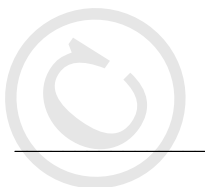


**Student-notities**  
**De programmeertaal Python**  
**Voorbeeld-hoofdstuk**

03. Statements

 **at computing**  
The Linux/UNIXperts

Nijmegen

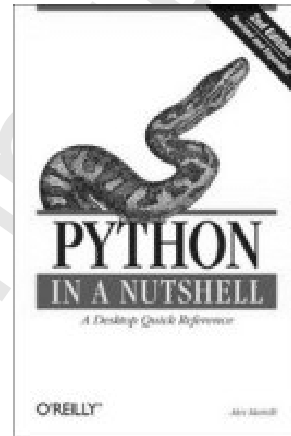
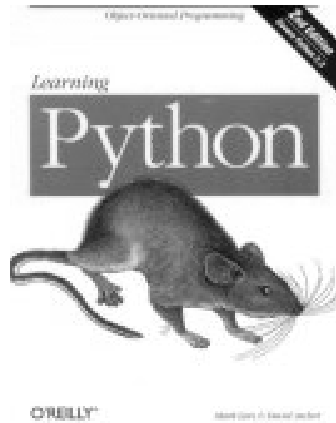


---

Copyright © AT Computing 2005-2006  
Versie: 1c

### Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Learning Python*" (van de auteurs M. Lutz en D. Ascher) en "*Python in a Nutshell*" (van de auteur A. Martelli) waarin deze materie verder is uitgewerkt.



## Programmastructuren

Statements worden één voor één uitgevoerd

Verandering van de loop van het programma

- Functieaanroepen
- **pass**
- **if/elif/elif/else**
- **while/else**
- **for/else**
- **break/continue**
- **try/except/else/finally**

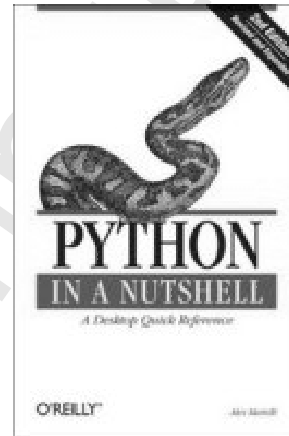
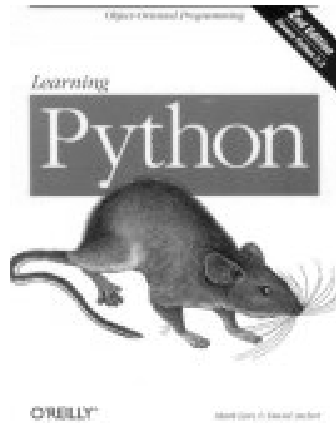
Verskil met "andere" talen:

- Geen **begin/end** of **{/}** paren
- Indentatie telt!

**Figuur 1.**

### Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Learning Python*" (van de auteurs M. Lutz en D. Ascher) en "*Python in a Nutshell*" (van de auteur A. Martelli) waarin deze materie verder is uitgewerkt.



## Blokstructuren

Bij andere blok-gestructureerde talen:

- Keywords
  - **begin .. end**
  - **while .. do .. done, case .. esac**
- Haakjesparen
  - **{ .. }**
- Combinatie
  - **<HTML> .. </HTML>**
- Daarnaast ook nog bijna altijd inspringen (conventie)
  - Het oog wil ook wat!
- Kan leiden tot verwarring!

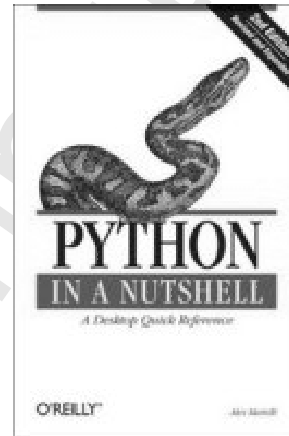
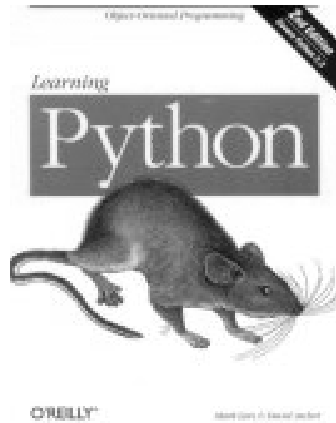
```
if cond1 then
    if cond2 then
        statement1
    else
        statement2
```

Waar hoort **statement2** bij?  
Bij **cond1** of bij **cond2**?

