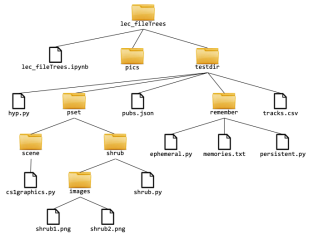
# File System Operations and File Tree Traversal




**CS111 Computer Programming**

Department of Computer Science
Wellesley College

## A trip to "The Office"



## Folders for organizing files

An empty folder

A folder with one document (file)

Nested folders

A folder with many files

Several folders with several files



## File Cabinets

An single-office's file cabinet

An organization's file cabinet

# Graphical User Interfaces

Susan Kare

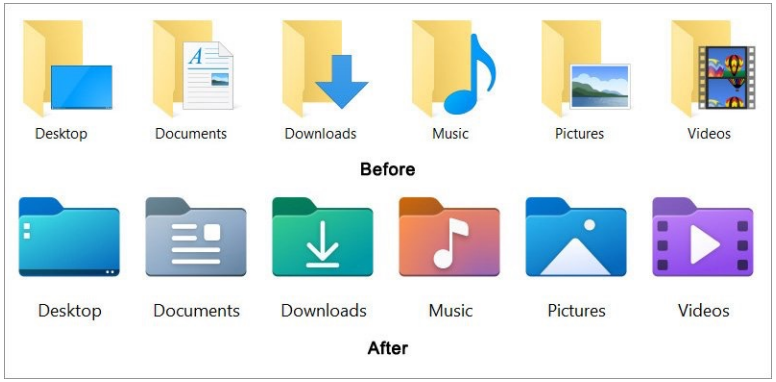
Graphic designer at Apple (1983-1986).

The original Mac OS interface icons, created by Susan Kare.

Read about Susan's work in this New Yorker article: https://www.newyorker.com/culture/cultural-comment/the-woman-who-gave-the-macintosh-a-smile

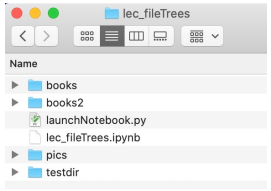


# Windows OS Folder Icons (over the years)

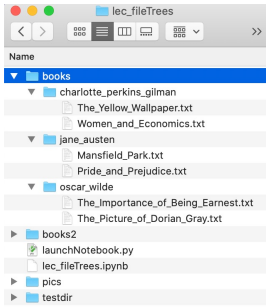




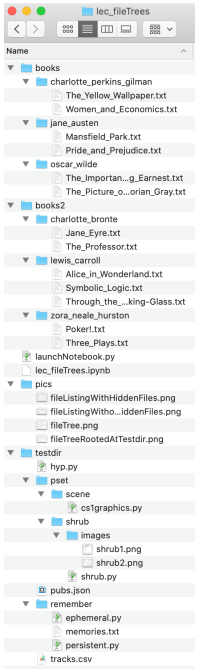

# Directory = Folder Expanded Folder View




Our folder lec_fileTrees




Expanding subfolders in the lec_fileTrees folder.

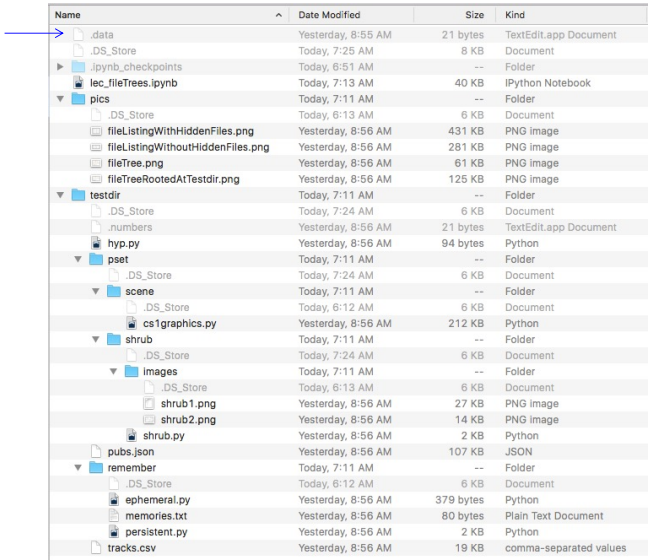



**Concepts in this slide**: Files in a computer are organized in nested folders.



# Sample Directory: Folder View Showing Dot Files

**Concepts in this slide**: Some files, known as dot files, are "hidden".




**To Notice**

Hidden files/folders start with the dot character, e.g., .data.

