**LITERATURE SURVEY**

1.[**https://www.tutorialspoint.com/django/**](https://www.tutorialspoint.com/django/)

**Django Tutorial:**

Django is a web development framework that assists in building and maintaining quality web applications. Django helps eliminate repetitive tasks making the development process an easy and time saving experience. This tutorial gives a complete understanding of Django.

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

**Note** − Django is a registered trademark of the Django Software Foundation, and is licensed under BSD License.

## History of Django

* **2003** − Started by Adrian Holovaty and Simon Willison as an internal project at the Lawrence Journal-World newspaper.
* **2005** − Released July 2005 and named it Django, after the jazz guitarist Django Reinhardt.
* **2005** − Mature enough to handle several high-traffic sites.
* **Current** − Django is now an open source project with contributors across the world.
* The Model-View-Template (MVT) is slightly different from MVC. In fact the main difference between the two patterns is that Django itself takes care of the Controller part (Software Code that controls the interactions between the Model and View), leaving us with the template. The template is a HTML file mixed with Django Template Language (DTL).
* The following diagram illustrates how each of the components of the MVT pattern interacts with each other to serve a user request −
* 
* The developer provides the Model, the view and the template then just maps it to a URL and Django does the magic to serve it to the user.

## Step 1 – Installing Python

Django is written in 100% pure Python code, so you'll need to install Python on your system. Latest Django version requires Python 2.6.5 or higher

If you're on one of the latest Linux or Mac OS X distribution, you probably already have Python installed. You can verify it by typing *python* command at a command prompt. If you see something like this, then Python is installed.

$ python

Python 2.7.5 (default, Jun 17 2014, 18:11:42)

[GCC 4.8.2 20140120 (Red Hat 4.8.2-16)] on linux2

Otherwise, you can download and install the latest version of Python from the link [http://www.python.org/download](https://www.python.org/downloads/).

## Step 2 - Installing Django

Installing Django is very easy, but the steps required for its installation depends on your operating system. Since Python is a platform-independent language, Django has one package that works everywhere regardless of your operating system.

You can download the latest version of Django from the link [http://www.djangoproject.com/download](https://www.djangoproject.com/download/).

### UNIX/Linux and Mac OS X Installation

You have two ways of installing Django if you are running Linux or Mac OS system −

* You can use the package manager of your OS, or use easy\_install or pip if installed.
* Install it manually using the official archive you downloaded before.

We will cover the second option as the first one depends on your OS distribution. If you have decided to follow the first option, just be careful about the version of Django you are installing.

Let's say you got your archive from the link above, it should be something like Django-x.xx.tar.gz:

Extract and install.

$ tar xzvf Django-x.xx.tar.gz

$ cd Django-x.xx

$ sudo python setup.py install

You can test your installation by running this command −

$ django-admin.py --version

If you see the current version of Django printed on the screen, then everything is set.

**Note** − For some version of Django it will be django-admin the ".py" is removed.

### Windows Installation

We assume you have your Django archive and python installed on your computer.

First, PATH verification.

On some version of windows (windows 7) you might need to make sure the Path system variable has the path the following C:\Python34\;C:\Python34\Lib\site-packages\django\bin\ in it, of course depending on your Python version.

Then, extract and install Django.

c:\>cd c:\Django-x.xx

Next, install Django by running the following command for which you will need administrative privileges in windows shell "cmd" −

c:\Django-x.xx>python setup.py install

To test your installation, open a command prompt and type the following command −

c:\>python -c "import django; print(django.get\_version())"

If you see the current version of Django printed on screen, then everything is set.

OR

Launch a "cmd" prompt and type python then −

c:\> python

>>> import django

>>> django.VERSION

## Step 3 – Database Setup

Django supports several major database engines and you can set up any of them based on your comfort.

* [MySQL (http://www.mysql.com/)](http://www.mysql.com/)
* [PostgreSQL (http://www.postgresql.org/)](http://www.postgresql.org/)
* [SQLite 3 (http://www.sqlite.org/)](http://www.sqlite.org/)
* [Oracle (http://www.oracle.com/)](http://www.oracle.com/index.html)
* [MongoDb (https://django-mongodb-engine.readthedocs.org)](https://django-mongodb-engine.readthedocs.org/en/latest/)
* [GoogleAppEngine Datastore (https://cloud.google.com/appengine/articles/django-nonrel)](https://cloud.google.com/appengine/articles/django-nonrel)

You can refer to respective documentation to installing and configuring a database of your choice.

