

# **django** **unchained**

- from a novice point of view -

# What is django?

- high-level Python Web framework
  - $\leq 1.11$ : Python 2.7 and 3.x
  - $\geq 2.0$ : only 3.5+
- not a CMS, though can be used to create CMS
- rapid development
- clean, pragmatic design
- licensed under **BSD**
- homepage

<https://www.djangoproject.com/>

# Features

- authentication
- database abstraction layer
  - sqlite, mysql, postgresql, ...
  - migrations when changing data models
- secure
  - no XSS, SQL Injection, clickjacking, etc
- content administration
- template support (jinja)
  - eg add bootstrap for mobile/desktop interface
- gazillions of plugins
  - <https://djangopackages.org/>

# Creating a project

- excellent tutorial for newbies - like me

<https://docs.djangoproject.com/en/stable/intro/tutorial01/>

- create your project

```
django-admin startproject hampug
```

- fire up test server

```
python manage.py runserver
```

# Create first app

- execute django command  
`python manage.py startapp gallery`
- update *settings.py*

```
INSTALLED_APPS = (  
    'gallery',  
)
```

- add a view (gallery/views.py)

```
from django.http import HttpResponse  
  
def index(request):  
    return HttpResponse("Hello World!")
```

# Create first app (2)

- add URL for view (gallery/urls.py)

```
from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^$', views.index, name='index'),
]
```

- add URL to project (hampug/urls.py)

```
from django.conf.urls import include, url
from django.contrib import admin

urlpatterns = [
    url(r'^gallery/', include('gallery.urls')),
    url(r'^admin/', admin.site.urls),
]
```

# Bootstrap

- create directory for static content  
hampug/static
- download and unzip bootstrap into *static*  
<http://getbootstrap.com/>
- let django know about static content (settings.py)

```
STATIC_URL = '/static/'  
STATICFILES_DIRS = [  
    'hampug/static/',  
]
```

# Bootstrap (2)

- create directory for templates  
hampug/templates
- create *base.html* template

```
<html>
<head>
  <title>{{ title }}</title>
  <link rel="stylesheet" href="/static/css/bootstrap.min.css" type="text/css"/>
  {% block head %}
  {% endblock %}
</head>
<body>
  {% block content %}this is the content{% endblock %}

  <script src="/static/js/jquery.min.js"></script>
  <script src="/static/js/bootstrap.min.js"></script>
</body>
</html>
```



# Bootstrap (3)

- add *base\_navbar.html* template

```
{% extends "base.html" %}
{% block content %}
{% block top %}

<nav class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target="#navbar" aria-expanded="false" aria-controls="navbar">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="/">Gallery</a>
    </div>
  </div>
</nav>
{% endblock %}

<div class="container">
{% block main %}
{% endblock %}
</div>

{% endblock %}
```

# Bootstrap (4)

- Create page for “Hello World!”

gallery/templates/gallery/hello.html

```
{% extends "base_navbar.html" %}
{% block main %}

<h3>Hello World</h3>
<div class="panel panel-default">
  <div class="panel-heading">
    <h3 class="panel-title">Details</h3>
  </div>
  <div class="panel-body">
    {{ text }}
  </div>
</div>

{% endblock %}
```

# Bootstrap (5)

- update view (gallery/views.py)

```
from django.template import loader

from django.http import HttpResponse

def index(request):
    template = loader.get_template('gallery/hello.html')
    context = {
        'text': "More details on the world"
    }
    return HttpResponse(template.render(context, request))
```

# Bootstrap (final)

- update *settings.py*

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [  
            'hampug/templates'  
        ],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    },  
]
```

# Mobile friendly website