# Operating Systems (OS) Commands: Navigating around

**Concepts in this slide**: Before GUIs there was the command line for text commands.

```
lec_fileTrees — -zsh — 73×24
(base) emustafa@emu-nsf ~ %
(base) emustafa@emu-nsf ~ % pwd
/Users/emustafa
(base) emustafa@emu-nsf ~ % cd Documents
(base) emustafa@emu-nsf Documents % cd CS111-Spring22
(base) emustafa@emu-nsf CS111-Spring22 % cd lec_fileTrees
(base) emustafa@emu-nsf lec_fileTrees % ls
books                   launchNotebook.py       pics
books2                  lec_fileTrees.ipynb     testdir
(base) emustafa@emu-nsf lec_fileTrees % ls -al
total 112
drwxr-xr-x@ 10 emustafa  staff    320 Mar 26 09:01 .
drwxr-xr-x@ 14 emustafa  staff    448 Mar 26 09:13 ..
-rw-r--r--@  1 emustafa  staff   6148 Mar 26 09:11 .DS_Store
-rw-r--r--   1 emustafa  staff     21 Mar 26 08:58 .data
drwxr-xr-x   6 emustafa  staff    192 Mar 26 08:57 books
drwxr-xr-x   6 emustafa  staff    192 Mar 26 08:57 books2
-rw-r--r--   1 emustafa  staff  10697 Nov  2 16:42 launchNotebook.py
-rw-r--r--   1 emustafa  staff  30902 Nov  2 16:42 lec_fileTrees.ipynb
drwxr-xr-x   6 emustafa  staff    192 Aug 13  2021 pics
drwxr-xr-x   8 emustafa  staff    256 Mar 26 08:57 testdir
(base) emustafa@emu-nsf lec_fileTrees % cd books
```

This window is the Mac app "Terminal".

It only accepts text commands in the command line.

The shown commands (in red):

`pwd`

`cd`

`ls`

`ls -al`

As a result of the `cd` command, we get to move inside that folder, indicated by the name on the left of %



# OS commands

Here is what the previously shown OS commands mean:

| | |
|---|---|
| `pwd` | – display the name of the parent working directory |
| `cd` | – change directory (to the provided argument) |
| `ls` | – list the content of the current directory |
| `ls -al` | – list all content of the current directory one line at at time, with extra info |
| `mkdir` | – make a new directory |
| `more` | – view the content of a text file |

Some of these commands take arguments, as shown in the previous slide.

After the last command, `cd books`, we can check with `pwd` the absolute path in which we have arrived (see screenshot below):

```
(base) emustafa@emu-nsf lec_fileTrees % cd books
(base) emustafa@emu-nsf books % pwd
/Users/emustafa/Documents/CS111-Spring22/lec_fileTrees/books
```



# Where do you live? Absolute vs Relative

To: Wendy Wellesley
Unit 1025
21 Wellesley College Rd
Wellesley MA 02481 -0210
[absolute address]

Cazenove Hall (Caz), 3rd floor
[relative address]

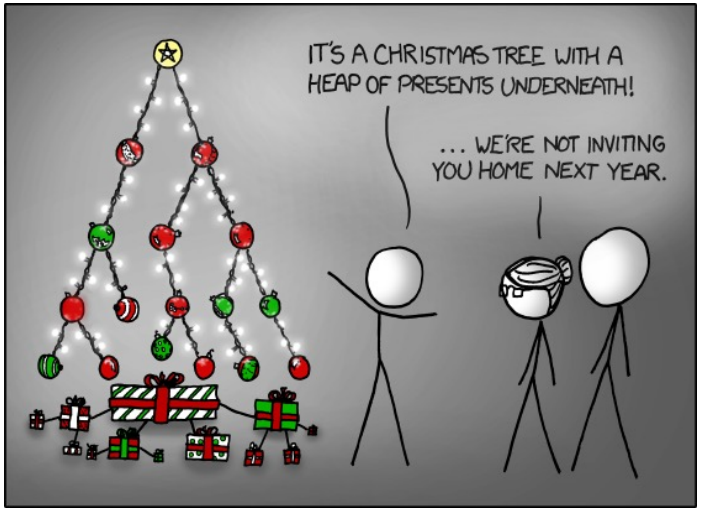
relative path

```
(base) emustafa@emu-nsf lec_fileTrees % cd books
(base) emustafa@emu-nsf books % pwd
/Users/emustafa/Documents/CS111-Spring22/lec_fileTrees/books
```

