

Django Tutorial

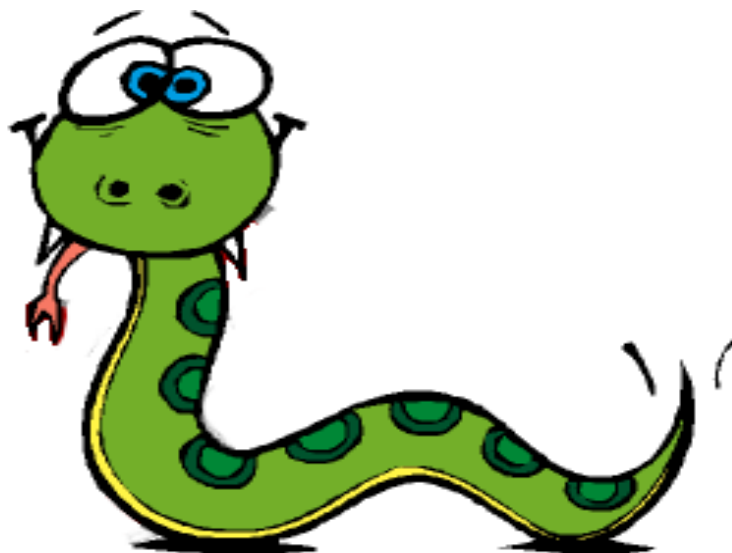
- 什么是 Django
- Django MVC
- Django Model 常见关系
- Django view 的使用
- 如何简单调试
- 正则表达式基础

你会得到：简陋的，只有你才能用的微博网站一枚～

Code is Poetry

什么是 Django

Python

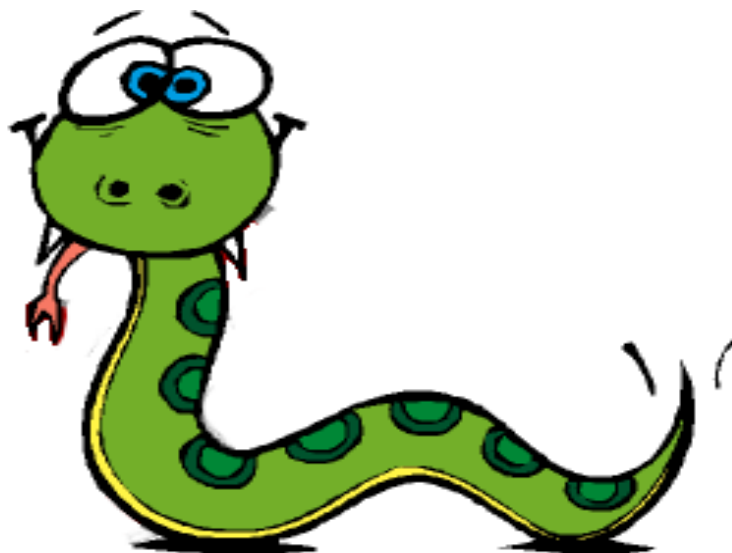


什么是 Django

Django



Python

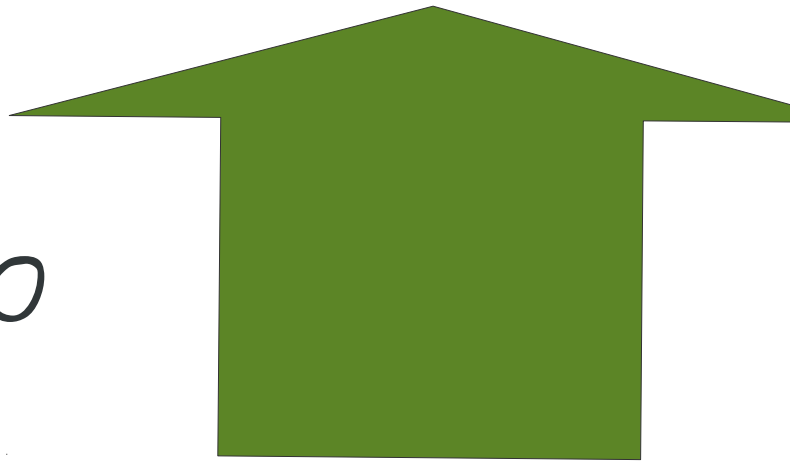


什么是 Django

You



Django



Python



什么是 Django

You

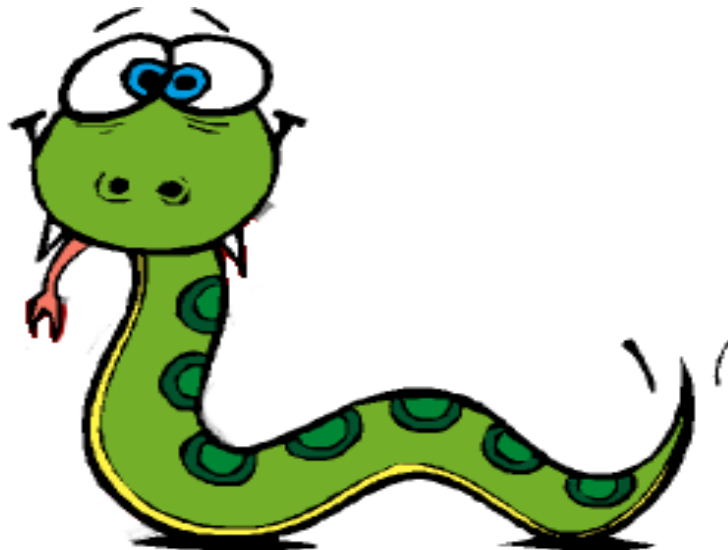
装修工 A



Django



Python



Python 相关工具的概念

virtualenv

虚拟环境程序

制造梦境用哦 ~



Python 相关工具的概念

virtualenv

虚拟环境程序

pip

Python 程序管理器



帮主人管好
程序哦~



Code Time! [1]

```
virtualenv example-env
```

```
# 建立新的虚拟环境
```

```
source example-env/bin/activate
```

```
# 使用新的虚拟环境
```

```
pip install Django
```

```
# 安装 Django
```

```
django-admin startproject weibo
```

```
# 建立新的项目
```

```
cd weibo
```


等等！

什么是 Project ???