**Figuur 2.**

### Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Learning Python*" (van de auteurs M. Lutz en D. Ascher) en "*Python in a Nutshell*" (van de auteur A. Martelli) waarin deze materie verder is uitgewerkt.



## Python: indentatie

Bij Python:

- Wat het oog wil, is wet voor Python: Visuele aanwijzing is de enige aanwijzing. Ook voor de compiler!
- Geen verwarring meer mogelijk!

```
if cond1:
    if cond2:
        statement1
    else:
        statement2
```

```
if cond1:
    if cond2:
        statement1
    else:
        statement2
```

- Regel te lang? \ aan het einde

```
lang1 = 1 + lang2 + lang3 \
        + lang4 + lang5
```

niet nodig binnen haakjesparen (), [] en {}

```
func( 0, 1, 2, 3, 4,
      5, 6, 7, 8, 9 )
var = ( 1 + 2 + 3 +
      4 + 5 + 6 )
```

NB: Het went! Zelfs vrij snel!

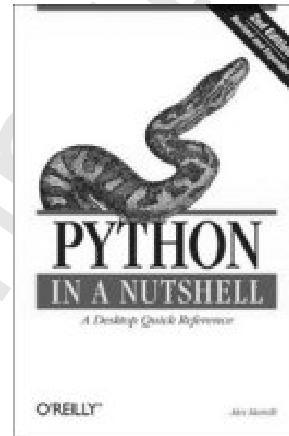
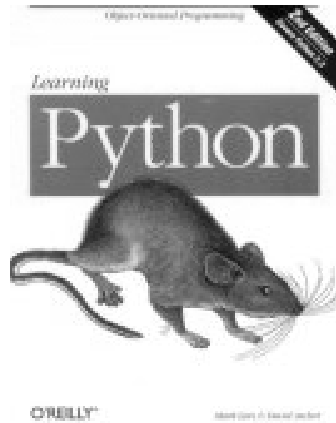
NB2: Gebruik geen tabs voor inspringen, alleen spaties

- Python telt tab altijd als 8-voud kolom-sprong
- Ook als in editor tab op 4 staat! Dat ziet python niet

**Figuur 3.**

### Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Learning Python*" (van de auteurs M. Lutz en D. Ascher) en "*Python in a Nutshell*" (van de auteur A. Martelli) waarin deze materie verder is uitgewerkt.



## If statement

### Algemene vorm

```
if conditie:  
    statement(s)  
elif conditie:  
    statement(s)  
else:  
    statements
```

### Een voorbeeld

```
if x%2 == 0:  
    print "x is even"  
elif x%3 == 0:  
    print "x oneven en een drievoud"  
else:  
    print "x oneven, geen drievoud"
```

### Slechts één statement bij de if?

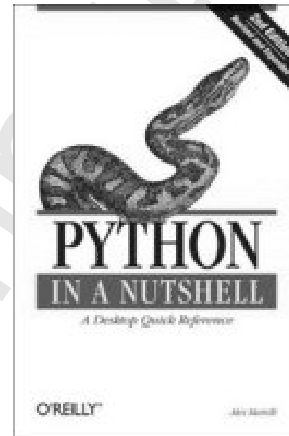
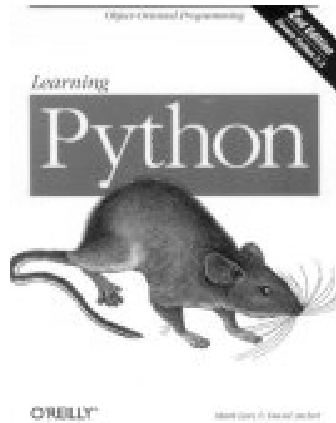
- ▶ Mag op zelfde regel!

```
if x>0: print "x is positief"
```

**Figuur 4.**

### Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Learning Python*" (van de auteurs M. Lutz en D. Ascher) en "*Python in a Nutshell*" (van de auteur A. Martelli) waarin deze materie verder is uitgewerkt.



