

Lab 2-6-2023




1. Download and install python. You can use the following link:

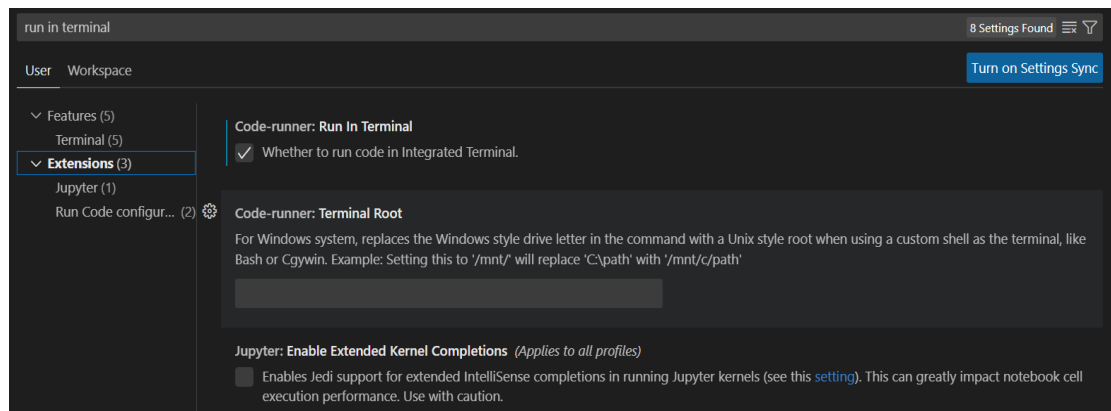
<https://www.python.org/downloads/>

2. Install VS Code. Please follow the instructions below for installation.

- a. Download VS Code for your desired operating system using following link.

<https://code.visualstudio.com/download>

- b. Follow the installation process as per instructions.
- c. Press Ctrl+Shift+X or go to the extensions tab (icon looking like this ). Install the “code runner” extension.
- d. Create a new folder and a new file. You can do that by pressing these Icons respectively ,  . Name the file in file_name.py format.
- e. Lastly, go to setting in bottom left corner or press (Ctrl + ,). Search for “run in terminal” in the search bar above.



Make sure that Run in Terminal checkbox has been checked.

3. Write your first code as `print ("hello world")`

```
1 print("hello world")
```

if it runs then you are good to go.

4. Now, for the first task you need to write a piece of code in python (or any language of your choice given that you can import hash libraries in it). In this piece of code, you will:

- a. Import hashlib and bcrypt libraries. For bcrypt use command “`pip install bcrypt`”.
- b. Take your name as an input string.
- c. Convert your name string to bytes using function `text.encode()`. Here “text” is the input string of your name.
- d. Generate MD5, SHA-1, SHA-256, SHA512, SHA-3, BLAKE2, bcrypt and RIPEMD-160 hashes of your name string.
- e. For bcrypt you can use the following piece of code:

```
salt = bcrypt.gensalt()
bcrypt_hash = bcrypt.hashpw(text.encode(), salt)
print("bcrypt Hash:", bcrypt_hash)
```

- In the code above, `bcrypt.gensalt()`: Generates a salt using the bcrypt hashing algorithm and assigns it to the variable `salt`. Where “salt” is a random value that is used as an additional input during the hashing process to make it more secure like a nonce.

- `bcrypt_hash = bcrypt.hashpw(text.encode(), salt):`
Generates the bcrypt hash of a text string using the provided salt and assigns it to the variable `bcrypt_hash`. Here “text” is the input string of your name.
- `print("bcrypt Hash:", bcrypt_hash):` Prints the string "bcrypt Hash:" followed by the value of `bcrypt_hash`.

f. For RIPEMD-160 you can use this:

```
ripemd160_hash = hashlib.new('ripemd160', text.encode()).hexdigest()
print("RIPEMD-160 Hash:", ripemd160_hash)
```

- Here `hashlib.new('ripemd160', text.encode())` creates a new hash object for the RIPEMD-160 algorithm.
- `hashlib` is a built-in Python module for hashing algorithms.
- `'ripemd160'` specifies the specific hashing algorithm to use, in this case, RIPEMD-160.
- `text.encode()` converts the string to bytes using the default encoding (UTF-8) since hash functions typically operate on bytes rather than strings.
- `hexdigest()` is a method that returns the hexadecimal representation of the calculated hash.
- The resulting hash value is assigned to the variable `ripemd160_hash`.
- And `print` just prints it out as discussed in the case of bcrypt hash.

5. Please repeat the above task taking the whole article of your choice from any newspaper. Personally I like this one ["https://thefinancialexpress.com.bd/views/reviews/silicon-valley-banks-collapse-what-happened-and-why-it-matters"](https://thefinancialexpress.com.bd/views/reviews/silicon-valley-banks-collapse-what-happened-and-why-it-matters).
6. Use the same code above and observe the “avalanche effect” by changing any value in the input string of the article.