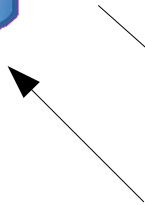
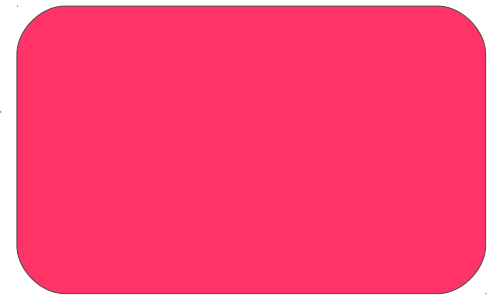
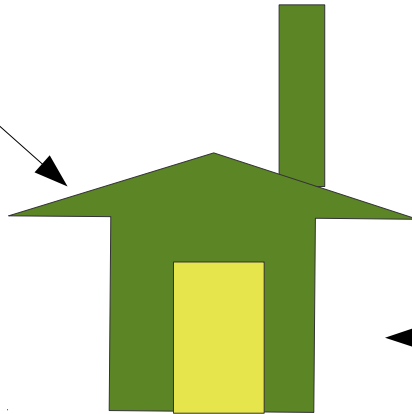
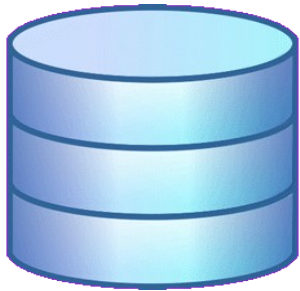
就是这个项目，名字你爱叫什么就是什么

Django M.V.C.

Model

Controller

View



Django M.V.C.

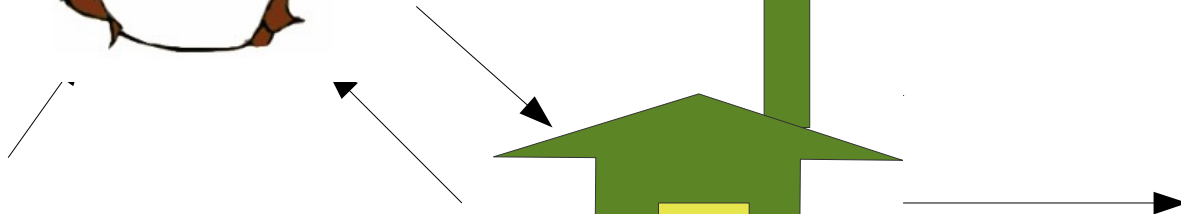
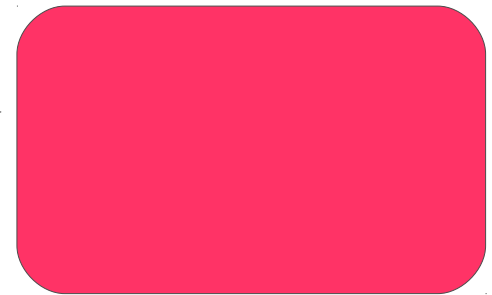
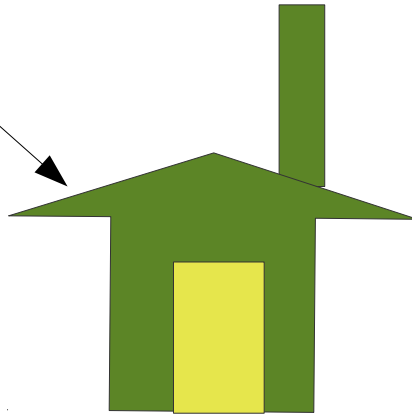
Model

Controller

View



证件照

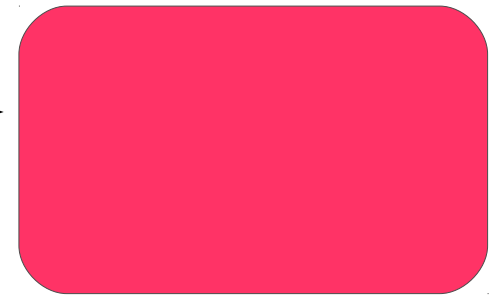
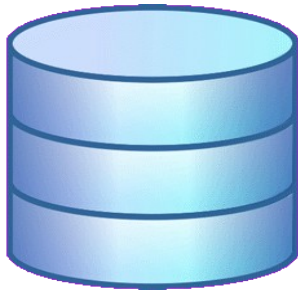


Django M.V.C.

Model

Controller

View



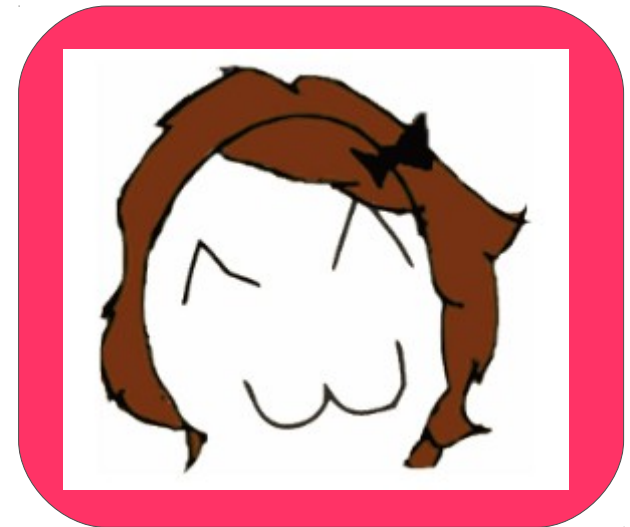
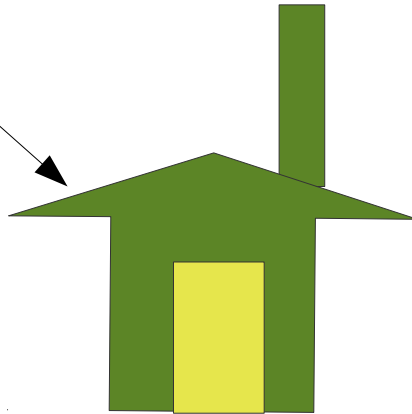
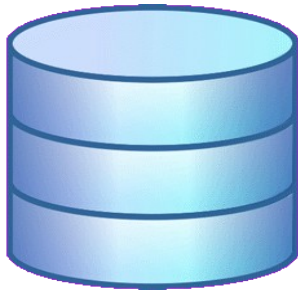
颠倒是非中

Django M.V.C.

