EasyChair Preprint

№ 4827

Coding Website (like Hacker Rank) Using Python and Django

Gade Uttej Kumar, Boina Varun and Shembelle Sangameshwar

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 29, 2020

CODING WEBSITE (LIKE HACKER RANK) USING PYTHON AND DJANGO

GADE UTTEJ KUMAR, BOINA VARUN, SHEMBELLE SANGAMESHWAR

MALLA REDDY COLLEGE OF ENGINEERING

Abstract:

At a time when organizations plan to pay 20% above the market for quality engineering talent, benchmarking candidates' coding capabilities and testing their domain knowledge is critical to identifying job-fit developers. A performance-based coding test platform, provides real-time feedback to help companies assess candidates' coding skills for performance benchmarking and effective recruitment.

The platform is integrated with domain-based programming tests to assess candidates' software development capability and benchmark it with industry standards. With this project you can make use of the comprehensive programming library to administer Python test, Java test, C programming test, SQL test, PHP developer test and other coding tests for hiring programmers. With our coding test platform, you can check candidate's programming skills in a few easy steps. The test results are auto evaluated, and the detailed test output. This not only helps organizations make efficient hiring decisions in less time but also makes it easy for recruiters from IT and non-IT backgrounds to administer coding tests. Programming skills tests, organizations can design an efficient developer recruitment cycle, bridge skill gaps for employee career progression, build customized coding scenarios based on business needs, and tailor pre- as well as post-training requirements. And mainly the application developed with Python and Django. Here majorly we have code console, input console and output console, in code console we can implement the entire our code and we pass the input values from input console, then finally it will display the output of the functionality.

Keywords: Python 3, Django.

INTRODUCTION:

In this article I will explain how you can make competitive coding website like geeks for geeks and hacker rank using Django framework. I will not explain whole coding because it is too lengthy process so i will explain some important part of this kind of websites. So what are the parts of these kind of website. Well, basically it contains 3 parts. First part is article or blog on some topic, second part is related question to articles an third part is online IDE so you can perform coding online.

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 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.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Local Environment Setup:

Open a terminal window and type "python" to find out if it is already installed and which version is installed.

- Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX, etc.)
- Win 9x/NT/2000
- Macintosh (Intel, PPC, 68K)
- OS/2
- DOS (multiple versions)
- Palm OS
- Nokia mobile phones
- Windows CE
- Acorn/RISC OS
- BeOS
- Amiga
- VMS/OpenVMS
- QNX
- VxWorks
- Psion
- Python has also been ported to the Java and .NET virtual machines

Getting Python:

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org/

You can download Python documentation from https://www.python.org/doc/. The documentation is available in HTML, PDF, and PostScript formats.

Installing Python:

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

If the binary code for your platform is not available, you need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

Here is a quick overview of installing Python on various platforms

Windows Installation:

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to https://www.python.org/downloads/.
- Follow the link for the Windows installer *python-XYZ.msi* file where XYZ is the version you need to install.
- To use this installer *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

Setting up PATH:

Programs and other executable files can be in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables.

The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.

The path variable is named as PATH in Unix or Path in Windows (Unix is case sensitive; Windows is not).

In Mac OS, the installer handles the path details. To invoke the Python interpreter from any particular directory, you must add the Python directory to your path.

Setting path at Windows:

To add the Python directory to the path for a particular session in Windows −

At the command prompt − type path %path%;C:\Python and press Enter.

Note − C:\Python is the path of the Python directory

IDLE [Integrated Development Learning Environment]:

Every Python installation comes with an **Integrated Development and Learning Environment**, which you'll see shortened to IDLE or even IDE. These are a class of applications that help you write code more efficiently. While there are many IDEs for you to choose from, Python IDLE is very bare-bones, which makes it the perfect tool for a beginning programmer.

Python IDLE comes included in Python installations on Windows and Mac. If you're a Linux user, then you should be able to find and download Python IDLE using your package manager. Once you've installed it, you can then use Python IDLE as an interactive interpreter or as a file editor.

