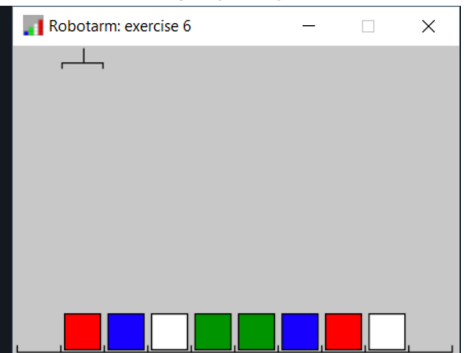
Verplaats alle blokken één plek naar rechts. Zorg ervoor dat de volgorde van de blokken gelijk blijft.

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 6')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.6 Oefening 7

Verplaats iedere stapel één plek naar links. Je mag maximaal 11 regels code gebruiken inclusief de import, het laden van de robotarm en de wait.

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 7')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.7 Oefening 8

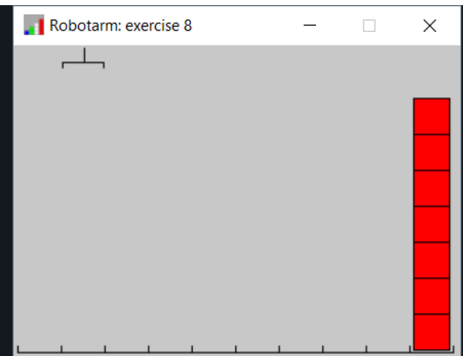
Verplaats de stapel naar de rechterkant.

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 8')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.8 Oefening 9

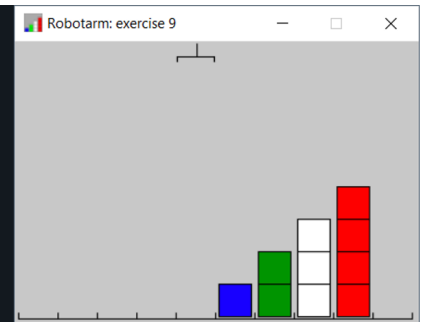
Verplaats alle stapels vijf stappen naar rechts. Je mag maximaal 12 regels code gebruiken inclusief de import, het laden van de robotarm en de wait.

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 9')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.9 Oefening 10

Draai de volgorde van de blokken om.

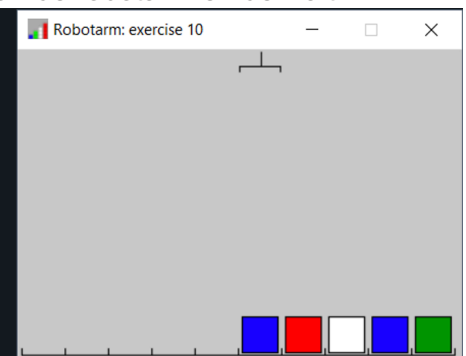
Je mag maximaal 15 regels code gebruiken inclusief de import, het laden van de robotarm en de wait

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 10')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.10 Oefening 11

Verplaats alle witte blokken één plek naar rechts.

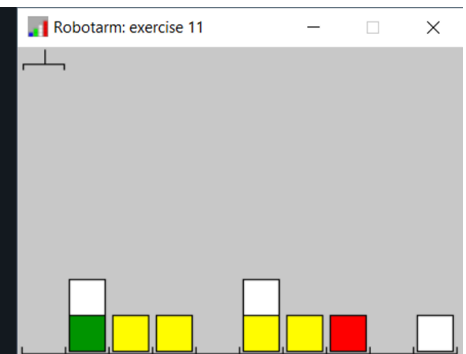
Let op, de blokken zijn iedere keer anders als je het programma start!

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 11')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.11 Oefening 12

Verplaats alle rode blokken naar het einde.

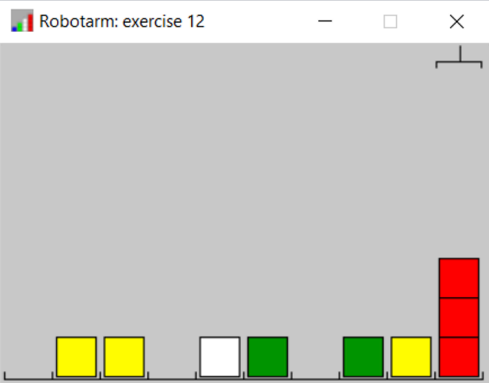
Let op, de blokken zijn iedere keer anders als je het programma start!

```
from RobotArm import RobotArm

robotArm = RobotArm('exercise 12')

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```



2.3.12 Oefening 13

Verdeel alle blokken over de lege plaatsen, zodra er geen blokken meer zijn moet de arm stoppen.

```
from RobotArm import RobotArm
# Let op: hier start het anders voor een random level:
robotArm = RobotArm()
robotArm.randomLevel(1,7)

# Jouw python instructies zet je vanaf hier:

# Na jouw code wachten tot het sluiten van de window:
robotArm.wait()
```

Voor	Na
