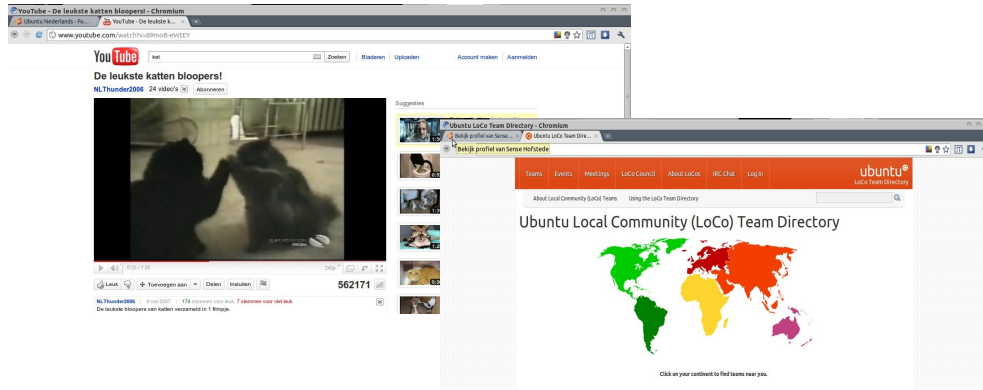


Les 1

Jullie willen allemaal Python leren, dus het leek mij een goed idee om eerst eens te laten zien wat je nou eigenlijk met python kunt maken.

Zo draaien bijv. verschillende websites op de achtergrond op python:

<http://youtube.com/> en <http://loco.ubuntu.com/> bijvoorbeeld zijn een webapplicatie gebaseerd op python.



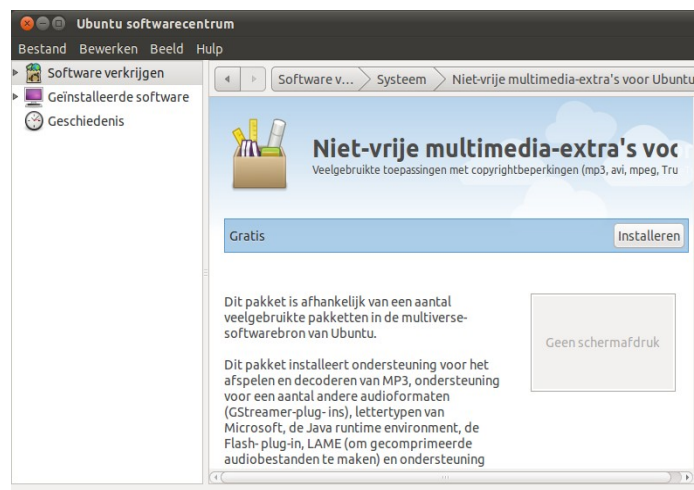
De eigenaar van YouTube is Google, een bedrijf dat erg veel doet met Python.

Daarnaast is het mogelijk om applicaties te maken die draaien in een terminal.

Een mooi voorbeeld is **Yum**, de tegenhanger van apt-get dat jullie misschien wel kennen van Ubuntu en Fedora.

```
root@XVirus:~ - Shell No. 3 - Konsole
Session Edit View Bookmarks Settings Help
[cvirus@XVirus cvirus]$ su -l root
Password:
[root@XVirus root]# yum install kplayer
Gathering header information file(s) from server(s)
Server: Fedora Core 2 - i386 - Base
Server: Fedora.us Extras (Stable)
Server: Fedora.us Extras (Testing)
Server: Fedora.us Extras (Unstable)
Server: Livna.org - Fedora Compatible Packages (stable)
Server: Livna.org - Fedora Compatible Packages (testing)
Server: Livna.org - Fedora Compatible Packages (unstable)
Server: macromedia.mplug.org - Flash Plugin
Server: Fedora Core 2 - i386 - Unreleased Updates
Finding updated packages
Downloading needed headers
Resolving dependencies
Dependencies resolved
I will do the following:
[install: kplayer 0.4.0-0.lvn.1.2.i386]
Is this ok [y/N]: y
```

Ook desktop applicaties zijn in python te schrijven. Een voorbeeld uit Ubuntu is het **Software centrum**.



