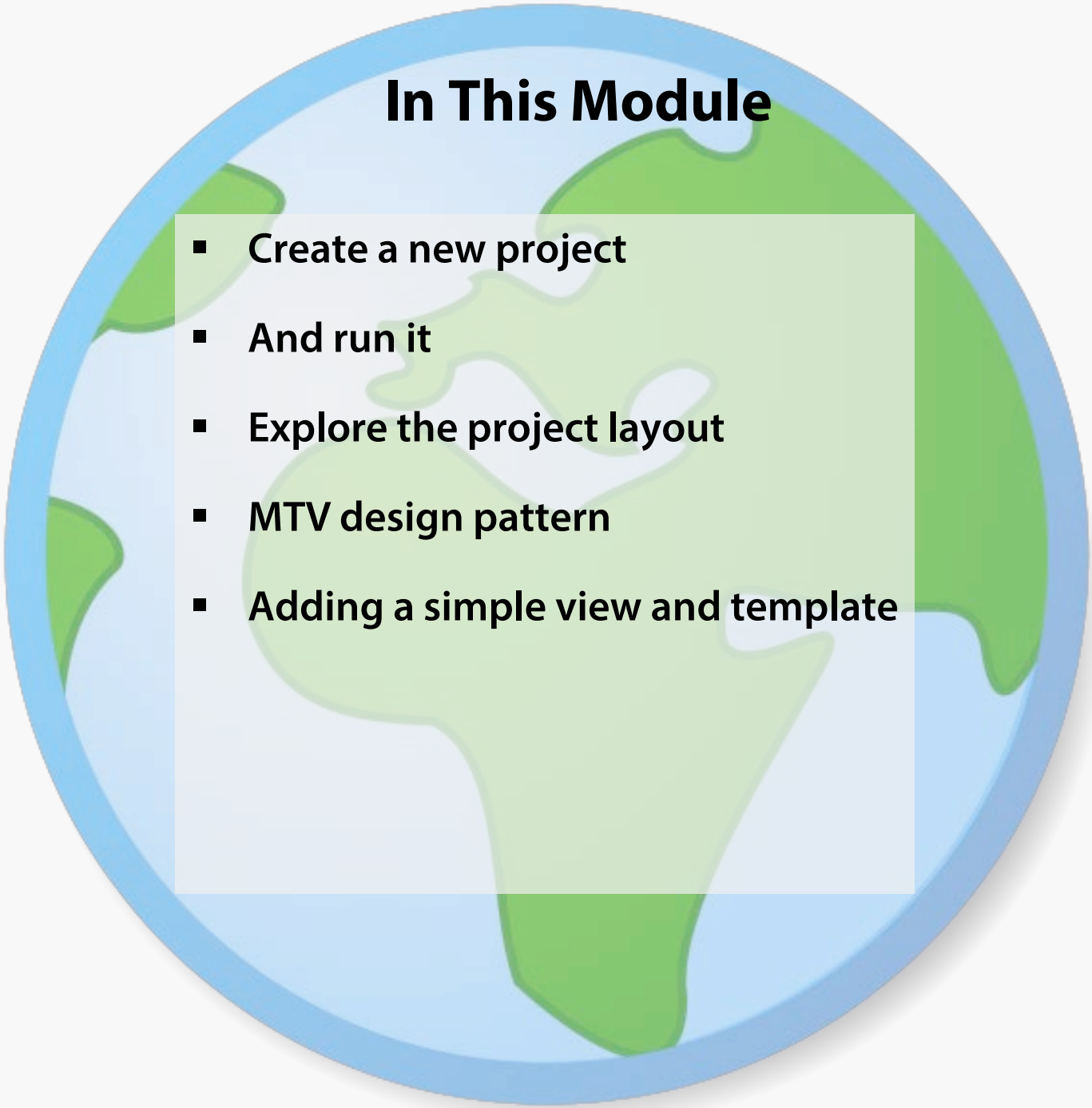
Starting a Django Project

Reindert-Jan Ekker
nl.linkedin.com/in/rjekker/
@rjekker

pluralsight
hardcore dev and IT training



In This Module

- Create a new project
- And run it
- Explore the project layout
- MTV design pattern
- Adding a simple view and template

