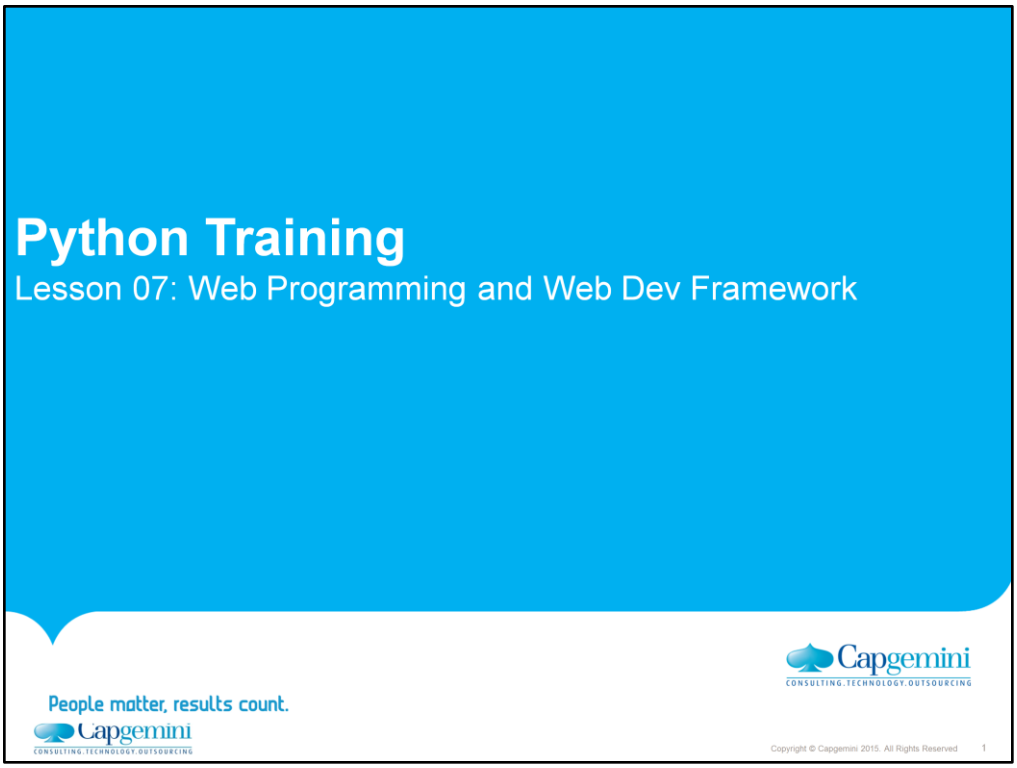

Python Training
Lesson 07: Web Programming and Web Dev Framework
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
People matter, results count.
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING
Copyright © Capgemini 2015. All Rights Reserved 1

Lesson Objectives

- What is Python Web Framework.
- Different frameworks in Python
- Django – Architecture
  - Urls
  - Views
  - Models
- Django Sample Implementation
- Django Benefits

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 2

Web Framework

- A Web framework is a collection of packages or modules which allow developers to write Web applications.
- It provides support for various activities such as getting requests, interpreting it, producing responses, storing data persistently, and so on.
- They are the de facto way to build web-enabled applications.
- The world of Python web frameworks is full of choices.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 3

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.
Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Frameworks provide functionality to perform different operations like,
URL routing
HTML, XML, JSON, and other output format templating
Database manipulation
Security against Cross-site request forgery (CSRF) and other attacks
Session storage and retrieval etc..