Model

Controller

View



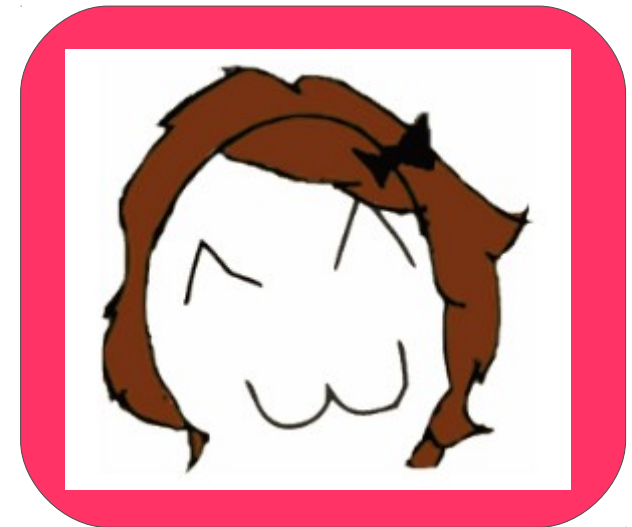
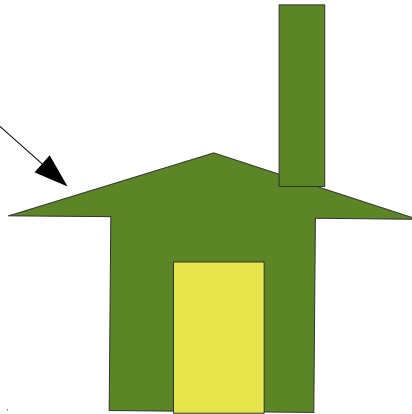
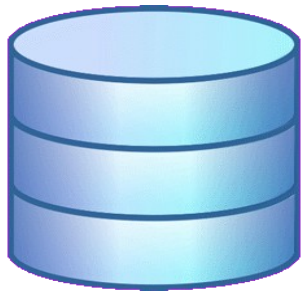
继续美化

Django M.V.C.

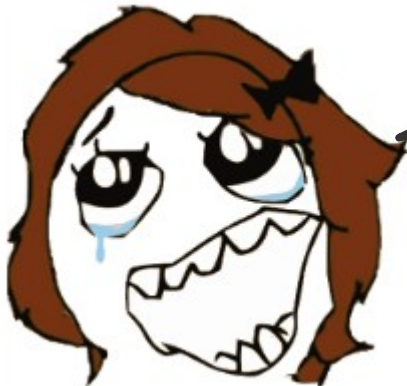
Model

Controller

View



好人啊！



Code Time! [2]

```
django-admin startapp tweet
```

```
django-admin startapp user
```

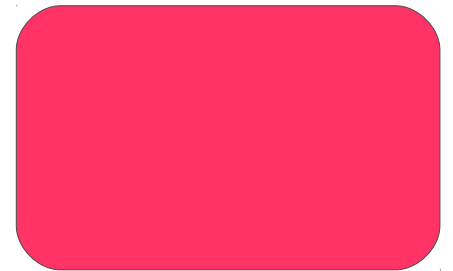
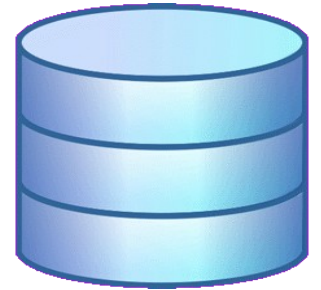
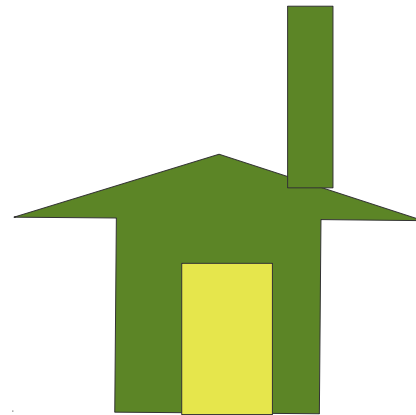
等等！

Startapp??? **app** 是什么???