Ten slotte wilde ik nog de mogelijkheid noemen om games te maken in Python, bekend is **Frets on Fire**.



Ik hoop dat jullie nu een indruk hebben, hoewel dit slechts het topje van de ijsberg is. Python wordt, zeker in de Open-Source-wereld, erg veel gebruikt.

Nu jullie dus een idee hebben wat je kan met python, wil ik het graag even hebben over wat een computerprogramma eigenlijk is. Een computerprogramma is een lijst met instructies die aangeven wat de computer moet doen. De taal waarin die lijst is beschreven, heet de programmeertaal.

Vraag: *Wat is python nou eigenlijk een scripttaal of iets anders?*

Antwoord: *Technisch gezien valt python onder de scripting talen. maar dat zegt in de praktijk niets over de mogelijkheden van een taal.*

In een computerprogramma zit vaak een structuur die je altijd terug kan herkennen:

- Invoer (van de gebruiker, uit een bestand, of uit een andere bron)
- Verwerken (de invoer wordt omgezet naar iets anders, iets nuttigs.)
- Uitvoer (het resultaat van het verwerken wordt weergegeven, bijvoorbeeld op een monitor, maar bijv. een server zou het naar de webbrowser van de gebruiker sturen)

Ten slotte moeten jullie nog één ding weten over programmeertalen op dit moment: Je mag geen enkele fout maken als je erin schrijft, want als dat gebeurt, zal je programma crashen of iets anders doen dan verwacht. Computers zijn namelijk dom, ze begrijpen niet wat ze doen.

OK, hoog tijd om je eerste programma te schrijven.

Om te beginnen, open je **Terminalvenster**.

In Ubuntu gaat dat via:

Toepassingen > Hulpmiddelen > Terminalvenster.

Typ nu in je Terminalvenster 'python' in, zonder aanhalingstekens, en druk je op enter. als het goed is zie je wat als:

```
Python 2.6.6 (r266:84292, Sep 15 2010, 16:22:56)
```

```
[GCC 4.4.5] on linux2
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

Je ziet nu >>> staan, met daarachter de cursor.

Typ nu de volgende tekst in, en druk daarna op enter:

```
print "Hallo Ubuntero's!"
```

Nogmaals, maak geen typ fouten, want dan krijg je mogelijk een foutmelding, of doet het programma niet wat je verwacht.

We gaan nu een wat ingewikkelder programma maken namelijk.
Typ de volgende zinnen over (of kopiëren/plakken):

```
naam = raw_input("Wat is je naam? ")
```

Na de eerste zin krijg je de vraag te zien "Wat is je naam?". Typ hier je naam in en druk weer op enter. Typ daarna de volgende zin en geef enter.

```
print naam, "is goed op weg met het leren van python!"
```

Je krijgt nu het volgende te zien.

(naam die je hebt ingetypt) is goed op weg met het leren van python!

Ik zal nu eens uitleggen wat hier eigenlijk gebeurd.

Het stukje tekst `raw_input("Wat is je naam? ")` doet twee dingen.

1) Het laat de tekst 'Wat is je naam?' zien.

2) Het vraagt de gebruiker om invoer, en nadat je die invoer hebt gegeven en op enter hebt gedrukt, gaat het programma verder.

Dan wordt nl. het stukje `'naam ='` uitgevoerd.

Dat stukje betekend iets als 'sla het resultaat van wat achter '=' staat op onder de naam 'naam'!.

Daarna is de regel daar volledig uitgevoerd, en gaat python verder naar de volgende regel.

Daar staat:

```
print naam, "is goed op weg met het leren van python!"
```

Daar staat dan het commando `'print'`.

Print doet niets anders dan stukjes informatie op het scherm laten zien.

Het eerste wat `'print'` laat zien is wat is opgeslagen met naam.

* onder de naam 'naam'.