Popular Frameworks

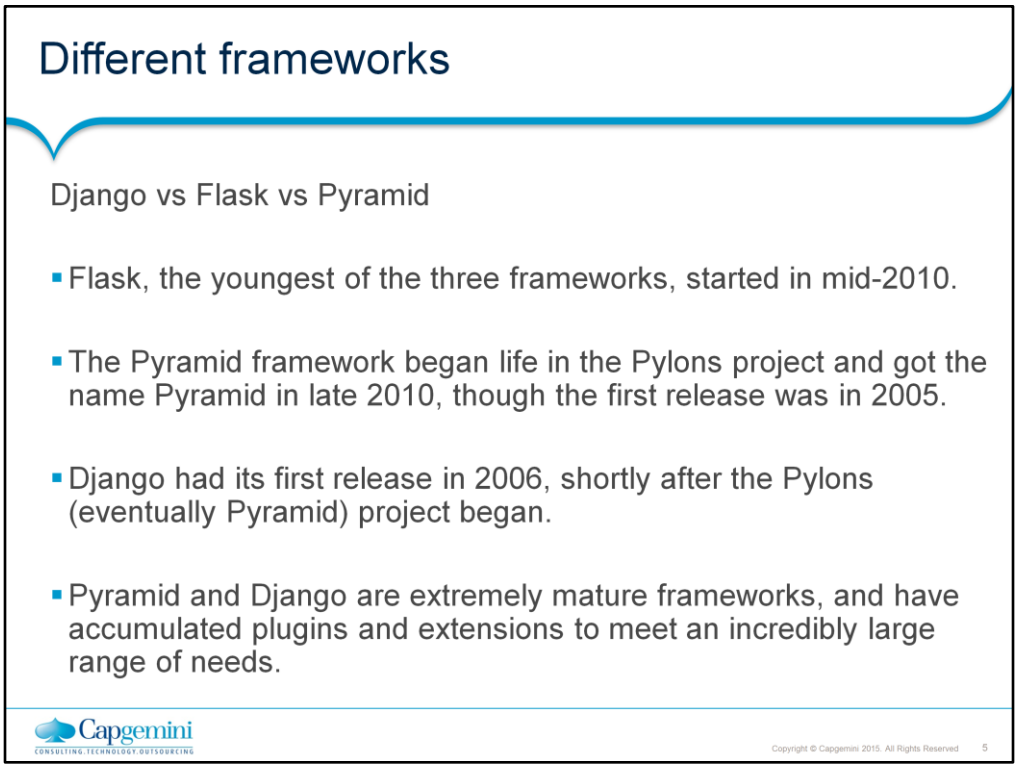
Following are few popularly used Python web frameworks.

- Django
- Pyramid
- Flask
- Bottle
- Other Frameworks
  - Web2py
  - Pylons
  - TurboGears and many..

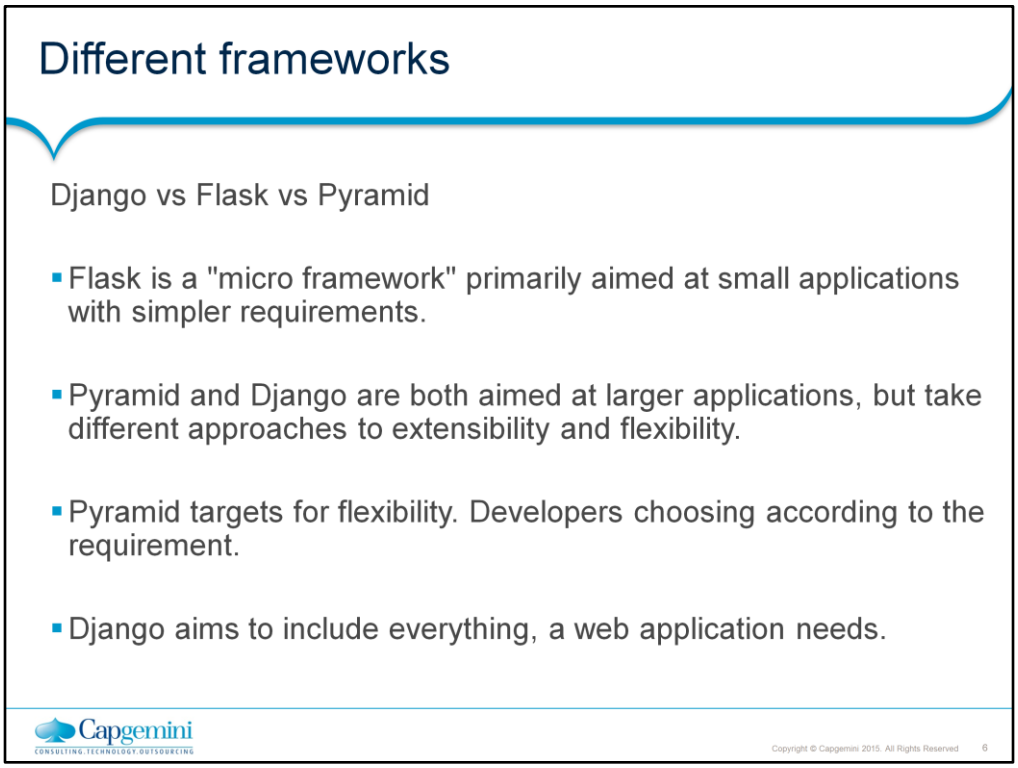
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 4

Add the notes here.