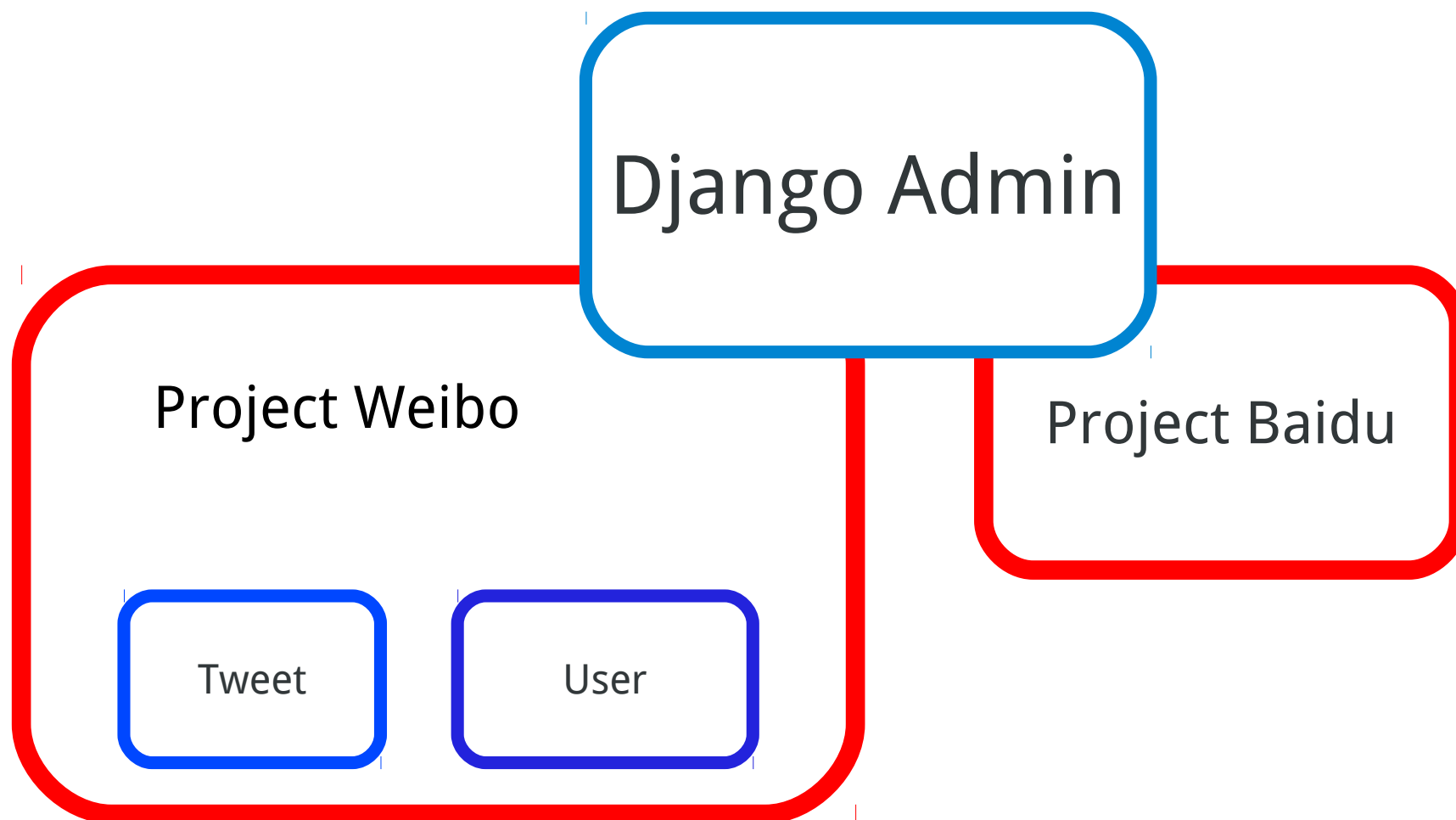
App 就是：

- 定义 model
- 控制相关 view
- 控制相关 url
- 控制相关后台管理

的**应用程序**



app 与 project 的关系



Django Model

任何事物

都可以转换成

Model（数据模型）

并确定这些事物之间的

关系

这就是 Model 的工作

Django Model

这就是“建模”

任何事物

都可以转换成

Model（数据模型）

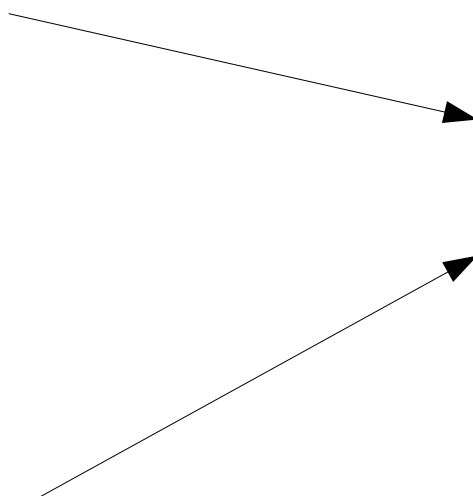
并确定这些事物之间的

关系

这就是 Model 的工作

Django Model Relation

ForeignKey



真人

Django Model Relation

ForeignKey

tweet(推文 = 微博)

person(用户)

有了麦当劳生活 ~
好滋味 ~

麻辣烫还挺好吃的



Code Time! [3]



```
// write into person/models.py
```

```
class Person( models.Model ):
```

```
    name = models.CharField(max_length=8,
```

```
                           verbose_name=u' 用户名 '
                           )
```

```
    def __unicode__(self):
```

```
        return self.name
```

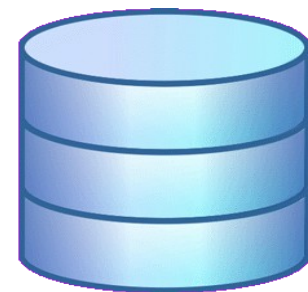
等等！

CharField 是什么？？

```
models.CharField(max_length=8,  
                  verbose_name=u'用户名')
```

Field 指实际存放的数据项：

一段文字 —————> CharField



Code Time! [4]

有了麦当劳生活
好滋味 ~

```
// Into tweet/models.py
from django.db import models
from person.models import Person
class Tweet( models.Model ):

    user = models.ForeignKey(Person,
                              related_name = 'tweet',
                              )

    content = models.CharField(max_length=140,
                               verbose_name=u'推文',
                               )

    def __unicode__(self):
        return u"%s的推文: %s" % (self.user.name, self.content)
```


等等！

related_name 是什么？？

```
user = models.ForeignKey(Person,  
                           related_name = 'tweet')
```

等等！

`related_name` 是什么？？



的 推

`Tweet.objects.filter(user=`



等等！

related_name 是什么??



