1. To what does a relative path refer?

Ans-> The **path** with **reference** to current **directory is** called **relative**. **Relative paths** make use of two special symbols, a dot (.) and a double-dot (..), which translate into the current directory and the parent directory. ... The current directory is sometimes referred to as the root directory. **Relative path** is defined as the **path** related to the present working directly(pwd) .

1. What is the first step on an absolute path?

Ans-> An **Absolute path** of any directory always starts with a slash (/) representing the directory root. Besides this, all slashes in the directory **path** separate the directories. All directories names in the **absolute path** are written in the hierarchy order. **absolute path** is a complete **path** from start of actual **file** system from / directory. **Relative path**.

1. On Windows, what does Path('C:/Users') / 'Al' evaluate to?

Ans->The C:\ part of the path is the root folder, which contains all other folders. On Windows, the root folder is named C:\ and is also called the C: drive. Users, Al refer to folders (also called directories) ('C:/Users') is a home directory and ‘AI’ is subdirectories.

1. On Windows, what does 'C:/Users' / 'Al' evaluate to?

Ans->The C:\ part of the path is the root folder, which contains all other folders. On Windows, the root folder is named C:\ and is also called the C: drive. Users, Al refer to folders (also called directories)

1. What do the functions os.getcwd() and os.chdir() do?

Ans-> **getcwd()** : CWD stands for Current Working Directory. This **function** allows you to see what your current working directory is. **chdir**("path-to-dir") : Short for CHange DIRectory, this **function** allows you to set the current working directory to a path of your choice.

**getcwd() method tells us the location of current working directory (CWD).**

1. Syntax: os.**getcwd**()
2. Parameter: No parameter is required.
3. Return Value: This method returns a string which represents the current working directory.

**chdir**() changes the current working directory to the given path.It returns None in all the cases. Syntax :os.chdir(path)

#!/usr/bin/python

import os

path = "/usr/tmp"

# Check current working directory.

retval = os.getcwd()

print "Current working directory %s" % retval

# Now change the directory

os.chdir( path )

# Check current working directory.

retval = os.getcwd()

print "Directory changed successfully %s" % retval

When we run above program, it produces following result −

Current working directory /usr

Directory changed successfully /usr/tmp

1. What is the difference between the. and.. folders?

Ans-> The single **dot** is a meta-location, meaning the **folder** you are currently **in**. The **double dot** is an indicator that you can move back from this location. That is, you're **in a folder** inside **of** another **folder**. **Single dot** "." stands **for** current directory and **double dots** ".." mean parent directory. Similarly "~" stands **for** home directory. These shortcuts come in very handy because you don't have to type in complete path **of** any file you want to access **in a** particular (or current) directory. **For** Example- typing cd ..

1. Which part of C:baconeggsspam.txt is the dir name and which part is the base name?

Ans-> **C**:\bacon\eggs is the **dir name**, while spam. **txt** is the **base name**.

1. What are the three “mode” arguments that the open() function accepts?

Ans-> **There are three kinds of mode, that open function accept:**

* “ r “, for reading.
* “ w “, for writing.
* “ a “, for appending.
* “ r+ “, for both reading and writing.

**The access modes available for the open() function are as follows:**

* r : Opens the file in read-only **mode**. ...
* rb : Opens the file as read-only in binary format and starts reading from the beginning of the file. ...
* r+ : Opens a file for reading and writing, placing the pointer at the beginning of the fil

1. What happens if you open existing file in write mode?

Ans->To **open** a **file in write mode**, “w” is specified. **When mode** “w” is specified, **it** creates an empty **file** for output operations. What **if** the **file** already exists? **If** a **file** with the same name already exists, its contents are discarded and the **file** is treated as a new empty **file**.

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

Ans->The only **difference between** the **Read() and ReadLines() is** that Console. **Read is** used to **read** only single character from the standard output device, while Console. **ReadLine is** used to **read** a line or string from the standard output device.

**read()** would treat each character **in the** file separately, meaning that the iteration would happen for every character. The **readlines()** function, on the other hand, only **reads** a single or list of line of the file in string.

**If you have a file (test.txt) like so:**

**first line**

**second line**

**third line**

**Then this code:**

**with open("test.txt", "r") as file:**

**line = file.readline()**

**print(line)**

**Will produce this output:**

**first line**

**That's because readline just reads the next line.**

**If you use this code instead:**

**with open("test.txt", "r") as file:**

**content = file.read()**

**print(content)**

**Output:**

**first line**

**second line**

**third line**

**read() reads the entire contents of the file into a string. You can also give read() an optional argument, which designates the number of characters to read from the file:**

**with open("test.txt", "r") as file:**

**content = file.read(15)**

**print(content)**

**Output:**

**first line**

**seco**

**readlines(), which returns a list of lines (strings):**

**with open("test.txt", "r") as file:**

**lines = file.readlines()**

**print(lines)**

**Output:**

**['first line\n', 'second line\n', 'third line\n']**

1. How does a shelf value resemble a data structure?

Ans-> A **shelf value resembles** a dictionary **value**; it has keys and **values**, along with keys() and **values**() methods that work similarly to the dictionary methods of the same names.

Data structures are **a way to organize information in a computer's memory**. It facilitate storage and retrieval of information In Arrays and lists ,Objects, hash maps and dictionaries indexing is used so, shelf value resemble a data structure.