

django

no, not the movie

(but first)

- regular expressions!
- fun! :D
- (just kidding no fun)

regular expressions!

- strings with special syntax
- not programming language specific
- represent patterns

regular expressions

- represent patterns
- do things follow the pattern?
- extract data encoded in strings

basic examples

proper noun

- **example:** Kanye
- **pattern:** [1 uppercase][1 or more lowercase]
- **regex:** [A-Z][a-z]+

basic examples

phone number

- **example:** (847) 404-8821
- **pattern:** ([3 numbers]) [3 numbers]-[4 numbers]
- **regex:** `\(\d{3}\) \d{3}-\d{4}`
- (parens must be escaped)

name that regex! :D

`[A-Z][a-z]+ [A-Z]\. [A-Z][a-z]+`

`name!`

`/watch?v=([a-zA-Z\d\-_]{11})`

youtube url!

$\backslash d\{1,3\}\backslash.\backslash d\{1,3\}\backslash.\backslash d\{1,3\}\backslash.\backslash d\{1,3\}$

ip address!

(kinda)

urls

- example: <http://mygreatblog.com/post/87/>
- encoded data:
 - type of content
 - which blog post to show

urls

- **example:** <http://mygreatblog.com/post/87/>
- **pattern:** /post/[1 or more digits]/
- **regex:** /post/\d+/
(\d+)
- **extracting post id:** /post/(\d+)/

so how does traditional
web development look?

php

request: /blog/authors/

- execute code in `/blog/authors/index.php`
 - db calls
 - any logic
 - generating HTML

why django?

- separates db, logic, and HTML
 - model, view, template (MVT)
- non-technical users can upload content
- get to use Python <3

model view template?

- Like MVC but for Django
- **model**: database-connected objects
- **view**: functions, business logic
- **template**: HTML + extra Django syntax
 - rendered into HTML

structure

```
└─ mygreatblog
    └─ manage.py # General utility script
    └─ mygreatblog
        └─ settings.py # project settings
        └─ urls.py # Maps URL to view
    └─ blogger # app
        └─ models.py # models
        └─ views.py # views
        └─ templates # templates
            └─ template1.html
            └─ template2.html
```

how are web pages made?

- when a mommy Django and a daddy Django love each other very much...

how are web pages made?

example url: <http://mygreatblog.com/post/12/>

1. **urls.py**: */post/12/*
 - URL in form */post/<post_id>/*
2. **views.py**: `get_post(request, post_id)`
 - find Post model with id `post_id`
3. **template**: `single_post.html`
 - Fill in template with data from the Post model

models

- "normal" Python objects
- define fields
- can be loaded from or saved to db

querying!

```
from blogger.models import Author, BlogPost
```

```
# all authors
```

```
all_authors = Author.objects.all()
```

```
# Author with first name "Josh"
```

```
a = Author.objects.get(first="Josh")
```

```
# posts by Josh
```

```
josh_posts = BlogPost.objects.filter(author=a)
```

```
print josh_post.title
```

show me all ur posts?

1. define URL + view:

- `/posts/ -> blogger.views.all_posts`

2. write template

- `blogger/templates/all_blog_posts.html`

show me all ur posts?

blog/urls.py

```
urlpatterns = patterns('blogger.views',  
    url(r'^posts/$', 'all_posts'),  
    ...  
)
```

show me all ur posts?

blogger/views.py

```
from django.shortcuts import render
from models import BlogPost

def all_posts(request):
    posts = BlogPost.objects.all()
    data = {'posts': posts}
    return render(
        request, 'all_blog_posts.html', data)
```


show me all ur posts?

blogger/templates/all_blog_posts.html

```
{% for post in posts %}
  <p>
    <a href="{{ post.get_absolute_url }}">
      {{ post.title }}
    </a>
  </p>
  <p>
    By
    <a href="{{ post.author.get_absolute_url }}">
      {{ post.author }}
    </a>
  </p>
{% endfor %}
```

all of josh's posts pls

1. define URL + view:

- `/authors/\d+/ -> blogger.views.author`

2. write template

- `blogger/templates/author.html`

all of josh's posts pls

blog/urls.py

```
urlpatterns = patterns('blogger.views',  
    url(r'^authors/(?P<author_id>\d+)/$',  
        'all_posts'),  
    ...  
)
```

all of josh's posts pls

blogger/views.py

```
from django.shortcuts import get_object_or_404, render
from models import Author, BlogPost
```

```
def author(request, author_id):
    author = get_object_or_404(Author, id=author_id)
    posts = BlogPost.objects.filter(author=author)

    data = {
        'author': author,
        'posts': posts
    }

    return render(request, 'author.html', data)
```

show me all ur posts?

blogger/templates/all_blog_posts.html

```
{% for post in posts %}
  <p>
    <a href="{{ post.get_absolute_url }}">
      {{ post.title }}
    </a>
  </p>
  <p>
    By
    <a href="{{ post.author.get_absolute_url }}">
      {{ post.author }}
    </a>
  </p>
{% endfor %}
```

moar pages!

- a single blog post?
- a list of all authors?