absolute path

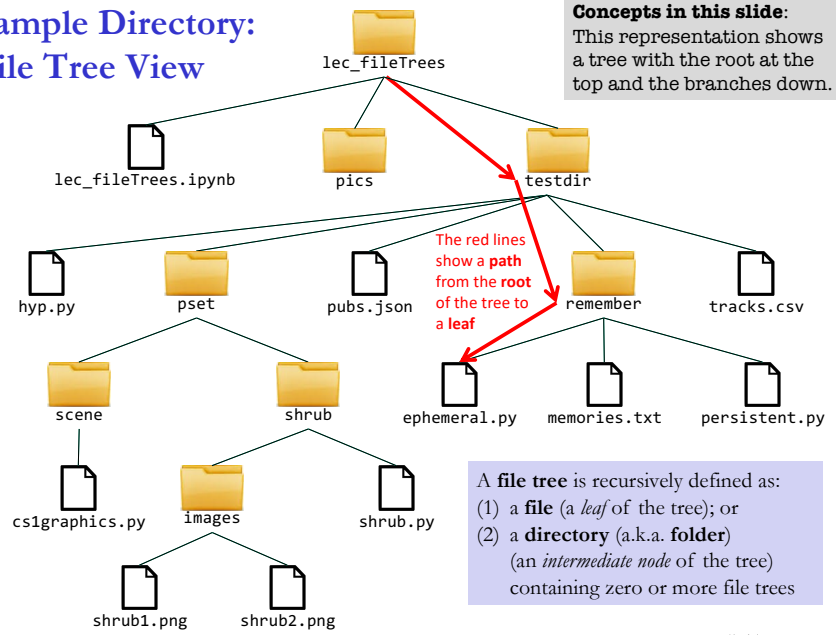


# Trees – a data structure in CS



# Sample Directory: File Tree View

**Concepts in this slide**: This representation shows a tree with the root at the top and the branches down.




A **file tree** is recursively defined as:
(1) a **file** (a *leaf* of the tree); or
(2) a **directory** (a.k.a. **folder**) (an *intermediate node* of the tree) containing zero or more file trees



# File System Operations: `os.getcwd`

**Concepts in this slide**: The Python module `os` has all functionalities needed to operate with files and folders.

Via the **`os`** module (os = operating system), Python provides a way to manipulate the directories and files in a file system. To use these features, we first need to import the **`os`** module:

```
import os
```

The **`os.getcwd`** function returns the **current working directory** as a string. This is the directory that the Python program is currently "connected" to. All **relative** file names will be interpreted **relative** to this directory.

```
In [1]: import os

In [2]: os.getcwd()
Out[2]: '/Users/wendy/Downloads/lec_fileTrees'
```

components of a **file path** are separated by /

**To Notice**
The function **`getcwd`** returns an **absolute** path, containing all directory names in the path to the current one. A path name that is not absolute is known as **relative**.



# File System Operations: `os.listdir`

The **`os.listdir`** function returns a list of all files/directories in the argument directory.

**`os.listdir`** shows "dot files" that are often hidden on Macs

```
In [3]: os.listdir(os.getcwd())
Out[3]: ['.DS_Store', '.data', '.ipynb_checkpoints', 'lec_fileTrees.ipynb',
'pics', 'testdir']

In [4]: os.listdir('.')
Out[4]: ['.DS_Store', '.data', .ipynb_checkpoints', 'lec_fileTrees.ipynb',
'pics', 'testdir']

In [5]: os.listdir('testdir')
Out[5]: ['.DS_Store', '.numbers', 'hyp.py', 'pset', 'pubs.json', 'remember',
'tracks.csv']

In [6]: os.listdir('testdir/remember')
Out[6]: ['.DS_Store', 'ephemeral.py', 'memories.txt', 'persistent.py']

In [7]: os.listdir('testdir/pset')
Out[7]: ['.DS_Store', 'scene', 'shrub']

In [8]: os.listdir('testdir/pset/shrub')
Out[8]: ['.DS_Store', 'images', 'shrub.py']

In [9]: os.listdir('testdir/pset/shrub/images')
Out[9]: ['.DS_Store', 'shrub1.png', 'shrub2.png']
```

`'.'` is a synonym for the current directory

components of a **file path** are separated by /

`'.DS_Store'` is a data file for a folder in macOS

**To Notice**
We used **relative** pathnames in lines 4 to 9. These are relative with respect to the current directory in which we are in.



