**8.SOFTWARE ENVIRONMENT**

**8.1.Introduction**

Becoming a world class data analytics practitioner requires mastery of the most sophisticated data analytics tools. The R and python languages are some of the most powerful and flexible tools in the data analytics toolkit. This course teaches delegates with no prior programming or data analytics experience how to perform data manipulation, data analysis and data visualisation. Mastery of these techniques will allow delegates to immediately add value in their work place by extracting valuable insight from company data to allow better, data-driven decisions.

Key points

* Software product lines are related applications that are developed from a common base.This generic system is adapted to meet specific requirements for functionality,target platform or operational configuration.
* COTS product reuse is concerned with the reuse of large-scale,off-the-shelf system.These provide a lot of functionality and their reuse can radically reduce costs and development time.Systems may be developed by configuring a single,generic COTS product or by integrating two or more COTS products.
* Enterprise Resource Planning system are examples of lare-scale COTS reuse.you create an instance of an ERP system by configuring a generic system with information about the customer’s business processes and rules.
* Potential problems with COTS-based reuse include lack of control over functionality and performance,lack of control over system evolution,the need for support from external vendors and difficulties in ensuring that systems can inter-operate.

**STEP BY STEP PROCEDURE**

1).Install the Python:-

For Python programming you need a working Python installation and a text editor. Python comes with its own editor ''IDLE'', which is quite nice and totally sufficient for the beginning. As you get more into programming, you will probably switch to some other editor like ''emacs'', ''vi'' or another.

The Python download page is [https://www.python.org/downloads/ https://www.python.org/downloads/]. The most recent version is 3.5.3 (as of January 2017), but any ''Python 2.x'' version since 2.2 will work for this tutorial. Be careful with ''Python 3'', though, as some significant details have changed, and hence this tutorial's examples will not work with it. A version of this tutorial for Python 3 is at [[Non-Programmer's Tutorial for Python 3]]. There are various different installation files for different computer platforms available on the download site. Here are some specific instructions for the most common operating systems:

==== Linux, BSD and Unix users ====

You are probably lucky and Python is already installed on your machine. To test it type <tt>python</tt> on a command line. If you see something like that in the following section, you are set.

If you have to install Python, just use the operating system's package manager or go to the repository where your packages are available and get Python. Alternatively, you can compile Python from scratch after downloading the source code. If you get the source code make sure you compile in the Tk extension if you want to use IDLE.<!-- Is this realistic for people who don't know how to program??? -->

==== Mac users ====

Starting from Mac OS X (Tiger), Python ships by default with the operating system, but you might want to update to the newer version (check the version by starting <tt>python</tt> in a command line terminal). Also IDLE (the Python editor) might be missing in the standard installation. If you want to (re-)install Python, you can get it from the [https://www.python.org/downloads/mac-osx/ Mac download page] on the Python site.

==== Windows users ====

Some computer manufacturers pre-install Python. To check if you already have it installed, open command prompt (Windows Key+R then type cmd) and type python. If it says "Bad command or file name" you will need to download the appropriate Windows installer from [https://www.python.org/download/releases/2.7.8/ the Windows download page]. If you do not have a 64-bit AMD or Intel chip processor, you will need the '''Windows x86 MSI Installer'''; otherwise the '''Windows x86-64 MSI Installer''' is the correct one (they are very similarly named, so check carefully which one you are downloading). Start the installer by double-clicking it and follow the prompts.

After installing you will need to add the installation path to the PATH system variable if you wish to use it from the command prompt instead of the IDLE editor. {{BookCat}}

2).Install the database(Mysql):-

The default installation on any version of Windows is now much easier than it used to be, as MySQL now comes neatly packaged with an installer. Simply download the installer package, unzip it anywhere and run the setup.exe file.

The default installer setup.exe will walk you through the trivial process and by default will install everything under C:\mysql.

Test the server by firing it up from the command prompt the first time. Go to the location of the **mysqld server** which is probably C:\mysql\bin, and type −mysqld.exe --console

**Verifying MySQL Installation**

After MySQL, has been successfully installed, the base tables have been initialized and the server has been started: you can verify that everything is working as it should be via some simple tests.

### Use the mysqladmin Utility to Obtain Server Status

Use **mysqladmin** binary to check the server version. This binary would be available in /usr/bin on linux and in C:\mysql\bin on windows.