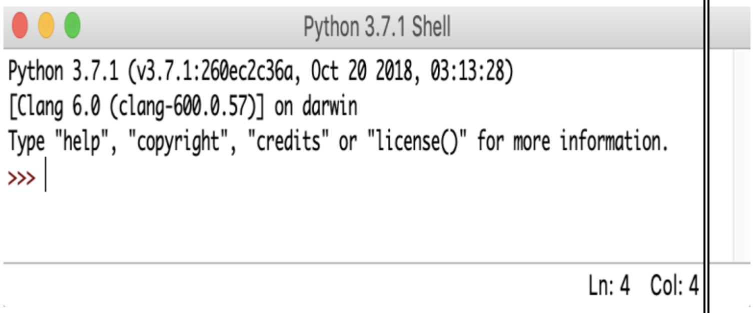
An Interactive Interpreter:

The best place to experiment with Python code is in the interactive interpreter, otherwise known as a **shell**. The shell is a basic Read-sEval-Print Loop (REPL). It reads a Python statement, evaluates the result of that statement, and then prints the result on the screen. Then, it loops back to read the next statement.

The Python shell is an excellent place to experiment with small code snippets. You can access it through the terminal or command line app on your machine. You can simplify your workflow with Python IDLE, which will immediately start a Python shell when you open it.

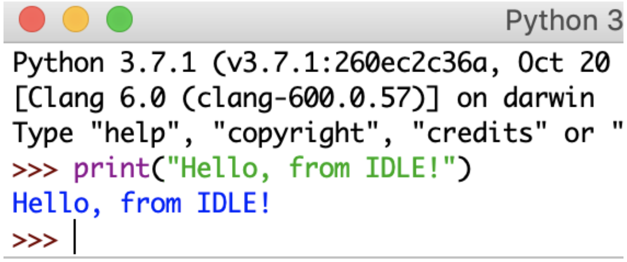
How to Use the Python IDLE Shell:

The shell is the default mode of operation for Python IDLE. When you click on the icon to open the program, the shell is the first thing that you see:



```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 03:13:28)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
```

This is a blank Python interpreter window. You can use it to start interacting with Python immediately. You can test it out with a short line of code:



```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "
>>> print("Hello, from IDLE!")
Hello, from IDLE!
>>> 
```

Here, you used print() to output the string "Hello, from IDLE!" to your screen. This is the most basic way to interact with Python IDLE. You type in commands one at a time and Python responds with the result of each command.

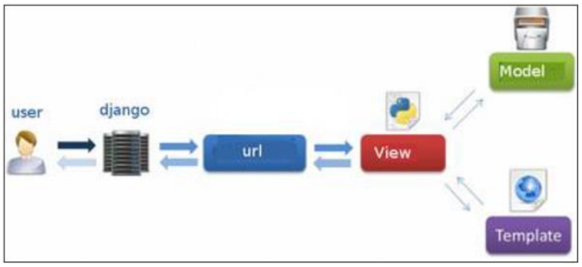
DJANGO:

As you already know, Django is a Python web framework. And like most modern framework, Django supports the MVC pattern. First let's see what is the Model-View-Controller (MVC) pattern, and then we will look at Django’s specificity for the Model-View-Template (MVT) pattern.

MVC Pattern: When talking about applications that provides UI (web or desktop), we usually talk about MVC architecture. And as the name suggests, MVC pattern is based on three components: Model, View, and Controller.

DJANGO MVC - MVT Pattern:

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.

Django development environment consists of installing and setting up Python, Django, and a Database System. Since Django deals with web application, it's worth mentioning that you would need a web server setup as well.

- **URLs:** While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.
- **View:** A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via *models*, and delegate the formatting of the response to *templates*.
- **Models:** Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
- **Templates:** A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A *view* can dynamically create an HTML page using an HTML template, populating it with data

from a *model*. A template can be used to define the structure of any type of file; it doesn't have to be HTML!

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

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- System: intel i5 10th Gen
- Hard Disk: 1 TB HDD +256 GB SSD
- Monitor: 15’’ LED
- Input Devices: Keyboard, Mouse
- Ram : 8GB.

SOFTWARE REQUIREMENTS:

- Operating System: Windows 10.
- Coding Language: Python
- Tool: Django, PyCharm

REFERENCES:

- GOOGLE
- HACKER RANK
- YOUTUBE
- GEEKS FOR GEEKS