# File System Operations: `os.path.exists`

The **`os.path.exists`** function determines whether the given name denotes a file/directory in the filesystem. It returns a Boolean value.

```
In [10]: os.path.exists('testdir/remember/memories.txt')
Out[10]: True

In [11]: os.path.exists('testdir/remember/catPlaysPiano.png')
Out[11]: False

In [12]: os.path.exists('testdir/remember')
Out[12]: True

In [13]: os.path.exists('remember/memories.txt')
Out[13]: False # this is not a path from working directory

In [14]: os.path.exists('memories.txt')
Out[14]: False # this is not a path from working directory
```

## File System Ops: `os.path.isfile` & `os.path.isdir`

**`os.path.isfile`** and **`os.path.isdir`** determine whether the given name is a file or directory, respectively. If the file does not exist, these return **`False`**.

```
In [15]: os.path.isdir('testdir/remember')
Out[15]: True

In [16]: os.path.isfile('testdir/remember')
Out[16]: False

In [17]: os.path.isdir('testdir/remember/memories.txt')
Out[17]: False

In [18]: os.path.isfile('testdir/remember/memories.txt')
Out[18]: True

In [19]: os.path.isdir('memories.txt')
Out[19]: False

In [20]: os.path.isfile('memories.txt')
Out[20]: False
```

These return **`False`** because we did not provide the relative path to the file. If no path is given, Python looks for it in the current working directory.



## File System Ops: `os.path.join` & `os.path.basename`

**`os.path.join`** is a clearer and less error-prone way of joining directories and a filename into a path than concatenating strings with **`'/'`**

```
In [21]: os.path.join('testdir', 'remember')
Out[21]: 'testdir/remember'

In [22]: os.path.join('testdir', 'remember', 'memories.txt')
Out[22]: 'testdir/remember/memories.txt'
```

**`os.path.basename`** returns the last component of a file path.

```
In [23]: os.path.basename('testdir/remember/memories.txt')
Out[23]: 'memories.txt'

In [24]: os.path.basename('testdir/remember')
Out[24]: 'remember'

In [25]: os.path.basename('testdir/remember/')
Out[25]: ''
```



## File System Operations: `path.getsize`

The **`os.path.getsize`** function returns the size of the file (in bytes). For text files, this is the number of characters.

```
In [26]: os.path.getsize('testdir/remember/memories.txt')
Out[26]: 80

In [27]: os.path.getsize('testdir/remember/persistent.py')
Out[27]: 1634

In [28]: os.path.getsize('testdir/pset/shrub/images/shrub1.png')
Out[28]: 27248

In [29]: os.path.getsize('testdir/pset/shrub/images')
Out[29]: 136
```

The size of a directory is related to the "meta information" the directory holds for subdirectories and files. It is **not** the sum total of the sizes of the contained files.



## File System Operations: A summary

**Concepts in this slide**: All functions from the **`os`** module that we encountered in this lecture.

| Function Name | Description |
|---|---|
| **`os.getcwd`** | Get current working directory (shows its full name) |
| **`os.listdir`** | List directory (list the names of files and subfolders within it) |
| **`os.chdir`** | Change directory to the provided argument. |
| **`os.path.exists`** | Returns true if the provided argument exists as a file or directory. |
| **`os.path.isfile`** | Returns true if the provided argument corresponds to a file. |
| **`os.path.isdir`** | Returns true if the provided argument corresponds to a directory. |
| **`os.path.join`** | Join any number of path components into a path name |
| **`os.path.basename`** | Returns the last component of a file or directory path |
| **`os.path.getsize`** | Returns an integer value, the size in bytes of a file or directory. |

There are many other file system operations. See the documentation at https://docs.python.org/3/library/os.html and https://docs.python.org/3/library/os.path.html