[root@host]# mysqladmin --version

It will produce the following result on Linux. It may vary depending on your installation −

mysqladmin Ver 8.23 Distrib 5.0.9-0, for redhat-linux-gnu on i386

If you do not get such a message, then there may be some problem in your installation and you would need some help to fix it.

### Execute simple SQL commands using the MySQL Client

You can connect to your MySQL server through the MySQL client and by using the **mysql** command. At this moment, you do not need to give any password as by default it will be set as blank.

You can just use following command −

[root@host]# mysql

It should be rewarded with a mysql> prompt. Now, you are connected to the MySQL server and you can execute all the SQL commands at the mysql> prompt as follows −

mysql> SHOW DATABASES;

+----------+

| Database |

+----------+

| mysql |

| test |

+----------+

2 rows in set (0.13 sec)

## Post-installation Steps

MySQL ships with a blank password for the root MySQL user. As soon as you have successfully installed the database and the client, you need to set a root password as given in the following code block −

[root@host]# mysqladmin -u root password "new\_password";

Now to make a connection to your MySQL server, you would have to use the following command −

[root@host]# mysql -u root -p

Enter password:\*\*\*\*\*\*\*

UNIX users will also want to put your MySQL directory in your PATH, so you won't have to keep typing out the full path everytime you want to use the command-line client.

For bash, it would be something like −

export PATH = $PATH:/usr/bin:/usr/sbin

## Running MySQL at Boot Time

If you want to run the MySQL server at boot time, then make sure you have the following entry in the /etc/rc.local file.

/etc/init.d/mysqld start

Also,you should have the mysqld binary in the /etc/init.d/ directory.

**Running and Shutting down MySQL Server**

First check if your MySQL server is running or not. You can use the following command to check it −

ps -ef | grep mysqld

If your MySql is running, then you will see **mysqld** process listed out in your result. If server is not running, then you can start it by using the following command −

root@host# cd /usr/bin

./safe\_mysqld &

Now, if you want to shut down an already running MySQL server, then you can do it by using the following command −

root@host# cd /usr/bin

./mysqladmin -u root -p shutdown

Enter password: \*\*\*\*\*\*

**Setting Up a MySQL User Account**

For adding a new user to MySQL, you just need to add a new entry to the **user** table in the database **mysql**.

The following program is an example of adding a new user **guest** with SELECT, INSERT and UPDATE privileges with the password **guest123;** the SQL query is −

root@host# mysql -u root -p

Enter password:\*\*\*\*\*\*\*

mysql> use mysql;

Database changed

mysql> INSERT INTO user

(host, user, password,

select\_priv, insert\_priv, update\_priv)

VALUES ('localhost', 'guest',

PASSWORD('guest123'), 'Y', 'Y', 'Y');

Query OK, 1 row affected (0.20 sec)

mysql> FLUSH PRIVILEGES;

Query OK, 1 row affected (0.01 sec)

mysql> SELECT host, user, password FROM user WHERE user = 'guest';

+-----------+---------+------------------+

| host | user | password |

+-----------+---------+------------------+

| localhost | guest | 6f8c114b58f2ce9e |

+-----------+---------+------------------+

1 row in set (0.00 sec)

When adding a new user, remember to encrypt the new password using PASSWORD() function provided by MySQL. As you can see in the above example, the password mypass is encrypted to 6f8c114b58f2ce9e.

Notice the FLUSH PRIVILEGES statement. This tells the server to reload the grant tables. If you don't use it, then you won't be able to connect to MySQL using the new user account at least until the server is rebooted.

You can also specify other privileges to a new user by setting the values of following columns in user table to 'Y' when executing the INSERT query or you can update them later using UPDATE query.

* Select\_priv
* Insert\_priv
* Update\_priv
* Delete\_priv
* Create\_priv
* Drop\_priv
* Reload\_priv
* Shutdown\_priv
* Process\_priv
* File\_priv
* Grant\_priv
* References\_priv
* Index\_priv
* Alter\_priv

**Administrative MySQL Command**

Here is the list of the important MySQL commands, which you will use time to time to work with MySQL database −

* **USE Databasename** − This will be used to select a database in the MySQL workarea.
* **SHOW DATABASES** − Lists out the databases that are accessible by the MySQL DBMS.
* **SHOW TABLES** − Shows the tables in the database once a database has been selected with the use command.
* **SHOW COLUMNS FROM *tablename:*** Shows the attributes, types of attributes, key information, whether NULL is permitted, defaults, and other information for a table.
* **SHOW INDEX FROM tablename** − Presents the details of all indexes on the table, including the PRIMARY KEY.
* **SHOW TABLE STATUS LIKE tablename\G** − Reports details of the MySQL DBMS performance and statistics.

