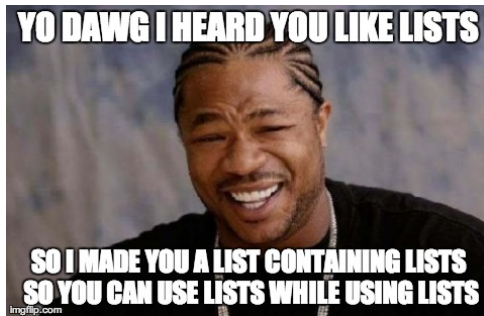




Vrije Universiteit Brussel

## Geneste lijsten

- Lijst met lijst als element



<http://www.memes.com/yo-dawg-i-heard-you-like-lists-so-i-made-you-a-list-containing-lists-so-you-can-use-lists-while-using-lists/>



Vrije Universiteit Brussel

## Geneste lijsten

- Structuur 2D lijst (matrix)

```
matrix = [ [0, 1, 2],
            [3, 4, 5],
            [6, 7, 8] ]
```

- Toegang tot elementen

– In 2 stappen

- `sublijst = matrix[i]` (geeft bij matrix een rij)
- `sublijst[j]` (geeft element in rij)

– In 1 stap

- `matrix[i][j]`

<http://www.memes.com/yo-dawg-i-heard-you-like-lists-so-i-made-you-a-list-containing-lists-so-you-can-use-lists-while-using-lists/>



Vrije Universiteit Brussel

## Geneste lijsten

- Maken van geneste lijsten

– Zoals gewone lijsten

- `List = [...]`
- `Sublist = [...]`
- `List.append(Sublist)`
- `List.insert(index, Sublist)`

– Lijsten zijn objecten zoals string, int, float, ...

<http://www.memes.com/yo-dawg-i-heard-you-like-lists-so-i-made-you-a-list-containing-lists-so-you-can-use-lists-while-using-lists/>



Vrije Universiteit Brussel

## Samenvatting Geneste lijsten

- Geneste lijst = lijst in een lijst

```
matrix = [ [0, 1, 2], [3, 4, 5], [6, 7, 8] ]
```

```
matrix[i] -> sublijst i
```

```
matrix[i][j] -> element j in sublijst i
```

```
len(matrix) -> aantal sublijsten (rijen in geval van matrix)
```

```
len(matrix[i]) -> aantal elementen in sublijst i
```

```
matrix.append([9, 10, 11]) -> nieuwe sublijst toevoegen
```

```
...
```

<http://www.memes.com/yo-dawg-i-heard-you-like-lists-so-i-made-you-a-list-containing-lists-so-you-can-use-lists-while-using-lists/>



Vrije Universiteit Brussel

## Gebruik van modules

- Laad de module
  - `import modulenaam`
- Gebruik de module
  - `modulenaam.eenfunctie()`
  - `modulenaam.eenconstante`
- Vb math module
 

```
import math
math.cos()  math.sin()  math.pi
```

[http://www.vub.ac.be/~i112002\\_2](http://www.vub.ac.be/~i112002_2)


Vrije Universiteit Brussel

## Schuine worp

- Parameters
  - Beginsnelheid, Hoek, Frequentie
- Functies
  - $X = v_0 \cdot \cos(\alpha) \cdot t$
  - $Y = v_0 \cdot \sin(\alpha) \cdot t - g \cdot t^2/2$
- cos en sin?
  - Gebruik math library! (radialen)

[http://www.vub.ac.be/~i112002\\_2](http://www.vub.ac.be/~i112002_2)


Vrije Universiteit Brussel

## Schuine worp

- Tijdsvariabele t
  - Niet continu berekenen ( $\infty$  berekening)
  - Dickeyt berekenen (op bepaalde tijdstippen/plaatsen)
- Frequentie = aantal keer per seconde
  - Frequentie  $f = 1/\text{tijd}$  (in seconde)
  - $\Rightarrow T$  op elke  $1/f$  berekenen
- Wanneer stoppen met berekenen
  - Als  $y < 0$

[http://www.vub.ac.be/~i112002\\_2](http://www.vub.ac.be/~i112002_2)


Vrije Universiteit Brussel

## Werken met bestanden

- Bestand openen
 

```
bestand = open(bestandsnaam, mode)
```

 modes:
  - » 'r' read
  - » 'w' write
  - » 'a' append
- Bestand sluiten
 

```
bestand.close()
```

[http://www.vub.ac.be/~i112002\\_2](http://www.vub.ac.be/~i112002_2)



Vrije Universiteit Brussel

## Werken met bestanden

- Bestand lezen

- Lees 1 lijn

`bestand.readline()`

geeft " terug bij EOF (end of file)

- Lees alle lijnen, lijn per lijn

*for lijn in bestand:*

...

- Ingelezen lijnen zijn strings

- Gebruik string operaties `str.strip()`, `str.split()`, ...

- Zet om naar float of int voor verdere verwerking

<http://www.vub.ac.be/~i1109a.12>



Vrije Universiteit Brussel

## Werken met bestanden

- Bestand schrijven

- schrijf lijn

`bestand.write(eenstring)`

- Gebruik strings!

- Nieuwe lijn -> `\n`

- ...

<http://www.vub.ac.be/~i1109a.12>



Vrije Universiteit Brussel

## Samenvatting bestanden

- Openen

`bestand = open(bestandsnaam, mode)`

met mode = 'r'(ead), 'w'(rite), 'a'(ppend)

- Lezen

*for lijn in bestand:*

OF `bestand.readline()`

...

- Schrijven

`bestand.write(eenstring)`

- Sluiten

`bestand.close()`

<http://www.vub.ac.be/~i1109a.12>