**Note** − Number 5 and 6 are NoSQL databases.

## Step 4 – Web Server

Django comes with a lightweight web server for developing and testing applications. This server is pre-configured to work with Django, and more importantly, it restarts whenever you modify the code.

However, Django does support Apache and other popular web servers such as Lighttpd. We will discuss both the approaches in coming chapters while working with different examples.

2.[**https://www.tutorialspoint.com/python/index.htm**](https://www.tutorialspoint.com/python/index.htm)

**Python** is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This **tutorial** gives enough understanding on **Python programming** language.

## Why to Learn Python?

**Python** is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

**Python** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python:

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## Characteristics of Python

Following are important characteristics of **Python Programming** −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

## Hello World using Python.

Just to give you a little excitement about Python, I'm going to give you a small conventional Python Hello World program, You can try it using Demo link.

[Live Demo](http://tpcg.io/4nOH9K)

print ("Hello, Python!");

## Applications of Python

As mentioned before, Python is one of the most widely used language over the web. I'm going to list few of them here:

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## Python Features

Python's features include −

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

## First Python Program

Let us execute programs in different modes of programming.

### Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

$ python

Python 2.4.3 (#1, Nov 11 2010, 13:34:43)

[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2

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

>>>

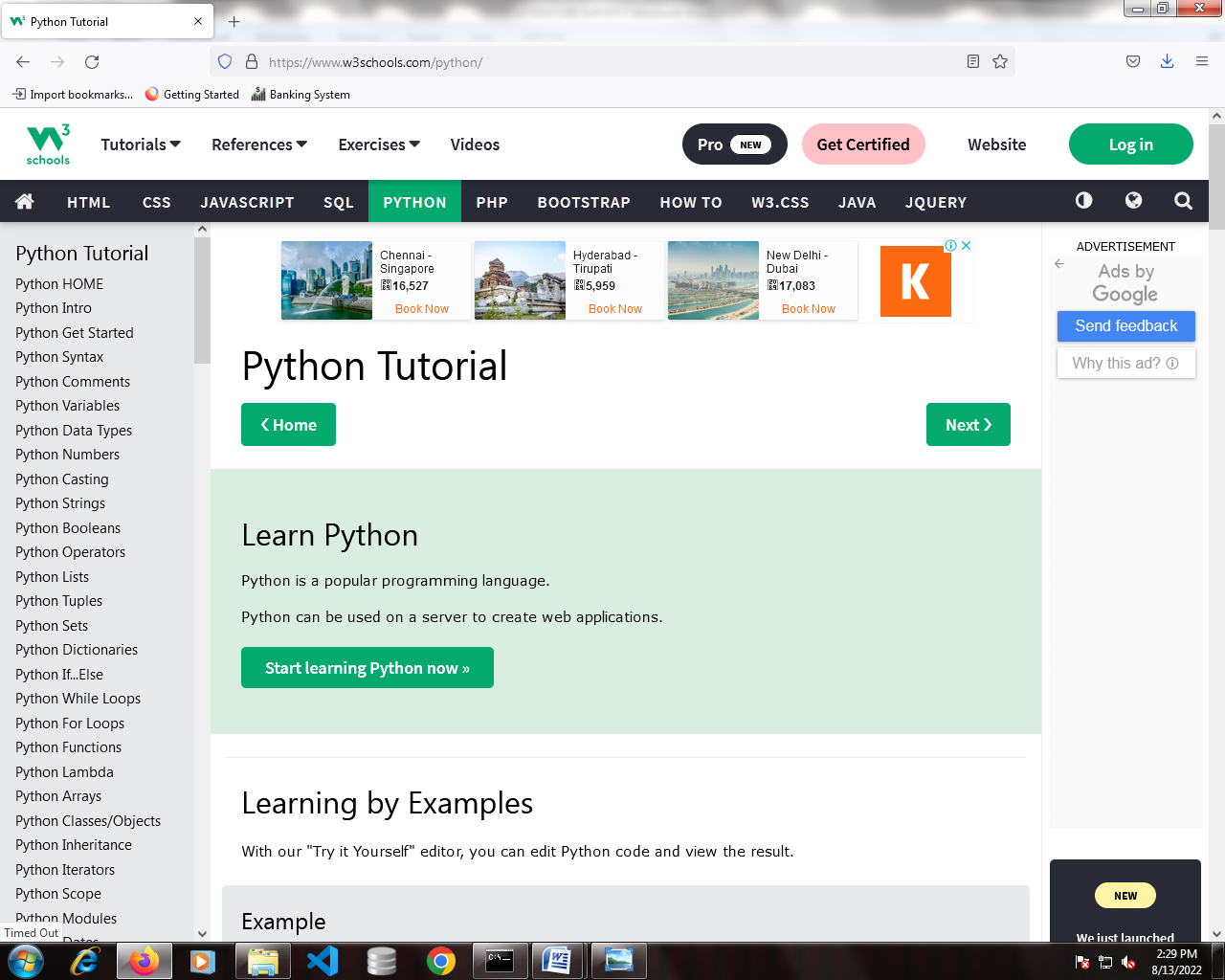
Type the following text at the Python prompt and press the Enter −