**3.Install the sublime text:-**

Sublime Text is a cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and its functionality can be extended by users with plugins, typically community-built and maintained under free-software licenses.

### Features

* Editing files side by side.
* It supported all Platforms.
* It provides functionality to find and replace with regular expressions.
* The Command Palette gives fast access to functionality.
* “Goto Anything,” quick navigation to files, symbols, or lines.
* Python-based plugin API.
* Compatible with many language grammars from TextMate.

**4.Install the Django:-**

Being a Python Web framework, Django requires Python. See [What Python version can I use with Django?](https://docs.djangoproject.com/en/2.0/faq/install/#faq-python-version-support) for details. Python includes a lightweight database called [SQLite](https://sqlite.org/) so you won’t need to set up a database just yet.

Get the latest version of Python at <https://www.python.org/downloads/> or with your operating system’s package manager.

You can verify that Python is installed by typing **python** from your shell; you should see something like:

Python 3.4.x

[GCC 4.x] on linux

Type "help", "copyright", "credits" **or** "license" **for** more information.

>>>

### Installing an official release with pip[¶](https://docs.djangoproject.com/en/2.0/topics/install/#installing-an-official-release-with-pip)

This is the recommended way to install Django.

1. Install [pip](https://pip.pypa.io/). The easiest is to use the [standalone pip installer](https://pip.pypa.io/en/latest/installing/#installing-with-get-pip-py). If your distribution already has **pip** installed, you might need to update it if it’s outdated. If it’s outdated, you’ll know because installation won’t work.
2. Take a look at [virtualenv](https://virtualenv.pypa.io/) and [virtualenvwrapper](https://virtualenvwrapper.readthedocs.io/en/latest/). These tools provide isolated Python environments, which are more practical than installing packages systemwide. They also allow installing packages without administrator privileges. The [contributing tutorial](https://docs.djangoproject.com/en/2.0/intro/contributing/) walks through how to create a virtualenv.
3. After you’ve created and activated a virtual environment, enter the command **pip install Django** at the shell prompt.
4. Check out Django’s main development branch like so:
5. **$** git clone https://github.com/django/django.git
6. After setting up and activating the virtualenv, run the following command:
7. **$** pip install -e django/