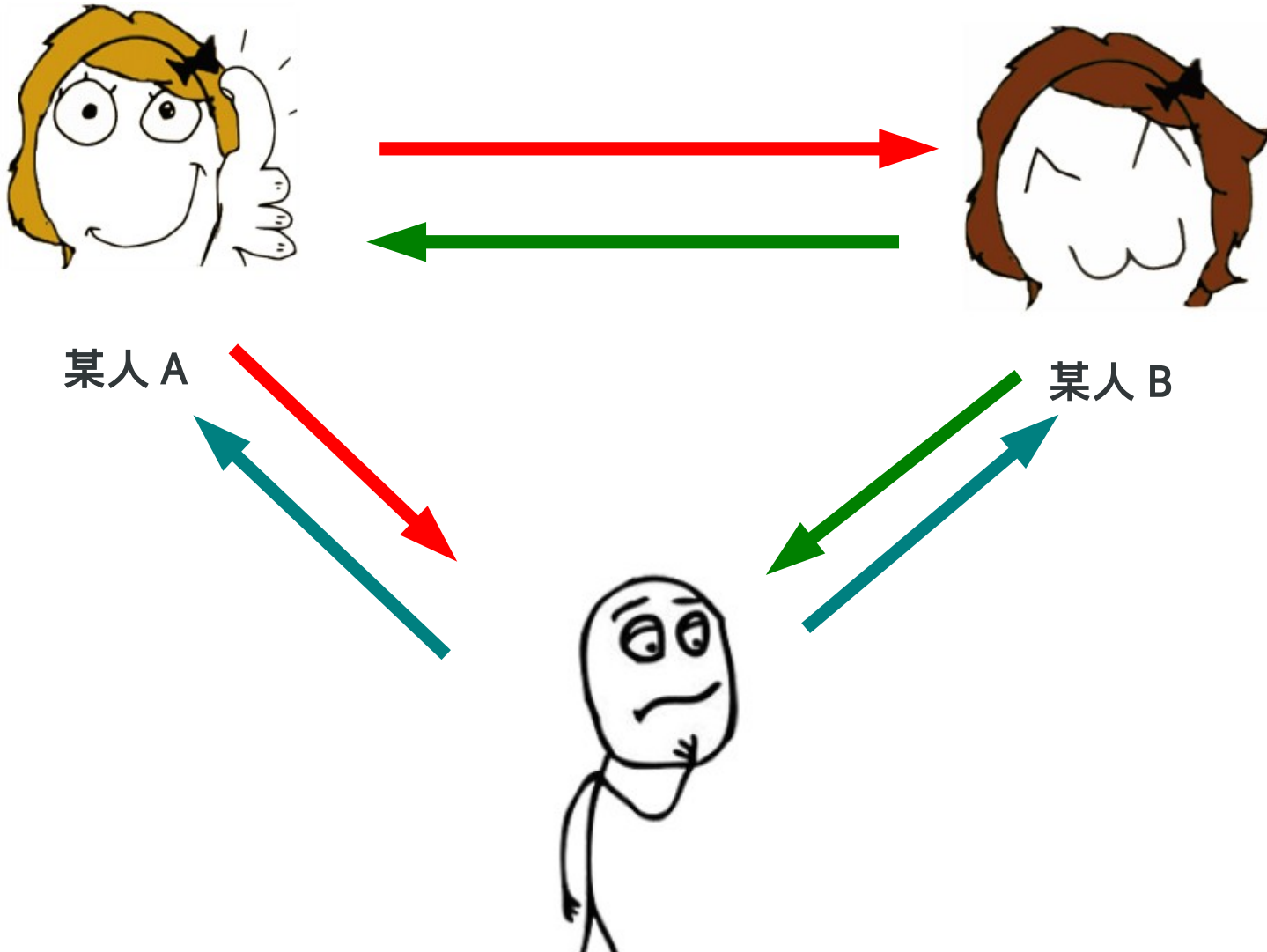
的 推



.tweet

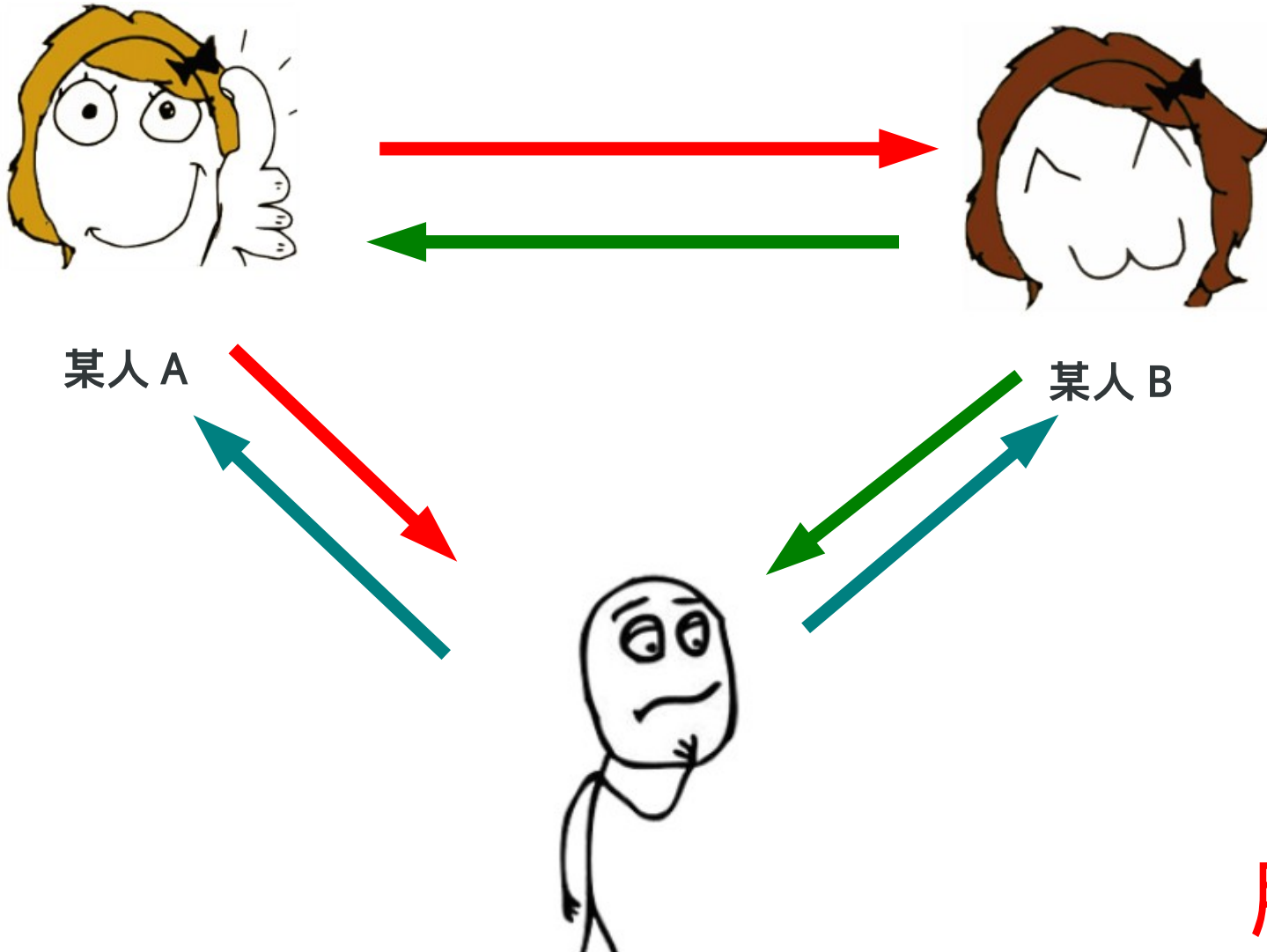
Django Model Relation