>>> print "Hello, Python!"

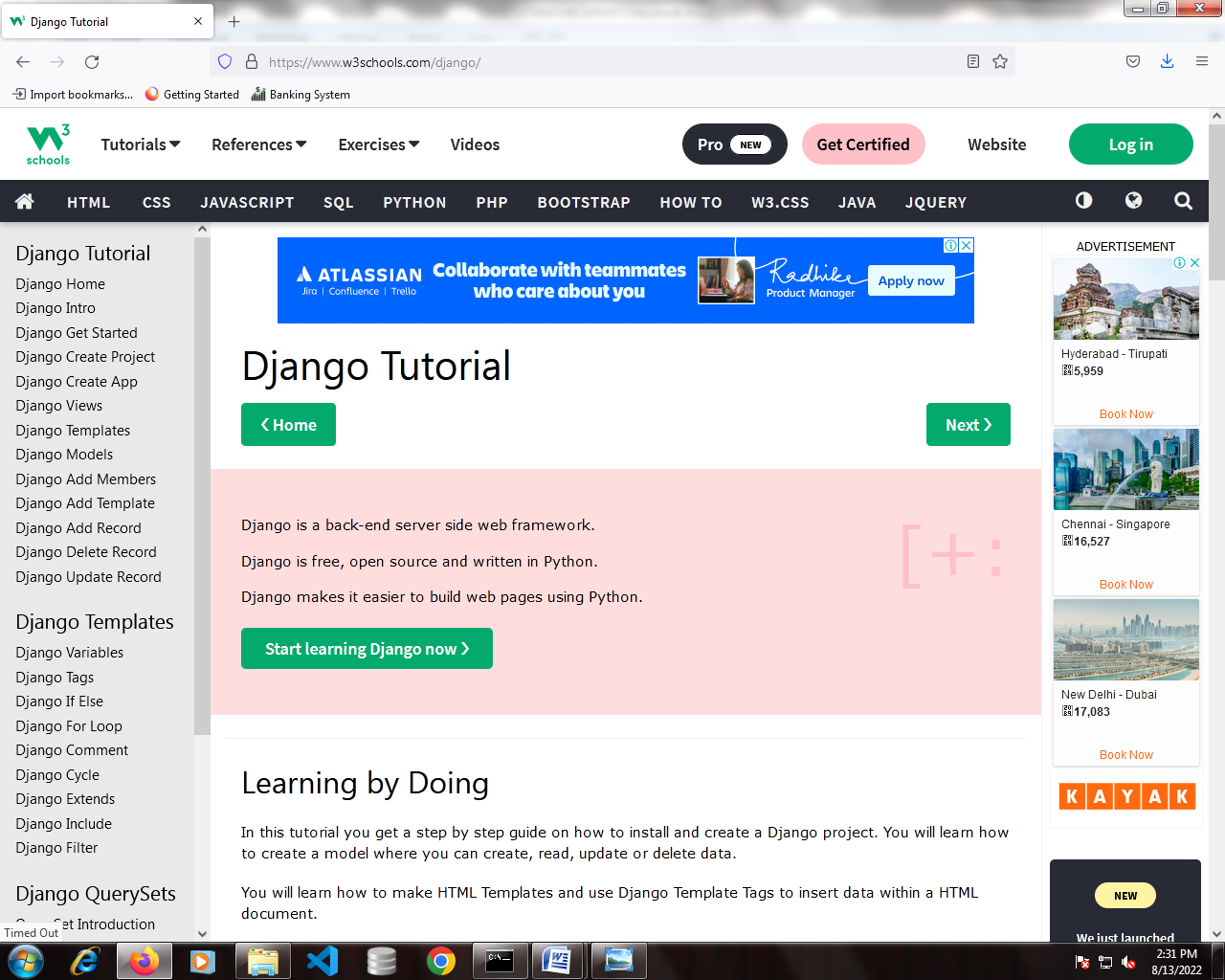
If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result −

Hello, Python!

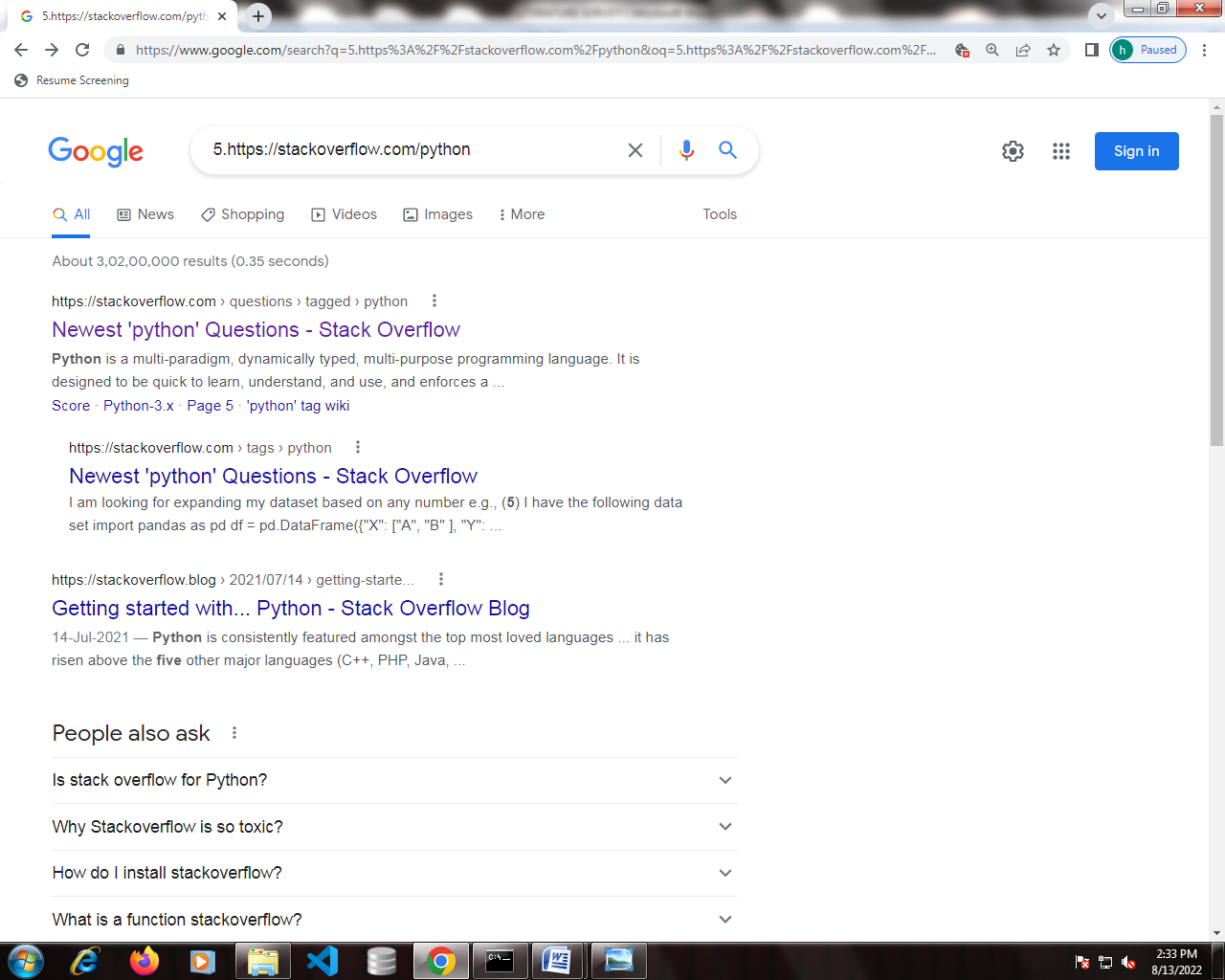
**3.** [**https://www.w3schools.com/python/**](https://www.w3schools.com/python/)

