

WEB APPLICATION DEVELOPMENT WITH PYTHON & DJANGO

PyCon Pakistan 2017

Ali Raza Bhayani

CTO at BitsWits (Pvt) Ltd.
CEO and Founder at DataLysis.io
CEO and Founder at LearningByDoing.io

Full Stack Developer, Open Source Enthusiast,
Hacker, Enabler, Do-Tank, Autodidact, Yogi and
an avid Reader.

Web: www.alirazabhayani.com
Web: www.LearningByDoing.io
Web: www.DataLysis.io

Email: alirazabhayani@gmail.com
Twitter: [@alirazabhayani](https://twitter.com/alirazabhayani)
Facebook:
<https://www.facebook.com/alirazabhayani>



- Widely used general-purpose, high-level programming language,
Design philosophy emphasizes code readability, Express concepts in fewer lines of code

PYTHON

Organizations using Python

- Yahoo
- Google
- IBM
- Nokia
- Blender 3D
- NASA
 - <https://wiki.python.org/moin/OrganizationsUsingPython>
- Instagram
- Bitbucket
- Disqus

PYTHON 101

Python 101

Python Basics

Whitespace Is Important

```
In [1]: listOfNumbers = [1, 2, 3, 4, 5, 6]
```

```
for number in listOfNumbers:  
    print number,  
    if (number % 2 == 0):  
        print "is even"  
    else:  
        print "is odd"  
  
print "All done."
```

```
1 is odd  
2 is even  
3 is odd  
4 is even  
5 is odd  
6 is even  
All done.
```

Python 101

Importing Modules

In [3]: `import numpy as np`

```
A = np.random.normal(25.0, 5.0, 10)
print A
```

```
[ 23.21559595  26.23663702  14.22030905  13.91716711  17.16732856
 26.53544908  20.48855379  32.01235201  28.08437063  28.74018971]
```

Python 101

Lists

```
In [2]: x = [1, 2, 3, 4, 5, 6]
        print len(x)
```

```
6
```

```
In [4]: x[:3]
```

```
Out[4]: [1, 2, 3]
```

```
In [5]: x[3:]
```

```
Out[5]: [4, 5, 6]
```

```
In [6]: x[-2:]
```

```
Out[6]: [5, 6]
```

```
In [7]: x.extend([7,8])
        x
```

```
Out[7]: [1, 2, 3, 4, 5, 6, 7, 8]
```

```
In [8]: x.append(9)
        x
```

```
Out[8]: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
In [9]: y = [10, 11, 12]
        listOfLists = [x, y]
        listOfLists
```

```
Out[9]: [[1, 2, 3, 4, 5, 6, 7, 8, 9], [10, 11, 12]]
```

Python 101

Tuples

```
In [13]: #Tuples are just immutable lists. Use () instead of []
x = (1, 2, 3)
len(x)
```

Out[13]: 3

```
In [14]: y = (4, 5, 6)
y[2]
```

Out[14]: 6

```
In [15]: listOfTuples = [x, y]
listOfTuples
```

Out[15]: [(1, 2, 3), (4, 5, 6)]

```
In [5]: (age, income) = "32,120000".split(',')
print age
print income
```

```
32
120000
```


Python 101

Dictionaries

```
In [17]: # Like a map or hash table in other languages
captains = {}
captains["Enterprise"] = "Kirk"
captains["Enterprise D"] = "Picard"
captains["Deep Space Nine"] = "Sisko"
captains["Voyager"] = "Janeway"

print captains["Voyager"]
```

Janeway

```
In [18]: print captains.get("Enterprise")
```

Kirk

```
In [19]: print captains.get("NX-01")
```

None

```
In [20]: for ship in captains:
          print ship + ": " + captains[ship]
```

Voyager: Janeway
Deep Space Nine: Sisko
Enterprise D: Picard
Enterprise: Kirk

Python 101

Functions

```
In [21]: def SquareIt(x):  
         return x * x  
  
         print SquareIt(2)
```

4

```
In [22]: #You can pass functions around as parameters  
         def DoSomething(f, x):  
             return f(x)  
  
         print DoSomething(SquareIt, 3)
```

9

```
In [23]: #Lambda functions let you inline simple functions  
         print DoSomething(lambda x: x * x * x, 3)
```

