Assignment 3

Python Modules

os

- This module implements some useful functions on accessing the filesystem.
- **OSError** in the case of ths, or other arguments that have , but are not accepted by the operating sys

os

os.**getcwd**()

Return a string representing the current working directory

os.**li**

R entries in the directory given by *path*.

os.**mkdir**(*path*)

Create a directory named *path*

os.**rmdir**(*path*)

Remove (delete) the directory *path.* The directory should be empty.

os

os.**chmod**(*path, mode*)

Change the mode of *path* to the numeric *mode*.

os.**c**

Change the owner and gr o the numeric *uid* and

gid. To leave one of thei et it to -1.

—

0777 (octal).

os

os.**open**(*file*, *flags*[, *mode*])

Open the file *file* and set various flags according to *flags* and poss . Return the file descriptor for t

Some Open() flag constan

os.**O_RDONLY**

os.**O_WRONLY**

os.**O_RDWR**

os.**O_CREAT**

os

os.**write**(*fd*, *str*)

Write the string *str* to file descriptor *fd*. Return the number of byte

os.**access**(*path*, *mode*)

To check if the user is au ss to *path.* Return True if access is allowed, False if not.

os.**lstat**(*path*)

Perform the equivalent of an lstat() system call on the given *path*

filecmp

filecmp.**cmp**(*f1*, *f2*[, *shallow*])

Compare the files named *f1* and *f2*, returning **True** if they seem equ

By default, cmp() compar ing inside the files. The shallow argument tells cmp() whether to look at the contents of the file, as well. - filecmp.**cmp**(*f1*, *f2*, False)

shutil

shutil.**copyfile**(*src*, *dst*)

Cop file named *src* to a file

nam https://eduassistpro.github.io/ de replaced. *src* and *dst* are

path names given as strin

shutil.**rmtree**(path)

Delete an entire directory tree.

glob

glob.glob(*pathname*)

For a ters in brackets. For exa

re

This module provides regular expression matching operations

The functions in this module could check if a particular string matches a give

Reg

specifies a set of string pattern

e.g.

- '**+**': indicates *one or more* occurrences of the preceding element.
- `ab+c` matches "abc", "abbc", "abbbc", and so on, but not "ac".

re

re.**compile**(pattern, flags=0)

Compile a regular expression pattern, returning a pattern obje

The expression's behavio d by specifying a ***flags*** value.

E.g. re.**I**: Ignore case.

combined using bitwise OR (the | operator).

natches the pattern

re

re.**search**(pattern, string, flags=0)

- Scan through string looking for the first location where the regular expression pattern produces a match
-
-

re.**match**(pattern, string,

- If zero or more characters string match the regular expression pattern
- Return a corresponding match object
- Return None if the string does not match the pattern;

re

re.**match**(pattern, string, flags=0)

e.g.

prog = re.compile(pattern)

result = prog.match(string)

but using re.compile() and saving the resulting regular expression object for reuse is more efficient when the expression will be used several times in a single program.

os.path

os.path.**abspath**(*path*)

Return a normalized absolutized version of the pathname *path*.

```
>>> os.path.dirname("/usr/local/python3/bin/python")
'/usr/local/python3/bin/python'
```

os.p

Return the base name of

```
>>> me("/usr/local/python3/bin/python")
'python'
```

os.path.**dirname**(*path*)

Return the directory name of pathname *path*.

```
>>> os.path.dirname("/usr/local/python3/bin/python")
'/usr/local/python3/bin/'
```

os.path

os.path.**join**(*path*, **paths*)

- Join one or more path components intelligently.
- *path* and any members of os.sep) following each non-empty part except t at the result will only end in a separator if the last part


```
>>> os.path.join("/tmp",file)
/tmp/file
```

- If a component is an absolute path, all previous components are thrown away and joining continues from the absolute path component.

os.path

os.path.**exists**(*path*)

- Return True if *path* refers to an existing path or an open file descriptor.
- Returns False for broken symbolic links.

os.p

- ry.
- This follows symbolic link d isdir() can be true for the same path.

os.path.**islink**(*path*)

Return True if *path* refers to an existing directory entry that is a symbolic link.