ManyToMany(默认)



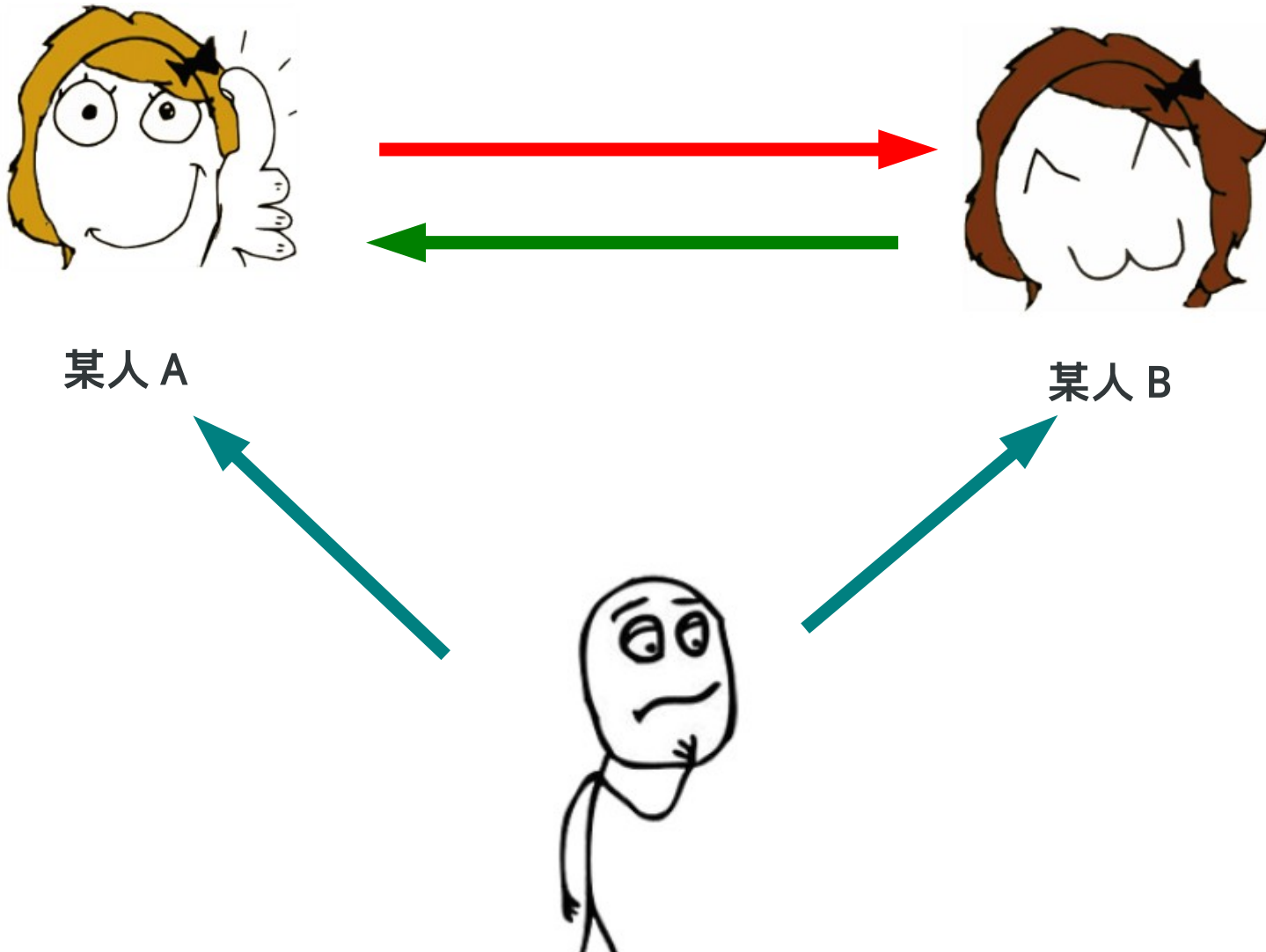
Django Model Relation

ManyToMany(默认)