## While statement

Algemene vorm

```
while conditie:  
    statement(s)  
else:  
    statements
```

- Voorbeelden

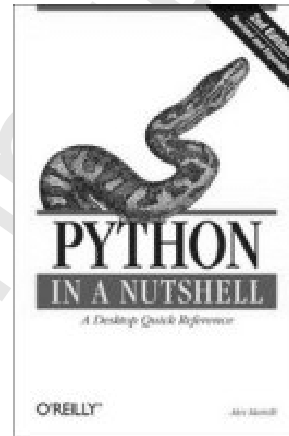
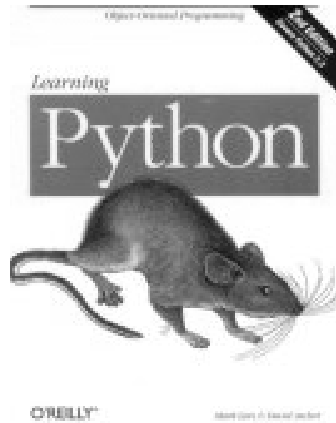
```
i=0  
n=42  
while n>0:  
    print "Lang leve Python!"  
    n=n-1  
else:  
    print "Negatieve gevoelens?"
```

```
while f()>0: print "f() is nog positief"
```

Figuur 5.

### Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Learning Python*" (van de auteurs M. Lutz en D. Ascher) en "*Python in a Nutshell*" (van de auteur A. Martelli) waarin deze materie verder is uitgewerkt.



## Break, continue, pass

**break** breekt een lus af

```
while i < 10:
    if s[i] == 'x':
        print "'x' gevonden op", i
        break
    # nog geen 'x' gevonden
    i += 1
else:
    print "x niet gevonden"
```

**continue** slaat alleen huidige slag over

```
n=0
s=0
while n < 6:
    n+=1
    if n == 2:
        continue
    s += n
print "de som van 1, 3, 4, 5, 6 is", s
```

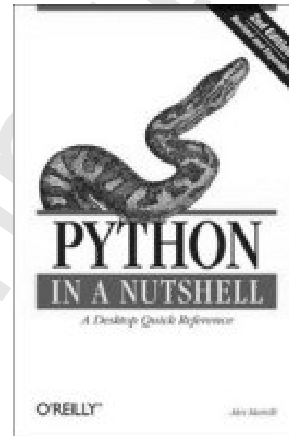
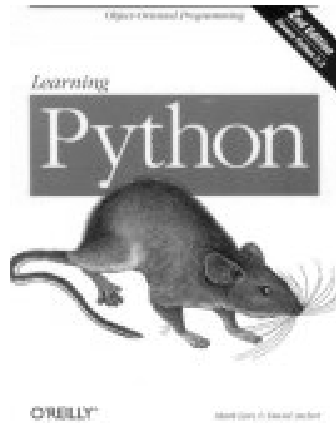
**pass** is een leeg statement als opvuller

```
if tekort or telang or ar[0]=="#":
    pass
else:
    print "af te handelen!"
```

Figuur 6.

### Student-notities

Bij deze cursus leveren wij — naast dit werkboek — de boeken "*Learning Python*" (van de auteurs M. Lutz en D. Ascher) en "*Python in a Nutshell*" (van de auteur A. Martelli) waarin deze materie verder is uitgewerkt.



## Exceptions - introductie

Python is streng

```
>>> i=5
>>> j=0
>>> print i/j
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
ZeroDivisionError: integer division or
modulo by zero

>>> a="hoi"
>>> print a[17]
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
IndexError: string index out of range
```

Het programma stopt...

Oplossing:

- Vooraf testen?  
Nee! Dan wordt er twee keer getest!
- Achteraf fout afvangen?  
Ja!

Het motto in Python: EAFP

It is easier to ask for Forgiveness than to ask for  
Permission

**Figuur 7.**

