1. To what does a relative path refer?

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

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

4. What are the . and .. folders?

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

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

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

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

9. What data structure does a shelf value resemble?

**Answers**

**Relative Path**:

* + A relative path refers to a file or directory location **relative to the current working directory**. It doesn’t start from the root of the filesystem but instead assumes your current position.
  + [For example, if your current working directory is /home/user/, a relative path like ../documents/report.txt would point to the file report.txt in the documents directory one level up from your current location1](https://www.redhat.com/sysadmin/linux-path-absolute-relative).

1. **Absolute Path**:
   * An absolute path starts from the **root folder** (e.g., / in Unix-like systems or C:\ in Windows) and specifies the complete path to a file or directory.
   * [For instance, the absolute path /usr/share/widgets/button always points to the same location, regardless of where you start in the filesystem2](https://phoenixnap.com/kb/absolute-path-vs-relative-path).
2. **os.getcwd() and os.chdir()**:
   * os.getcwd() returns the **current working directory** (where your Python script is executing).
   * os.chdir(path) changes the current working directory to the specified path.
   * Example:
   * import os
   * current\_dir = os.getcwd()
   * print(f"Current directory: {current\_dir}")
   * os.chdir("/usr/share/widgets")
   * print(f"Changed directory: {os.getcwd()}")
3. **. and … Folders**:
   * The . folder represents the **current directory**, and .. represents the **parent directory**.
   * [For instance, if you’re in /home/user/documents, . refers to /home/user/documents, and .. refers to /home/user](https://www.redhat.com/sysadmin/linux-path-absolute-relative)[3](https://quizlet.com/499228518/chapter-8-python-flash-cards/).
4. **C:\bacon\eggs\spam.txt**:
   * In this path:
     + The **directory name** (dir name) is C:\bacon\eggs.
     + The **base name** is spam.txt.
5. **Mode Arguments for open()**:
   * The three common modes are:
     + 'r': Read mode (default). Opens a file for reading.
     + 'w': Write mode. Opens a file for writing (creates a new file or overwrites an existing one).
     + 'a': Append mode. Opens a file for writing (appends to an existing file or creates a new one).
6. **Opening an Existing File in Write Mode**:
   * If you open an existing file in write mode ('w'), its content will be **overwritten**. Be cautious!
7. **read() vs. readlines()**:
   * read() reads the entire file content as a single string.
   * readlines() reads the file line by line and returns a list of strings (each line).
8. **Shelf Value Resembles**:
   * A shelf value in Python resembles a **persistent dictionary**. It’s used for storing key-value pairs in a file, allowing data to persist across program runs.