**8.2.PYTHON LIBRARIES**

1. [**Agate**](https://agate.readthedocs.io/en/1.4.0/)**:** data analysis library with human-readable code
2. .[**Basemap**](https://pypi.python.org/pypi/basemap/1.0.7/)**:** matplotlib toolkit to plot data on maps with coastlines, lakes, rivers, and political boundaries
3. [**Beautiful Soup**](https://www.crummy.com/software/BeautifulSoup/): library for parsing HTML, JSON and XML data
4. [**Cufflinks**](https://pypi.python.org/pypi/cufflinks)**:** library to bind Plotly directly to pandas dataframes
5. [**Engarde**](http://engarde.readthedocs.io/en/latest/index.html): package for defensive data analysis
6. [**Emcee**](http://dan.iel.fm/emcee/current/): An MIT [MCMC](http://msp.org/camcos/2010/5-1/index.xhtml) library
7. [**Folium**](https://folium.readthedocs.io/en/latest/)**:** library to build Leaflet.js maps in Python
8. [**Gensim**](https://radimrehurek.com/gensim/)**:** library for unsupervised semantic modeling from plain text
9. [**Geomap**](https://pypi.python.org/pypi/geomap/0.2.1): package to generate maps of geolocations
10. [**HDBSCAN**](https://pypi.python.org/pypi/hdbscan/)**:** Hierarchical Density-Based Spatial Clustering of Applications with Noise
11. [**Jsonify**](https://pypi.python.org/pypi/jsonify/0.5): takes a .csv file and converts to .json format
12. [**Keras**](https://keras.io/): high-level neural networks API that can run on TensorFlow or Theano
13. [**Lifelines**](http://lifelines.readthedocs.io/en/latest/)**:** an implementation of survival analysis in Python
14. [**Lifetimes**](https://pypi.python.org/pypi/Lifetimes): package for analyzing user behavior
15. [**Matplotlib**](http://matplotlib.org/): 2D plotting library
16. [**Networkx**](https://networkx.github.io/)**:** package to create, manipulate, and study complex networks
17. [**NLTK**](http://www.nltk.org/)**:** natural language toolkit for working with human language data
18. [**Numexpr**](https://pypi.python.org/pypi/numexpr): fast numerical array expression evaluator
19. [**NumPy**](http://www.numpy.org/)**:** scientific computing library
20. [**Pandas**](http://pandas.pydata.org/)**:** library of data structures and data analysis tools
21. [**PandaSQL**](https://pypi.python.org/pypi/pandasql)**:** queries pandas dataframes using SQL syntax
22. [**Patsy**](https://pypi.python.org/pypi/patsy)**:** library for describing statistical models/building design matrices
23. [**Plotly**](https://plot.ly/python/getting-started/): library for data visualization, dashboards & collaborative analysis
24. [**PrettyTable**](https://pypi.python.org/pypi/PrettyTable)**:** library for easily displaying tabular data in ASCII table format
25. [**Prophet**](https://facebook.github.io/prophet/)**:** a procedure for forecasting time series data
26. [**PyZipCode**](https://pypi.python.org/pypi/pyzipcode)**:** query zip codes and location data
27. [**Requests\***](http://docs.python-requests.org/en/master/): library that allows HTTP requests
28. [**Scikit**-Image](http://scikit-image.org/): image processing library
29. [**Scikit**-Learn](http://scikit-learn.org/stable/): library of tools for data mining and analysis
30. [**Scipy**](https://www.scipy.org/): library for mathematics, science and engineering
31. [**Seaborn**](https://stanford.edu/~mwaskom/software/seaborn/): visualization library that allows developers to draw statistical graphics
32. [**SexMachine**](https://pypi.python.org/pypi/SexMachine/): predict gender from first names
33. [**Statsmodels**](http://statsmodels.sourceforge.net/): library for estimating statistical models/performing statistical tests
34. [**SymPy**](http://www.sympy.org/en/index.html)**:** library for symbolic mathematics
35. [**Tabulate**](https://pypi.python.org/pypi/tabulate)**:** pretty-print tabular data
36. [**TensorFlow**](https://www.tensorflow.org/)**:** library for numerical computation using data flow graphs
37. [**TextBlob**](http://textblob.readthedocs.io/en/dev/)**:** simple API for common NLP tasks
38. [**UA-Parser**](https://pypi.python.org/pypi/ua-parser)**:** fast and reliable user agent parser for Python
39. [**Wordcloud**](https://github.com/amueller/word_cloud)**:** a little wordcloud generator in Python

**PYTHON WEBKIT:-**

The Python Webkit DOM Project makes python a full peer of javascript when it comes to accessing and manipulating the full features available to Webkit, such as HTML5. Everything that can be done with javascript, such as getElementsbyTagName and appendChild, event callbacks through onclick, timeout callbacks through window.setTimeout, and even AJAX using XMLHttpRequest, can also be done from python.

Let’s start our project: if you’re on the [Welcome screen](https://www.jetbrains.com/help/pycharm/welcome-screen.html), click **Create New Project**. If you’ve already got a project open, choose **File | New Project**.