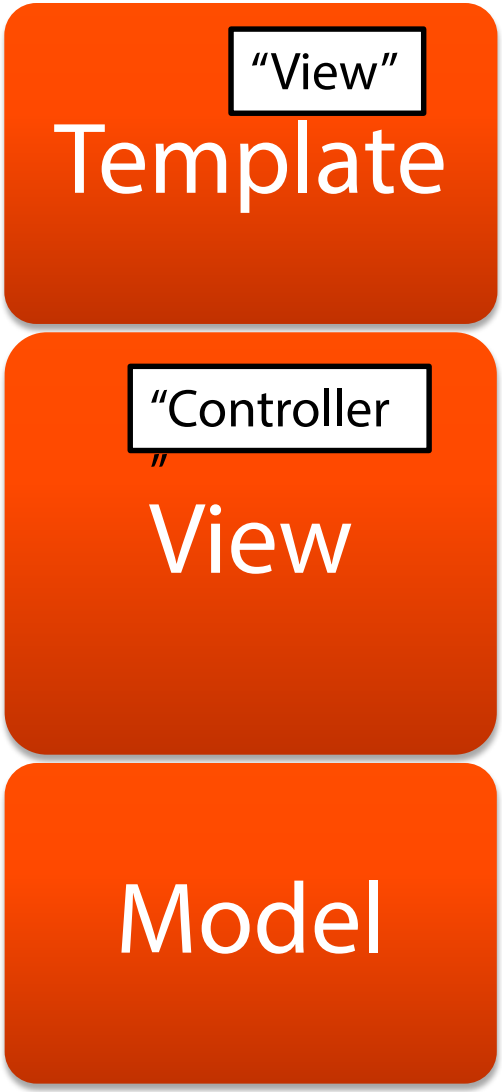
Creating a New Project

- Don't forget to activate your virtualenv
- `django-admin.py startproject projectname`
- On Windows:
  - `python virtualenv\Scripts\django-admin.py startproject projectname`

Running the Project

- **To run the new project**
  - `cd boardgames`
  - `python manage.py runserver`
- **The development server**
  - Will reload python code automatically
  - Don't use this as a production server!
- **Run on a different port**
  - `python manage.py runserver 4321`

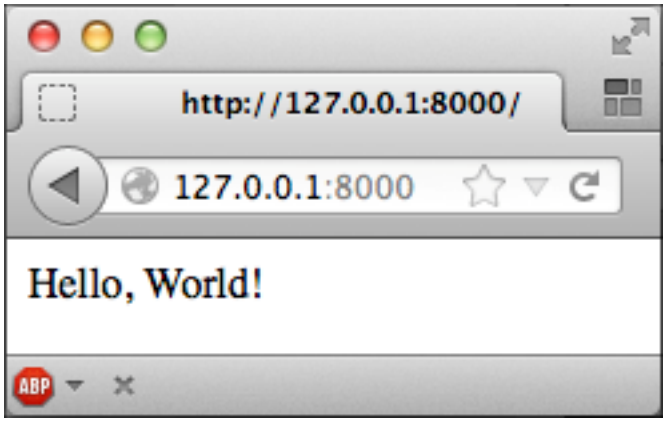
Model View Template



- Generates HTML
- Presentation logic only

- Takes HTTP request and returns response
- May use model to retrieve/store data
- May call a template to present data

- Represents your data
- Each model class represents a database table




```
GET / HTTP/1.1
```