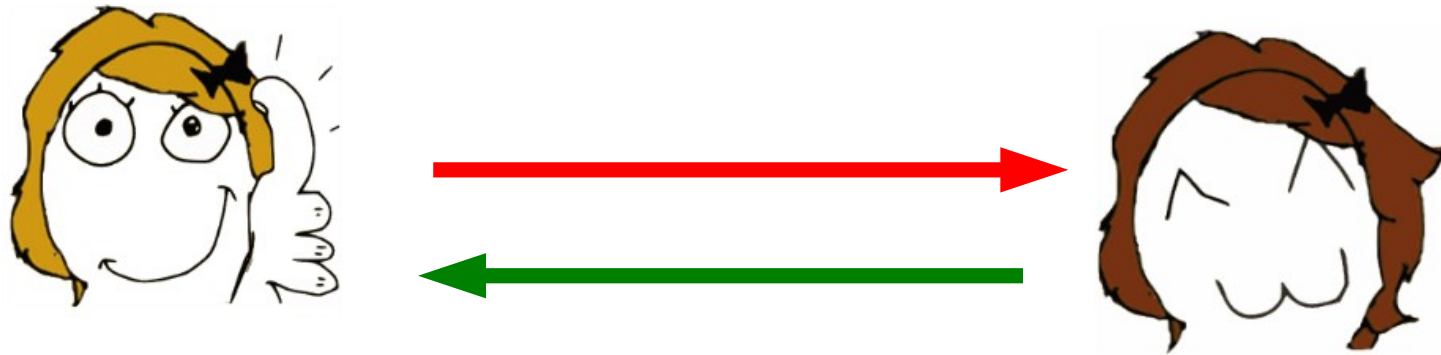
Django Model Relation

ManyToMany (symmetrical=False)



Django Model Relation

ManyToMany (symmetrical=False)



某人 A

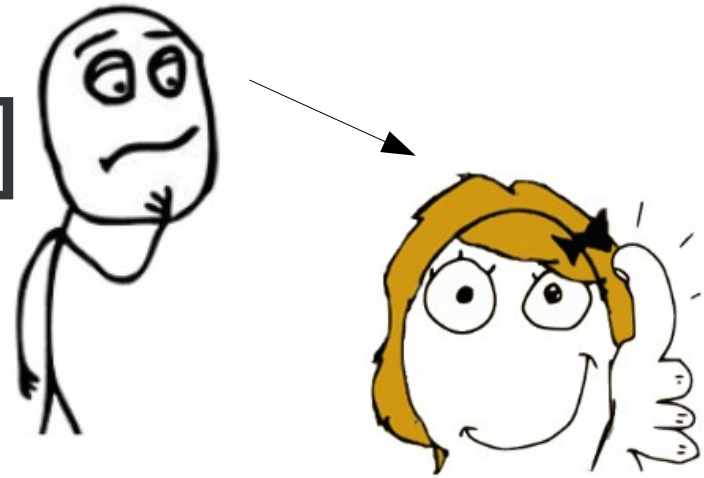
某人 B

求粉啊！
Fo 我啊！
这样就免费啊！



微博类
Twitter 类

Code Time! [5]



```
// in person/models.py
```

```
follow = models.ManyToManyField(  
    'self',  
    related_name = 'follower',  
    symmetrical=False)
```


Model 终于讲完了……

接下来，内部 App：

URL （路径）编写

View （视图）编写

Model 终于讲完了，但是……

在这之前：

什么是漂亮的 URL？

- /person?name=KFC
- /person?userid=1090124
- /person/1090124
- /person/KFC

URL 设计

正则表达式……正则……新手的噩梦

怎么在以下字符串里找： **Aragon**
Aragon is a 7-years old company
which has about 60 employees right
now. Now, I'm showing this very
presentation to those **Aragon's**
employees who are interested in
Django.

URL 设计

正则表达式……正则……

怎么在以下字符串里找：数字

Aragon is a 7-years old company which has about 60 employees right now. Now, I'm showing this very presentation to those Aragon's employees who are interested in Django.

URL 设计

正则表达式……正则……

$\left. \begin{matrix} 60 \\ 7 \end{matrix} \right\} \backslash d^+$

$\backslash d$ **digit** : 0,1,2,3...9
+ 必须匹配一个及其以上

URL 设计

正则表达式……正则…

怎么在以下字符串里找： **S 结尾的单词**

Aragon **is** a 7-**years** old company
which **has** about 60 **employees** right
now. Now, I'm showing this very
presentation to those Aragon's
employees who are interested in
Django.

URL 设计

正则表达式……正则…

Employees
Years
Is
has

} \w+\s

\w word : 字母 A-Z, a-z

\s space: 代表一个空格

URL 设计

正则表达式……正则

怎么在以下字符串里找：**带 S 结尾的词**

Aragon **is** a 7-**years** old company
which **has** about 60 **employees** right
now. Now, I'm showing this very
presentation to those **Aragon's**
employees who are interested in
Django.

URL 设计

正则表达式……正则

Employees
Years
Is
has
Aragon's

} [\w']+\s\s

[] 占位符

回到 URL 设计

怎么截取这些 **Name**?

- /person/**KFC**
- /person/**R2D2**



URL 设计

正则表达式

KFC
R2D2 } (?P<name>[\w\d]+)

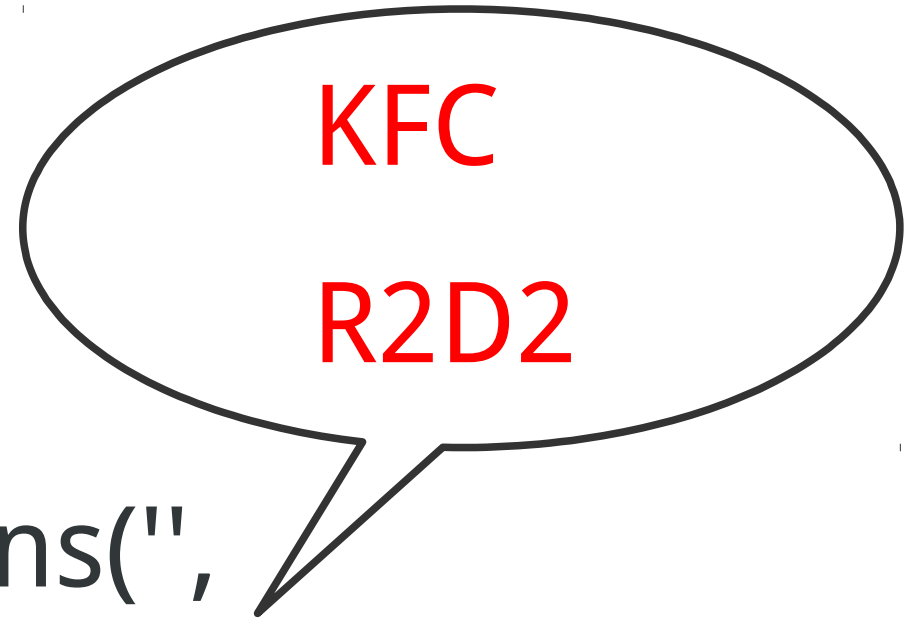
()

