

## 1.To what does a relative path refer ?

A relative path is a file path that is relative to the current working directory of a program or to the location of another file. It specifies the location of a file or directory relative to the location of the program or file that is using the path.

## 2.What does an absolute path start with your operating system?

The format of an absolute file path can vary depending on the operating system being used, but in general an absolute path starts with the root directory of the file system.

## 3.What do the functions `os.getcwd()` and `os.chdir()` do?

The `os` module in Python provides functions for working with the operating system, including functions for working with file paths and directories. `os.getcwd()` and `os.chdir()` are two such functions:

`os.getcwd()`: This function returns the current working directory (CWD) of the Python interpreter as a string. The current working directory is the directory in which the Python script or the Python REPL (interactive shell) is currently running. For example:

```
import os
```

```
cwd = os.getcwd() print(cwd) # Outputs the current working directory
```

`os.chdir(path)`: This function changes the current working directory to the directory specified by the path argument. The path argument can be an absolute path or a relative path. If the path is a relative path, it is interpreted relative to the current working

## 4.What are the `.` and `..` folders?

`.` Represents the Current Directory Whereas `..` Represents the Parent Directory of the Current Directory For Example: if the below path is my absolute path:

C:\Users\vishnu\Documents\iNeuron-Assignments\Python Basic Assignment Then `.` represents the path C:\Users\vishnu\Documents\iNeuron-Assignments\Python Basic Assignment Where as `..` represents the path C:\Users\vishnu\Documents\iNeuron-Assignments

## 5. In C:\bacon\eggs\spam.txt which part is the dir name and which part is the base name ?

For C:\bacon\eggs\spam.txt The dir name is C:\bacon\eggs The Base name is spam.txt

```
In [1]: import os
path = r'C:\bacon\eggs\spam.txt'
print(os.path.dirname(path))
print(os.path.basename(path))
```

```
C:\bacon\eggs
spam.txt
```

## 6. What are the three mode arguments that can be passed to the open() function ?

A file can be Accessed in python using open() function. open function takes two arguments filename and mode of operation (optional). if mode is not provided the default mode of opening is read mode  
So, the syntax being: open(filename, mode)

'r' - Read Mode: This is the default mode for open(). The file is opened and a pointer is positioned at the beginning of the file's content.  
'w' - Write Mode: Using this mode will overwrite any existing content in a file. If the given file does not exist, a new one will be created.  
'r+' - Read/Write Mode: Use this mode if you need to simultaneously read and write to a file.  
'a' - Append Mode: With this mode the user can append the data without overwriting any already existing data in the file.  
'a+' - Append and Read Mode: In this mode you can read and append the data without overwriting the original file.  
'x' - Exclusive Creating Mode: This mode is for the sole purpose of creating new files. Use this mode if you know the file to be written doesn't exist beforehand.

## 7. What happens if an existing file is opened in write mode ?

Using this mode will overwrite any existing content in a file. If the given file does not exist, a new one will be created.

## 8. How do you tell the difference between read() and readlines() ?

The main difference is that read() will read the whole file at once and then print out the first characters that take up as many bytes as you specify in the parenthesis

Whereas the `readline()` that will read and print out only the first characters that take up as many bytes as you specify in the parenthesis. You may want to use `readline()` when you're reading files that are too big for your RAM. The `read()` would treat each character in the file separately, meaning that the iteration would happen for every character. The `readline()` function, on the other hand, only reads a single line of the file. This means that if the first line of the file were three lines long, the `readline()` function would only parse (or iterate/operate) on the first line of the file.

## What data structure does a shelf value resemble ?

it contains key and values it represents dictionary

In [ ]: