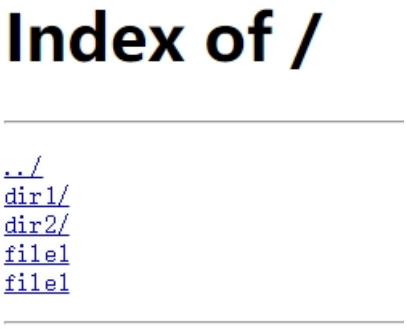
Assignment 4.2 Web File Browser

In this assignment, you need to use python3 asyncio stream to create a web file browser. HTTP/1.0 should be used, because HTTP/1.1 supports keep-alive, which makes our implementation more complex.
When running in a directory, the home page of your server should be the list of files and sub directories. The functions should include: browsing directory, jumping, and open files. Editing directory and files are not asked to supported
An example is given like this.



I. Logic
   i. Your server should only support GET/HEAD method. For more details about the difference between GET and HEAD method, please read the Lecture 03 slides.
   When receive other method (POST etc.), you should return an error code **405 Method Not Allowed**.
   ii. Remember to add Connection: close to your response header.
   iii. When receive a request path, check if it is a directory.
   You might want to add './' to the client path to make sure you are working on current directory.*

```
path = './' + header.get('path')
```

   If the path is a directory, render a html page with files and directories listed in it.
      i. If the path is a file, return the file with a correct mime type. If server cannot decide the mime type from the file extension (.exe, .mp3, etc.), the mime type should be application/octet-stream.
      ii. If the path doesn't exist, return an error 404 Not Found.

* pathlib might be helpful for doing path operation. https://docs.python.org/3/library/pathlib.html

II. Functions you might need

i. os.listdir

This can list current directory or a given path.

ii. os.path.isfile

This function can check if a path is a file.

```
Python 3.6.6 (default, Jun 28 2018, 10:27:26)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: import os

In [2]: os.listdir()
Out[2]:
['.idea',
 '.vscode',
 'asyncio_web_hello.py',
 'asyncio_web_withparse.py',
 'echo.py',
 'echo_multithreading.py',
 'parse_header.py',
 'web_hello.py',
 '__pycache__']

In [3]: os.path.isfile('echo.py')
Out[3]: True
```

iii. os.path.getsize

This function can get the filesize in bytes, which might be useful if you want to add **Content-Length** to your response header.

iv. open

Example:

```
file = open(name)
writer.write(file.read())
```

Calling these function may raise `FileNotFoundError` Exception, you can organize your code like follow example:

```
try:
    # some server logic
except FileNotFoundError:
    # write 404 Not Found Response
```

III. HTML Example

```
<html><head><title>Index of .//</title></head>
<body bgcolor="white">
<h1>Index of .//</h1><hr>
<pre>
<a href="dir1/">dir1/</a><br>
<a href="dir2/">dir2/</a><br>
<a href="file1">file1</a><br>
<a href="file1">file1</a><br>
</pre>
<hr>
</body></html>
```