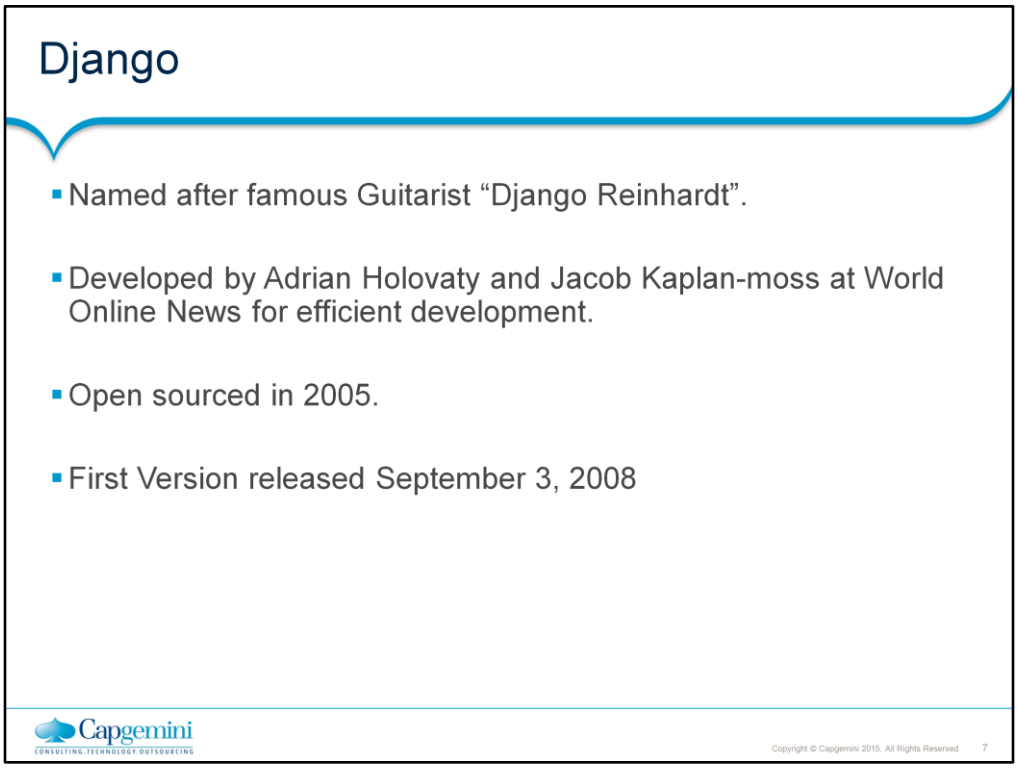


Django and Pyramid both come with bootstrapping tools built in. While Flask is not including it because Flask is not targeting large MVC application.



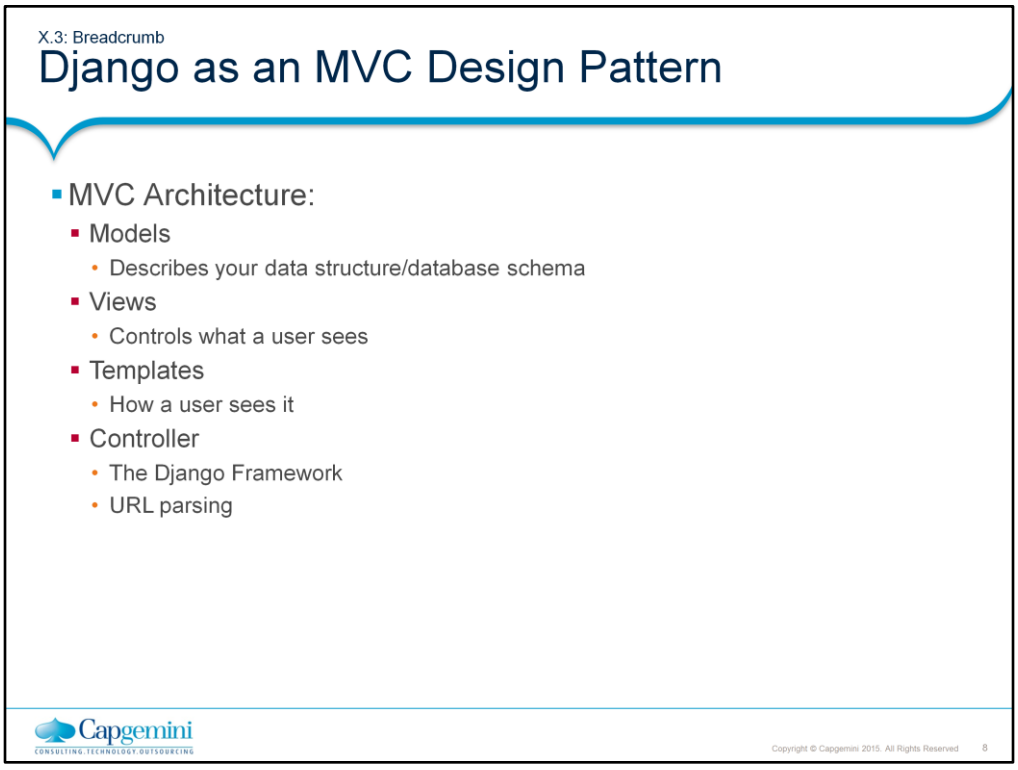
Flask is a "microframework" primarily aimed at small applications with simpler requirements. Pyramid and Django are both aimed at larger applications, but take different approaches to extensibility and flexibility. Pyramid targets flexibility and lets the developer use the right tools for their project. This means the developer can choose the database, URL structure, templating style, and more. Django aims to include all the batteries a web application will need so developers need only open the box and start working, pulling in Django's many modules as they go.