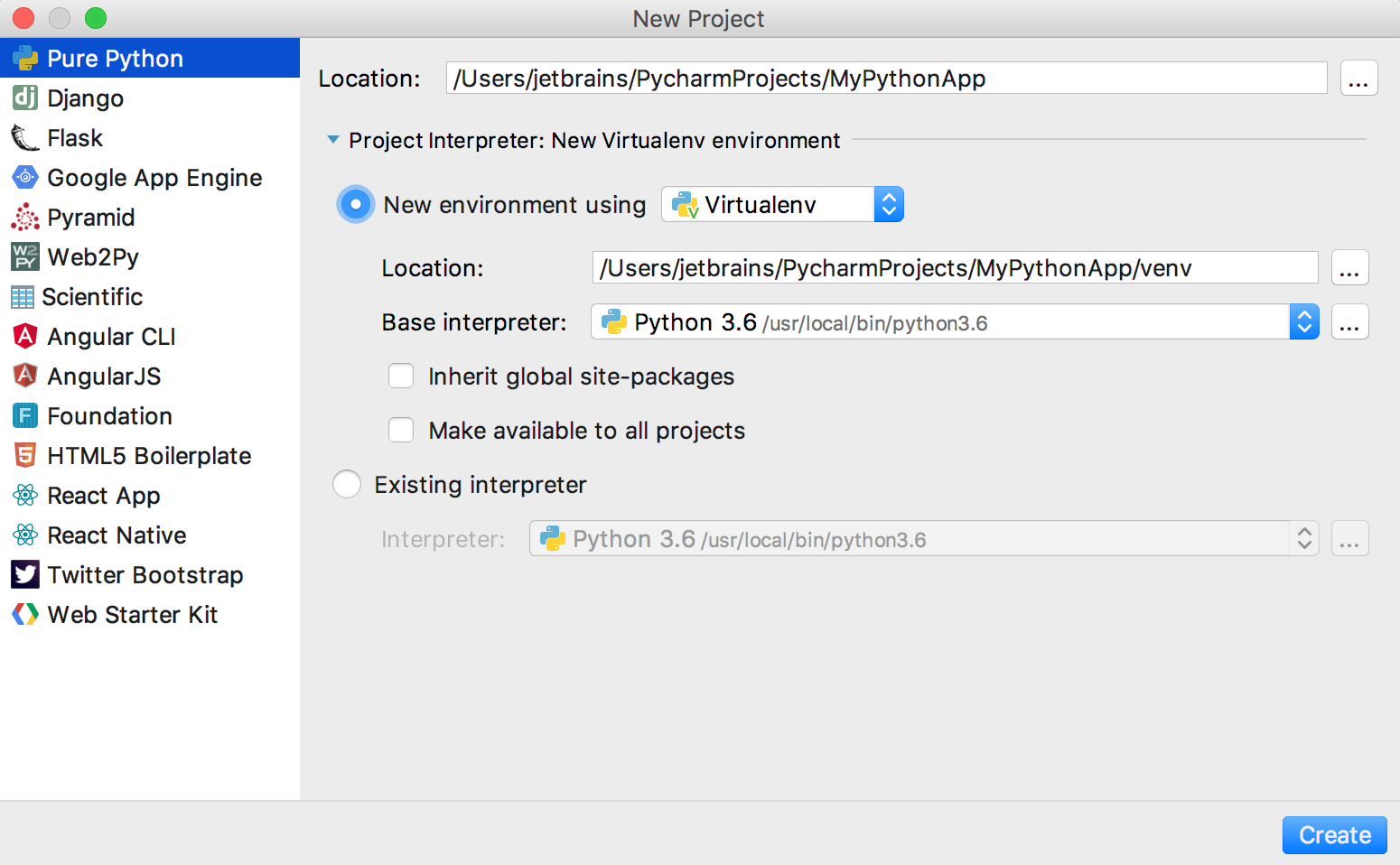
PyCharm suggests several project templates for the creation of the various types of applications (Django, Google AppEngine,, etc.). When PyCharm creates a new project from a project template, it produces the corresponding directory structure and specific files, and any needed run configurations or settings.

In this tutorial we’ll create a simple Python script, so we’ll choose **Pure Python**. This template will create an empty project for us.

Choose the project location. To do that, click browseButton button next to the **Location** field, and specify the directory for your project.

Python best practice is to create a virtualenv for each project. To do that, expand the **Project Interpreter: New Virtualenv Environment** node and select a tool used to create a new virtual environment. Let's choose Virtualenv tool, and specify the location and base interpreter used for the new virtual environment. Select the two check boxes below if necessary.

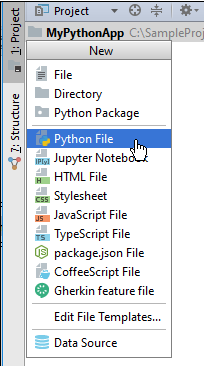
Then click the Create button at the bottom of the New Project dialog.



If you’ve already got a project open, after clicking **Create** PyCharm will ask you whether to open a new project in the current window or in a new one. Choose **Open in current window** - this will close the current project.

**Creating a Python file**

Select the project root in the Project tool window, and press Alt+Insert.



Choose the option **Python file** from the pop-up window, and then type the new file name **Car**.

PyCharm creates a new Python file and opens it for editing.

