

Vrije Universiteit Brussel

Vraag 2: Keuzes maken

- ‘Als’... ‘Dan’ → Gebruik ‘if’ structuur!
- Syntax:

```

if condition:
    block
elif condition2:
    block2
else:
    block3
    
```

Optioneel!

http://python.eu.be/ 1.200-1

Vrije Universiteit Brussel

Vraag 2: Keuzes maken

- ‘Als’... ‘Dan’ → Gebruik ‘if’ structuur!
- Syntax:

```

if condition:
    block
    
```

→ **Expressie:**

- **Booleaanse expressie:** e.g. $x > 5$
- **Andere expressies:**
 - **False:** 0, [], ""
 - **True:** 1, elk ander getal verschillend van 0, elke niet-lege lijst, niet-lege string, ...

↓ Clause

http://python.eu.be/ 1.200-2

Vrije Universiteit Brussel

Vraag 2: Keuzes maken

1. Type 1 → Geen schrikkeljaar: niet deelbaar door 4
2. Type 2 → Wel schrikkeljaar: deelbaar door 4 maar niet door 100
3. Type 3 → Geen schrikkeljaar: deelbaar door 100 maar niet door 400
4. Type 4 → Wel schrikkeljaar: deelbaar door 400

```

def typeSchrikkel(jaar):
    if (jaar%4)!=0:                ((jaar%4)!=0)
        return 1
    elif (jaar%100)!=0:          ((jaar%4)==0) and ((jaar%100)!=0)
        return 2
    elif (jaar%400)!=0:         ((jaar%4)==0) and ((jaar%100)==0) and ((jaar%400)!=0)
        return 3
    else:
        return 4                ((jaar%4)==0) and ((jaar%100)==0) and ((jaar%400)==0)
    
```

http://python.eu.be/ 1.200-3

Vrije Universiteit Brussel

Vraag 3: Iteraties : For

- ‘Doe .. x keren’
- For loop: aantal expressies een bepaald aantal keren uitvoeren
- Syntax:

```

for variable in mylist:
    block
    
```

Ingebouwde functies:

- **range(begin,end)** of **range(end)** of **range(begin,end,step)**
- **range(2,5)** → [2,3,4]
- **range(4)** → [0,1,2,3]
- **range(10,100,20)** → [10,30,50,70,90]
- **range(100,10,-30)** → [100, 70, 40]

http://python.eu.be/ 1.200-4

Vrije Universiteit Brussel

Vraag 3: Iteraties: *While*

- ‘Zolang “een conditie” voldoet, doe...’
- **While** loop: **Onbepaald** keren iets uitvoeren → tot een bepaalde conditie is voldaan
- **Syntax:**

while condition:
block

http://python1.vub.ac.be/1.1000_3

Vrije Universiteit Brussel

Vraag 4: Iteraties over lijsten

For lus

```
mylist = [4,8,12]
```

Klassieke manier:

```
for i in range(3):
    print mylist[i]
```

4
8
12

http://python1.vub.ac.be/1.1000_4

Vrije Universiteit Brussel

Vraag 4: Iteraties over lijsten

For lus

```
mylist = [4,8,12]
```

Iteratie over lijst zelf:

```
for x in mylist:
    print x
```

4
8
12

```
for x in enumerate(mylist):
    print x
```

(0,4)
(1,8)
(2,12)

```
for (i,value) in enumerate(mylist)
```

http://python1.vub.ac.be/1.1000_7

Vrije Universiteit Brussel

Vraag 4: Iteraties over lijsten

c. Maak een functie waaraan je een string meegeeft. De functie zal dan teruggeven of deze string al dan niet een palindroom is.

Hoever moeten we gaan? → hoe lang moet uw for lus lopen?
Wanneer mogen we stoppen? → onder welke conditie mogen we reeds stoppen

lepel
↑↑ ↑↑

noren
↑↑ ↑↑

http://python1.vub.ac.be/1.1000_8