****

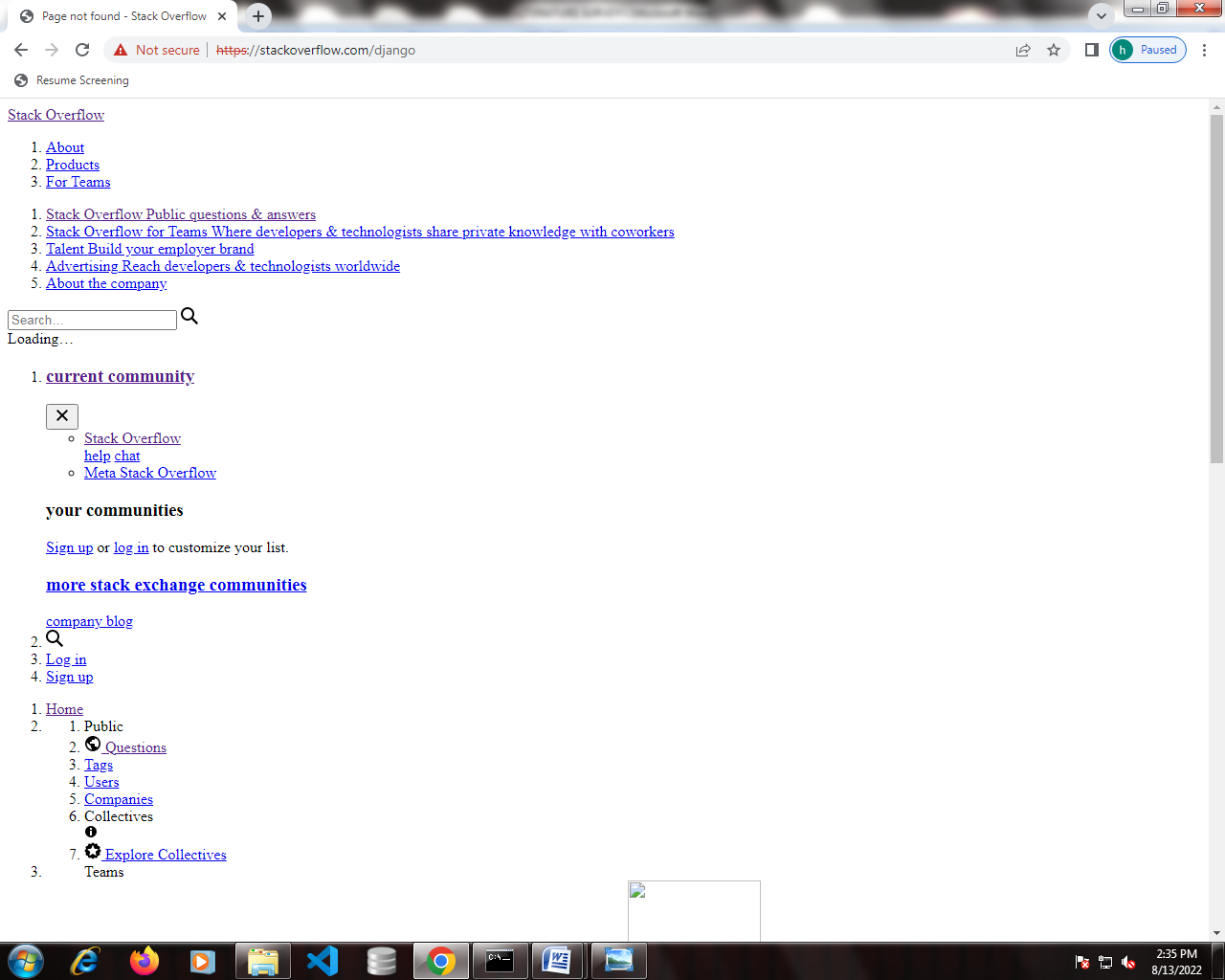
4.[**https://www.w3schools.com/django/**](https://www.w3schools.com/django/)

****

5.[**https://stackoverflow.com/python**](https://stackoverflow.com/python)



**6.** [**https://stackoverflow.com/django**](https://stackoverflow.com/django)

****