Django has for templating, forms, routing, authentication, basic database administration, and more built in. In contrast, Pyramid includes routing and authentication, but templating and database administration require external libraries.



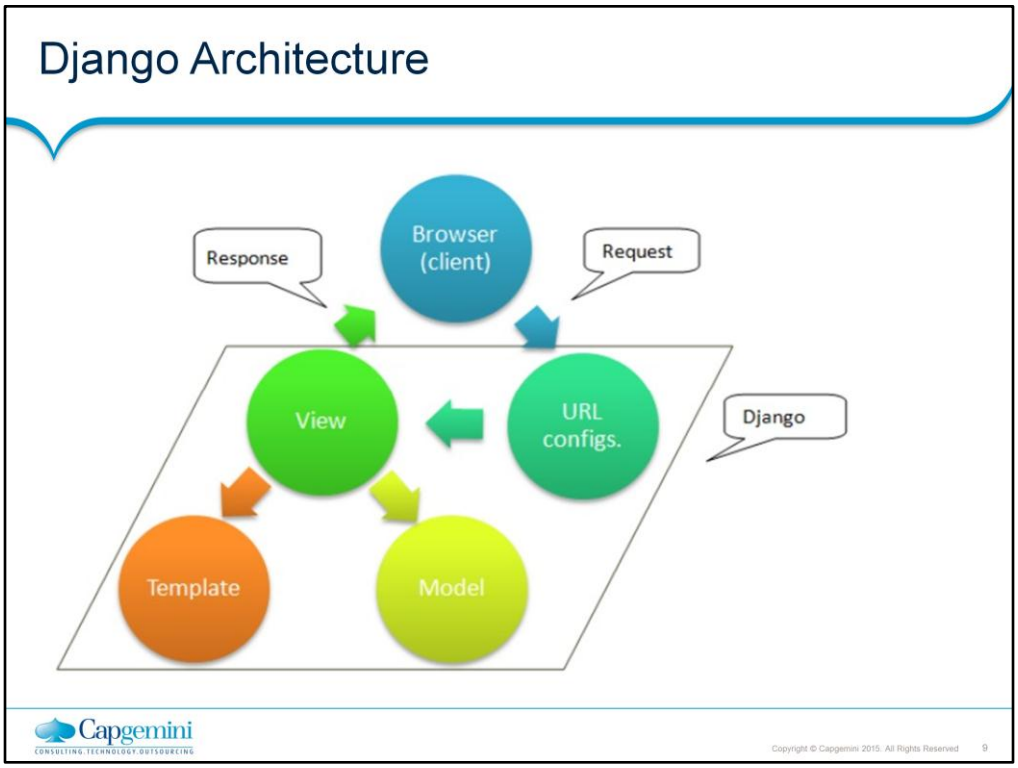
Django makes it easier to build better Web apps more quickly and with less code. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.

Django makes use of the *MVC* pattern and requires code using the framework to do the same. *MVC*, or "Model-View-Controller" is simply a way of logically separating the different responsibilities of the application. In Django, database tables are represented by models, controllers contains business logic of application and views contain information which they require to dynamically generate HTML representation of the page.



Models:
A model is the single, definitive source of information about data. It contains the essential fields and behaviors of the data which is being stored. Each model is a Python class that subclasses django.db.models.Model. Each attribute of the model represents a database field. Once models are defined, Djankgo must be informed about it. This is done by editting seetings.py files and changing INSTALLED_APPS as shown in demo.