runserver

How URLs Are Mapped

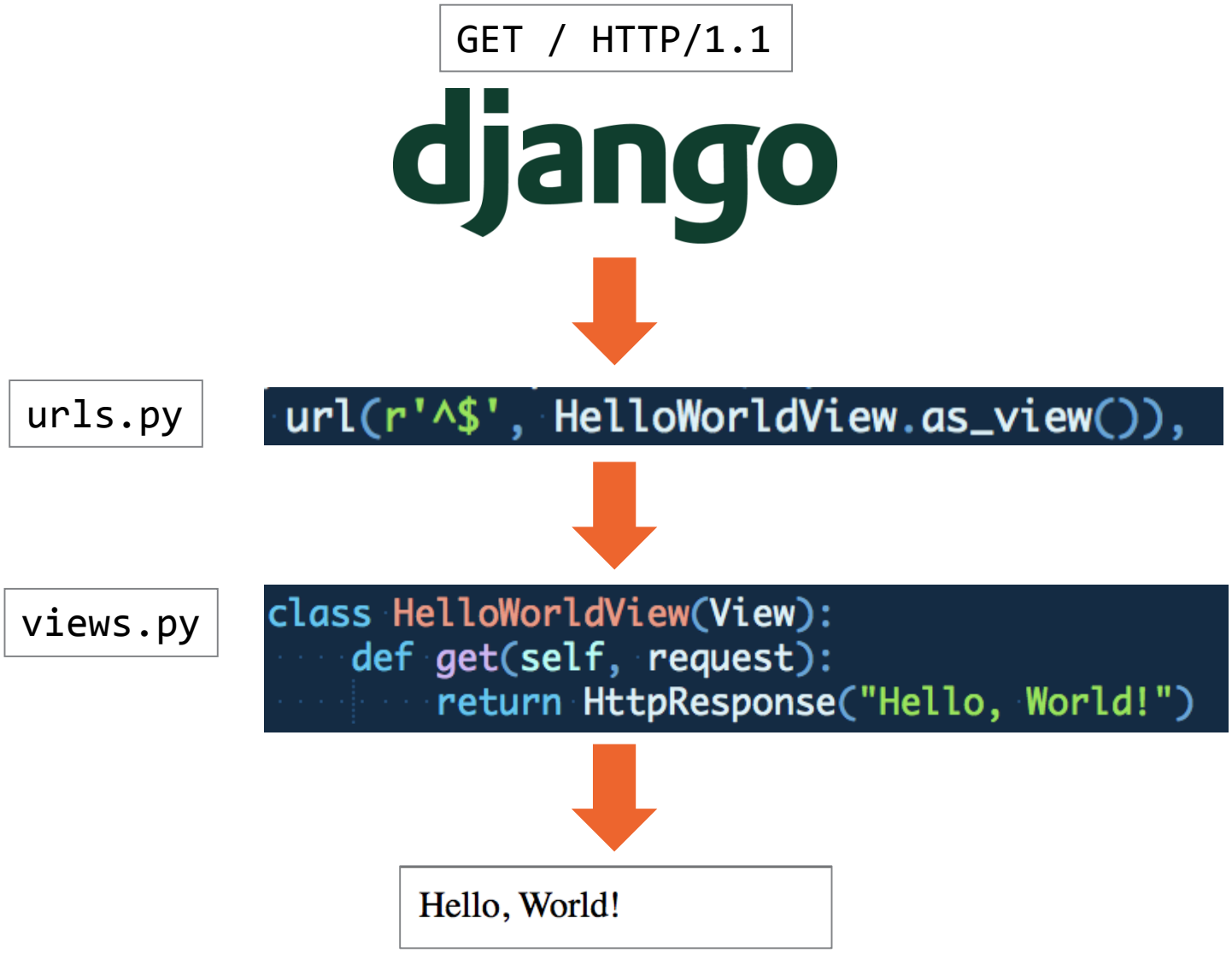
- When an HTTP request comes in for some URL
- Django looks in urls.py
  - Finds a `urlpatterns` variable
  - This holds a list of url mappings
- Tries to find a pattern that matches the URL

```
urlpatterns = patterns('',
    url(r'^$', HelloWorldView.as_view()),
    url(r'^admin/', include(admin.site.urls)),
)
```

Regular Expressions

- **URLs are matched by regular expressions**
  - Very powerful way to match strings
  - ^ Denotes start of string
  - $ Denotes end of string
- **To match the URL "hello"**
  - `r'^hello$'`
  - The "r" before the first quote means "raw string" notation
- **To match any URL starting with "prefix/"**
  - `r'^prefix/.*$'`
- **More about python regex: http://goo.gl/5uJsfy**

MTV in Action: URLs and Views



Django Views

- **Django Views are what other MVC frameworks call “Controllers”**
- **A view is a callable**
  - That takes a request object
  - And returns a response object
  - Can be a function
- **We will focus on Class-based Views**
  - Allow reuse of code by inheritance
  - Allow use Django’s predefined generic views

Class-based Views

- **Inherit from `django.views.generic.base.View`**
- **HTTP method is mapped to method name (usually `get()`)**
  - It receives a `django.http.HttpRequest`
  - Should return a `django.http.HttpResponse`
- **HTTP Response contains:**
  - A status code (200 OK, 404 Not Found, etc.)
  - Headers (Mime type, Date, Cookies, Caching info, etc.)
  - The actual document (HTML, Image, Javascript, CSS, etc.)

Returning a HTTP Response

- **Returning a HTML page**
  - Status code 200
  - Response contains HTML
  - Simply return a new HttpResponse instance

```
class HelloWorldView(View):
    def get(self, request):
        return HttpResponse("Hello, World!")
```

- **But we don't want our Views to contain presentation logic**
  - Move generation of HTML to a template instead

Summary

- **Creating a project**
- **Running it**
- **Model-template-view**
- **Mapping URLS**
- **Views**
  - Class-based views
  - Generic Views
- **Templates**