27

Python 101

Boolean Expressions

```
In [24]: print 1 == 3
```

```
False
```

```
In [25]: print (True or False)
```

```
True
```

```
In [26]: print 1 is 3
```

```
False
```

```
In [27]: if 1 is 3:  
         print "How did that happen?"  
         elif 1 > 3:  
             print "Yikes"  
         else:  
             print "All is well with the world"
```

```
All is well with the world
```

Python 101

Looping

```
In [28]: for x in range(10):  
         print x,
```

```
0 1 2 3 4 5 6 7 8 9
```

```
In [29]: for x in range(10):  
         if (x is 1):  
             continue  
         if (x > 5):  
             break  
         print x,
```

```
0 2 3 4 5
```

```
In [30]: x = 0  
         while (x < 10):  
             print x,  
             x += 1
```

```
0 1 2 3 4 5 6 7 8 9
```

**AT TIMES,
UNEXPECTED
THINGS
HAPPEN
IN THE
WORLD OF
OPEN SOURCE
PROGRAMMING**





DJANGO FOR FAST WAY OF GOING FROM DEVELOPMENT
TO PRODUCTION



SO LET'S START OUR DJANGO QUEST!

DJANGO BASICS

- **M** stands for “Model,” the data access layer. This layer contains anything and everything about the data: how to access it, how to validate it, which behaviors it has, and the relationships between the data.
- **T** stands for “Template,” the presentation layer. This layer contains presentation-related decisions: how something should be displayed on a Web page or other type of document.
- **V** stands for “View,” the business logic layer. This layer contains the logic that accesses the model and defers to the appropriate template(s). You can think of it as the bridge between models and templates

DJANGO FEATURES

- Object Relational Mapper – ORM
-
- MVC (MVT) Architecture
-
- Focuses on automating as much as possible and adhering to the DRY principle
-
- Template System
-
- Out of the box customizable Admin Interface, makes CRUD easy
-
- Built-in light weight Web Server
-

DJANGO FEATURES

- Elegant URL design
-
- Custom Middleware
-
- Authentication / Authorization
-
- Internationalization support
-
- Cache framework, with multiple cache mechanisms
-
- Fast Development
-
- Free, and Great Documentation.



**DJANGO
COMES
IN FOR
HELP.**

**IN WEB
APPLICATION
DEVELOPMENT,
MANAGING
REQUESTS,
URLS,
ROUTING,
TEMPLATES,
DATABASE
CAN BE
OVERWHELMING.**



BACKEND (DJANGO) CODE STRUCTURE

HTTP Request



Middleware

Router

Decorators

Controller (a.k.a View)

Template / Serializer

Middleware



Model (ORM)



HTTP Response

Installing & Configuring Django Components

- Django Pre-Requisites
- Downloading & Installing Django
- Choosing a Database
- Creating a New Project

Installing & Configuring Django Components

- Python PIP for installing Python packages (<http://www.pip-installer.org/en/latest/installing.html>)
- pip install Django OR <https://www.djangoproject.com/download/> - python setup.py install
- pip install pysqlite
- Add Python and Django to env path
- PYTHONPATH D:\Python27
- Path D:\Python27; D:\Python27\Lib\site-packages; D:\Python27\Lib\site-packages\django\bin;
- Testing installation
- shell> import django; django.VERSION;

STARTING A NEW PROJECT

```
django-admin startproject myproject
```

```
|-- myproject/          <-- django project folder
|   |-- myproject/
|   |   |-- __init__.py
|   |   |-- settings.py
|   |   |-- urls.py
|   |   |-- wsgi.py
|   +-- manage.py
```

DJANGO PROJECT STRUCTURE

- Our initial project structure is composed of five files:
- `manage.py`: a shortcut to use the `django-admin` command-line utility. It's used to run management commands related to our project. We will use it to run the development server, run tests, create migrations and much more.
- `__init__.py`: this empty file tells Python that this folder is a Python package.
- `settings.py`: this file contains all the project's configuration. We will refer to this file all the time!