Views:
A view is a place where we put the "logic" of application. It will request information from the model created before and pass it to a template. In simple terms view are just Python functions.



Templates:
A Django template is a string of text that is intended to separate the presentation of a document from its data. A template defines placeholders and various bits of basic logic (template tags) that regulate how the document should be displayed. Usually, templates are used for producing HTML, but Django templates are equally capable of generating any text-based format.

Controller:
A controller is the heart of the system, it steers everything. For a web framework, this means handling requests and responses, setting up database connections and loading add-ons. For this, Django reads a *settings file* so that it knows what to load and set up. And Django reads a *URL config file* that tells it what to do with the incoming requests from browsers.

What Django generates…

- MySite/
  - __init__.py
  - Manage.py // Script to interact with Django
  - Settings.py // Config
  - URLs.py // My Site URL mapping
  - MyProject /
    - __init__.py
    - URLs.py // Project specific URL mapping
    - Models.py // Data Models
    - Views.py // Contains the call back functions
    - Admin.py
    - Templates

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 10

Add the notes here.

What Django generates…

- The Django root directory will be named according to the project name you specified in
  - django-admin startproject [projectname].
- This directory is the project's connection with Django.
  - **[projectname]/settings/:** Instead of a plain *settings*-file, the configuration is split into several files in this Python module.
  - **[projectname]/urls.py:** The root URL configuration of the project. The only configured set of urls is the admin-application. Whenever user requests something from website then django will move to urls.py file.
  - **[projectname]/wsgi.py:** Deploying Django makes use of WSGI, the Pythonic way of deploying web applications.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 11

Add the notes here.

What Django generates…

- manage.py
  - manage.py is automatically created in each Django project. It does the following,
    - It puts your project's package on sys.path.
    - It sets the DJANGO_SETTINGS_MODULE environment variable so that it points to your project's settings.py file.
- mysite/__init__.py
  - An empty file that tells Python that this directory should be considered a Python package.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 12

Add the notes here.

Django commands

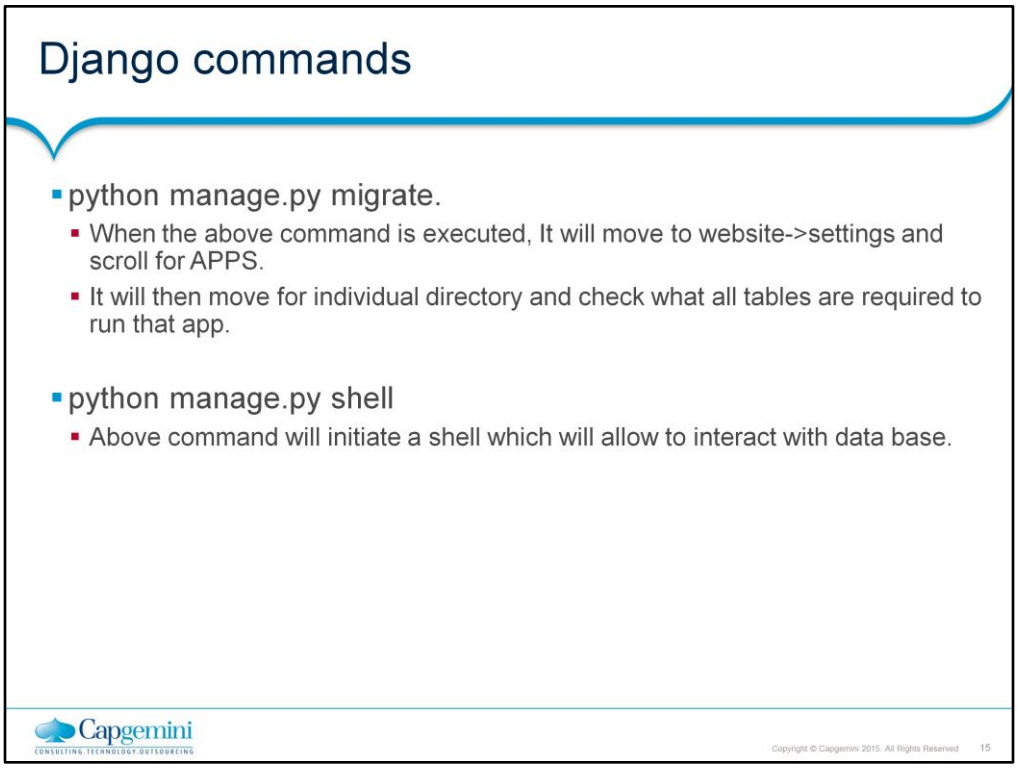
- django-admin startproject [projname]
  - django-admin is Django's command-line utility for administrative tasks.
  - The django-admin script should be on your system path if you installed Django via its setup.py utility
- python manage.py runserver
  - To start the server, change into your project container directory (cd projname) and run the above command
- python manage.py startapp [appname]
  - This command will help to create an app. Once the app is created, its directory structure Is as follows,

