



### Model/View/Template principe: het database-model

Een model is de omschrijving van de te gebruiken database-tabel. Het bevat de essentiële velden en het gedrag van de data die wordt opgeslagen. Elk model is gekoppeld aan een enkele tabel van de database.

- Elk model is een Python class van de subclass [django.db.models.Model](#).
- Elk attribuut van het model is een veld van de database-tabel.
- Django zorgt voor een geautomatiseerde gegenereerde toegang tot de database-API

### Model/View/Template principe: Een model van Werken maken

**Daarvoor gaan we het bestand: /werken/models.py aanpassen.**

Open een terminal venster (Windows: cmd | OSX: terminal.app):

```
(project-venv) ...$ cd Project-website/portfolio-project/werken/
```

En open daarin **models.py**

Voeg het volgende (database)model aan het bestand toe:

```
from django.db import models
from django.contrib.auth.models import User

class Werken(models.Model):
    title = models.CharField(max_length=200)
    body = models.TextField()
    url = models.TextField()
    pub_date = models.DateTimeField()
    image = models.ImageField(upload_to='media/images/')
    icon = models.ImageField(upload_to='media/icon/')
    votes_total = models.IntegerField(default=1)
    hunter = models.ForeignKey(User, on_delete=models.CASCADE)
```

### Werken aan het Admin-panel toevoegen

Voeg de volgende definities toe. Deze zorgen er voor dat de veldnamen worden getoond in het admin-panel.

```
# admin panel
def __str__(self):
    return self.title

def summary(self):
    return self.body[:200]
```



De functie `pub_date_pretty` zorgt er voor dat de datum straks netjes getoond kan worden op een pagina.

Voeg deze ook toe aan `models.py`

```
# pub_date_pretty
def pub_date_pretty(self):
    return self.pub_date.strftime('%b %e %Y')
```

Let bij het invoegen op het juist inspringen! Dit is cruciaal bij Python!