



Ontwikkelomgeving

Installeren/configureren ontwikkelomgeving

- Maak een directory/map aan. (bijvoorbeeld in Dropbox-omgeving).
- Gebruik de naam: *Project-website*.
- Open een terminal venster (Windows: cmd | Osx: terminal.app)
en ga naar de map *Project-website*

Virtualenv

Dit is een tool om een geïsoleerde Python omgeving te creëren. Geïsoleerde of wel ingesloten met alle benodigde applicaties voor de ontwikkeling van een website voor een klant.

Terminal of cmd

Ga naar de aangemaakte directory: *Project-website* en voer de volgende commando's in.

Het dollarteken(\$) voor de commando's geeft de prompt aan, en hoeft je dus niet in te voeren.

Dit promptteken verschilt per besturingssysteem. Bij OSX, Linux en Git-Bash op windows is dit een dollarteken.

Bij Windows CMD en Powershell is dat het pad naar de map waar je op dat moment bent, gevolgd door een 'kleiner-dan'-teken (>).

We houden in de tutorials het dollarteken (\$) aan om de prompt aan te geven.

Installeren virtualenv-software

```
$ pip install virtualenv
```

Aanmaken virtualenv -omgeving

De virtuele omgeving moet eerst worden aangemaakt, of wel de basis-configuratie moet worden geïnstalleerd.

Dit doe je door het volgende commando:

```
$ virtualenv project-venv
```

Activeren virtualenv

De virtuele omgeving moet vervolgens geactiveerd worden, of wel worden aangezet. Dit doe je door het volgende commando:

op **OSX**:

```
$ source ./project-venv/bin/activate
```

op **Windows met CMD of Powershell**:

```
$ .\project-venv\Scripts\activate
```



op Windows met Git Bash

```
$ source ./project-venv/Scripts/activate
```

Zie de verandering in de terminal. Voor de prompt komt :

op **OSX** komt voor de prompt (project-venv) in beeld:

```
(project-venv) Project-website $
```

op **Windows** komt dit boven of voor de prompt (PROJE~1):

```
(PROJE~1) [pad naar Project-website]>
```

Hieraan kun je zien dat de virtuele omgeving is geactiveerd.

Python PIP

Met het commando pip installeer je de gewenste Python software pakketten. Pip is het package-management systeem voor Python. Pip commando's ga je ook toepassen bij de installatie van je Python Django omgeving.

Ze hebben de volgende vorm:

```
pip install some-package-name  
pip uninstall some-package-name
```

Installeer nu met de volgende commando's de gewenste Pythonapplicaties in de virtualenv-omgeving:

```
(project-venv) ...$ pip install Django  
(project-venv) ...$ pip install psycopg2-binary  
(project-venv) ...$ pip install pillow
```

Upgrade geïnstalleerde packages

Om eerder geïnstalleerde Python packages te updaten of upgraden naar een nieuwe of hogere versie gebruik je het volgende commando:

```
pip install --upgrade some-package-name
```

Overzicht met pip geïnstalleerde applicaties

De meest handige functie is de mogelijkheid om met pip een volledige lijst met geïnstalleerde packages te generen met volgende commando. Deze lijst kun je later gebruiken om de benodigde pakketten te installeren als je je project wilt installeren op een andere plek.

```
(project-venv) ...$ pip freeze > requirements.txt
```



Het volgende commando zorgt er voor dat het bestand requirements.txt wordt gebruikt voor het installeren van de benodigde Python packages. Dit is handig als je je project naar een andere plek hebt verplaatst.

```
pip install -r requirements.txt
```

Inhoud van requirements.txt weergeven

op **OSX**:

```
(project-venv)...$ nano requirements.txt
```

op **Windows**:

```
(project-venv)...$ notepad requirements.txt
```