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

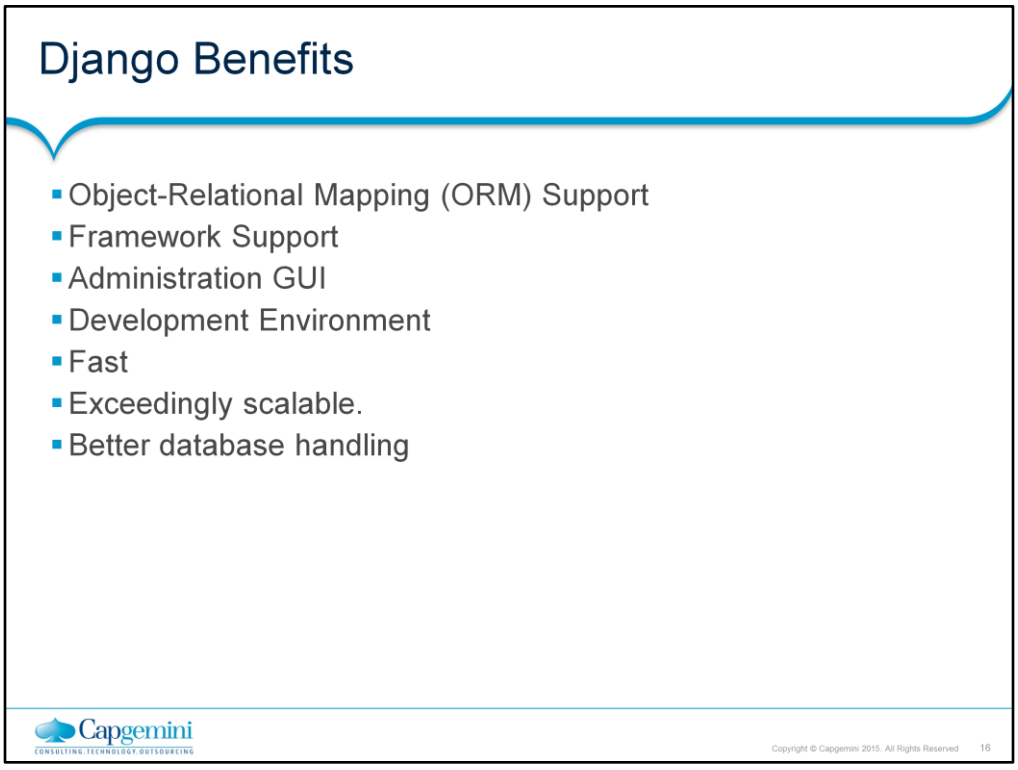
 13

Add the notes here.

Add the notes here.



Add the notes here.



As Django is the only framework that has the ability to generate admin panels on the fly depending on the database schema and table relations, it makes it the most powerful framework as of today when comes to sites where admin panel is widely used.

Django provides a complete development environment. It comes with a lightweight Web server for development and testing. When the debugging mode is enabled, Django provides very thorough and detailed error messages, with a lot of debugging information which, in turn, makes debugging easier, and allows one to quickly isolate bugs.

Django has a powerful Object-Relational Mapping system. It supports a large set of database systems, and switching from one engine to another is a matter of changing a configuration file.

Django has built-in support for Ajax, RSS, Caching and various other frameworks.

It has better database handling. whether you use MySQL, PostgreSQL, or Oracle Django uses the powerful python database handling to provide easy data manipulation, migration, etc.

Also Django has a very good module handling because of the powerful python language that lays beneath meaning you can connect python code with C/C++ libraries without installing any apache modules, etc.

Lab

- Demo 1
- Demo 2
- Demo 3
- Demo 4

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 17

Refer demos.

Summary

- Python Web framework
- Different Frameworks and differences
  - Django
  - Pyramid
  - Flask
- Django Architecture
- Django sample Implementation
- Django Benefits

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 18

Add the notes here.