DJANGO PROJECT STRUCTURE

- `urls.py`: this file is responsible for mapping the routes and paths in our project. For example, if you want to show something in the URL `/about/`, you have to map it here first.
- `wsgi.py`: this file is a simple gateway interface used for deployment. You don't have to bother about it. Just let it be for now.
-

Configuring Django Components

- Choosing a Database
- Creating a New Project
-

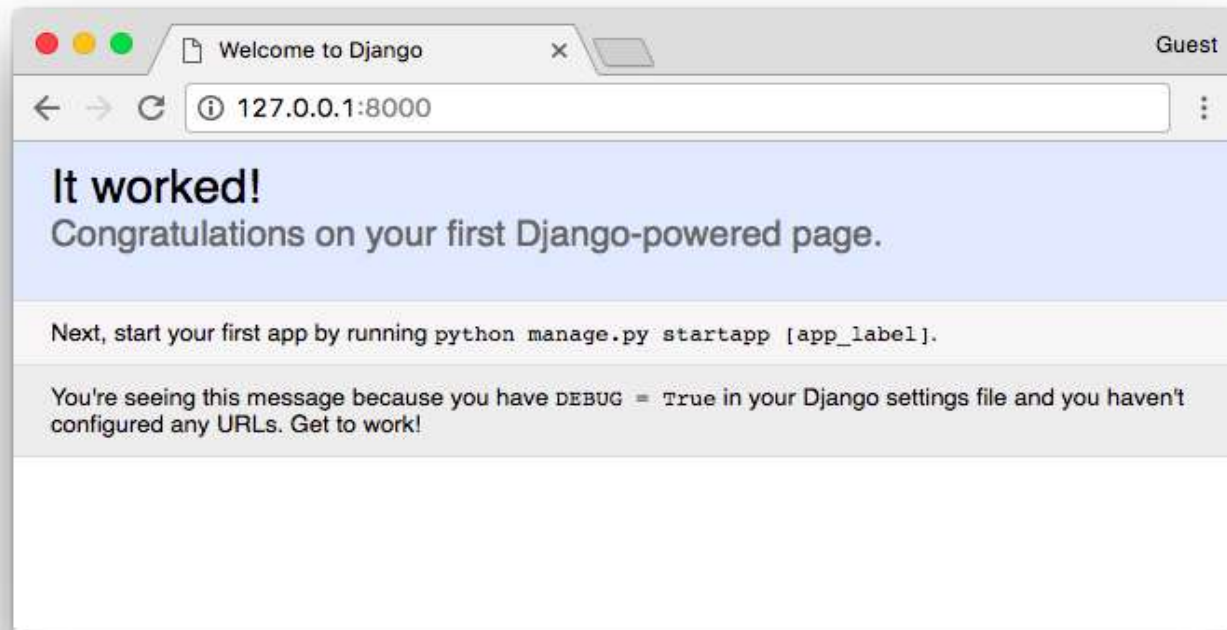
STARTING A NEW PROJECT

```
django-admin startproject myproject
```

```
|-- myproject/          <-- django project folder
|   |-- myproject/
|   |   |-- __init__.py
|   |   |-- settings.py
|   |   |-- urls.py
|   |   |-- wsgi.py
|   +-- manage.py
```

DJANGO DEVELOPMENT SERVER

```
python manage.py runserver
```



DJANGO PROJECT STRUCTURE

- App: is a Web application that does something. An app usually is composed of a set of models (database tables), views, templates, tests.
- Project: is a collection of configurations and apps. One project can be composed of multiple apps, or a single app.
-

Generating Simple Django Views

- About View Functions
- Using Django's HttpResponse Class
- Understanding HttpRequest Objects
- Using QueryDict Objects

Django Templates

- Template Fundamentals
- Creating Template Objects
- Loading Template Files
- Filling in Template Content (Context Objects)
- Template Filters
- Template Tags
- More on For Loops
- Easy Rendering of Templates
- Request Context Processors

BUILDING A REAL WORLD APP

- Creating Models
- Create Router URLs
- Our First View
- Create a Template
- Adding Dynamic Data
- Add a ListView Page
- Add a Base Template
-

A photograph of a railway track curving through a misty, autumnal landscape. The tracks are made of dark metal rails on wooden sleepers, leading the eye into the distance. The sky is a soft, hazy orange and yellow, suggesting a sunrise or sunset. Trees with some autumn-colored foliage are visible on either side of the tracks. The overall mood is contemplative and serene.

**WHERE
TO GO
FROM HERE**