?P< 名字 > 正则

分组
命名

Code Time! [6]

- //in weibo/person/urls.py



```
urlpatterns = patterns('',  
    url(r'^(?P<name>[\w\d]+)',  
    show, name="show_person"),  
)
```

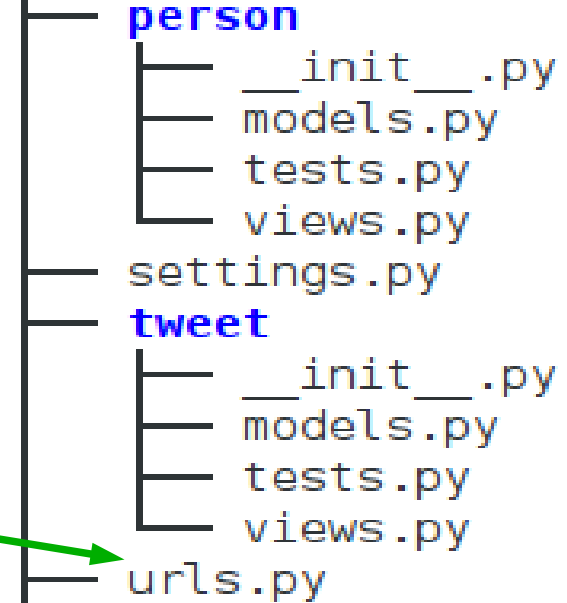
Code Time! [7]

- //in weibo/urls.py

```
url(r'^person/', include('person.urls')),
```

正则

告诉 Django,
符合这段正则的 URL
都让这个 'person.urls' 包了



- /person/KFC

- /person/R2D2

.....Zzz.....

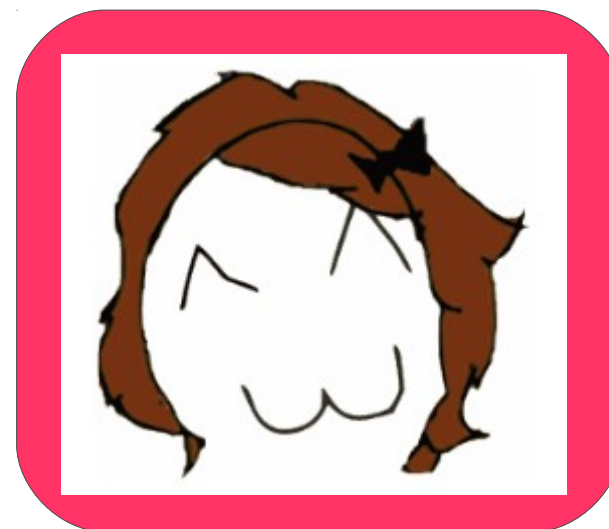
刚才我们：

造了条通向用户的路



那怎么显示用户呢？？

views(视图) 登场！



Code Time![8]

```
// in person/views.py  
from django.http import HttpResponse  
  
from person.models import Person  
  
def show( request, name ):  
    who = Person.objects.filter(name=name)  
    return HttpResponse(who)
```


等等！

HttpResponse ? ? ?

简易 HTTP 请求响应

等等！

HttpResponse ???

简易 HTTP 请求响应

我找 R2D2 ！

哎～来了！

/person/R2D2

那怎么显示 R2D2 的推？



.tweet

那怎么显示 R2D2 的推？



.tweet

建模建得真好！

Code Time![9]

```
// in person/views.py  
  
from django.http import HttpResponse  
  
from person.models import Person  
  
def show( request, name ):  
    who = Person.objects.get(name=name)  
    tweets = who.tweet.all()[0].content  
    #import ipdb; ipdb.set_trace()  
    return HttpResponse("%s:%s" % (who, tweets))
```

大功告成？

其实……

还需要管理界面（ admin ）……

不过咱就略过吧……

请注意各个 app 下的

admin.py 文件

跑起来 ~ Code Time[10]

// in settings.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': 'local.db',  
        'USER': '',  
        'PASSWORD': '',  
        'HOST': '',  
        'PORT': '',  
    }  
}  
  
INSTALLED_APPS = (  
    'django.contrib.admin',  
    'tweet',  
    'person',  
)
```

跑起来 ~ Code Time[11]

`python manage.py syncdb` # 同步数据库

`python manage.py runserver` # 运行服务器

简单调试

忘记 print!!!!

简单调试

忘记 print!!!!

```
import ipdb; ipdb.set_trace()
```



简单调试

忘记 print!!!!

```
import ipdb; ipdb.set_trace()
```

L	Line
C	Continue
R	Return

终于讲完了，但是……

怎样的代码算美??

以下是某段搜索代码……

终于讲完了，但是……

怎样的代码算美??

```
if cid != '':
    try:
        cid = int(cid)
        preclient = preclient.filter(category=cid)
    except:
        pass
#####
req = request.GET.get('q', '')

if req != '':
    if req.isdigit():
        preclient = preclient.filter
(phone__contains=int(req))
    else:
        preclient = preclient.filter
(username__contains=req)
```

终于讲完了，但是……

怎样的代码算美？？

```
q_dict = reduce(lambda x,y: x&y,  
                [ Q(**{'%s__slug_name__in' %  
taxonomy:request.GET.getlist(taxonomy)}  
                  ) for taxonomy in request.GET.keys() if  
taxonomy in SEARCHABLE_LIST ]  
                )
```

简直是我的化身



终于讲完了

怎样的代码算美??

- D.R.Y. 原则
- 命名清晰
- 函数变量少
- 鲁棒?

努力地偷懒!

推荐书目

- Head First Python (中文版)

<http://book.douban.com/subject/10561367/>

- Python 编程 (第三版)

<http://book.douban.com/subject/1915868/>