```
print naam, "is goed op weg met het leren van python!"
```

Eerst wordt op het scherm de tekst bewaart in 'naam' gezet

de komma verteld python dat het verder moet kijken naar het tweede stuk tekst.

* vertelt.

Dat stuk tekst wordt vervolgens ook op het scherm weergeven

het resultaat is één zin op je beeldscherm:

jouw naam is goed op weg met het leren van python!

Extra opmerkingen.

Die komma zorgt ook dat de 2 stukken gescheiden worden door een spatie.

print naam + " is goed op weg met het leren van python!" (zou ook kunnen, met een + ipv een , - het verschil is dat bij een komma er een extra spatie tussen komt)

Dan wordt het tijd om te zien dat python een beetje kan rekenen.

type het volgende in:

```
6+5
```

Je ziet dan dat python meteen met het antwoord komt, 11.

Python kan nog meer, bijvoorbeeld vermenigvuldigen.

```
5 * 80
```

Geeft 400 terug.

Ook aftrekken gaat zonder problemen: $6-3 = 3$

Machtsverheffen is ook een mogelijkheid, $2^{**} 3$ (2^3) geeft natuurlijk 8

(De eerste vorm is wat python begrijpt, de tweede is de mathematische notatie)

Goed, we hebben alleen nog niet gedeeld.

Ook dat kan python:

```
20 / 10
```

Geeft netjes 2

Het gaat fout bij $60/10$, je zult zien dat python 60 geeft.

Python heeft een reden om hier een geheel getal te geven, we komen er later op terug.

Je kan ook cijfers in een variabele zetten overigens.

(Zo'n ding als we eerder zagen bij naam= ...)

in dit geval doen we:

```
getal = 6
```

We kunnen vervolgens allerlei dingen doen met 'getal'

Bijvoorbeeld:

```
print getal + 5
```

Print is momenteel niet vereist, maar later wel als we straks uit de 'console' gaan. Dat is zeg maar wat je krijgt als je 'python' intikt in een Terminalvenster zoals je allemaal hebt gedaan.

Goed, als je dit begrijpen, kunnen we weer een stapje verder gaan.

Probeer deze code eens, de hele tijd één keer op enter drukken nadat je iets hebt ingevoerd.

Ook tabs zijn belangrijk bij het overtypen/kopieren:

```
answer = input("Wat is 2+2?")
```

Geef hier de uitkomst van de som.

```
if answer == 4:  
    print "Je had het goed."  
else:  
    print "Je had het fout."
```

Dit ziet er in het Terminalvenster als volgt uit.

```
>>> answer = input("Wat is 2+2?")
Wat is 2+2?4
>>> if answer == 4:
...     print "Je had het goed."
... else:
...     print "Je had het fout."
...
Je had het goed.
```

Je ziet hier dat twee lijnen zijn ingesprongen met behulp van de tab-toets. Het is belangrijk in Python om dat goed over te nemen.

```
answer = input("Wat is 2+2?")
if answer == 4:
    print "Je had het goed."
else:
    print "Je had het fout."
```

Wat hier gebeurt:

Er wordt aan de gebruiker gevraagd om een *cijfer* op te geven. Het resultaat wordt opgeslagen in de variabele 'answer'

Vervolgens kijkt Python met behulp van 'if' of het antwoord gelijk is aan 4.

Als dat zo is, wordt de code die ingesprongen is uitgevoerd.

Dat is in dit geval dat de tekst '*je had het goed*' op het scherm wordt weergegeven.

Als je iets anders hebt ingevuld bij de input(), dan wordt alles uitgevoerd wat ingesprongen onder '*else:*' staat.

Dat is in dit geval de tekst:

je had het fout weergegeven met behulp van '*print*'.