Usage of os.path.join()

In [10]:
```
import os
#path usr\bin\spam-Windows
#path usr/bin/spam
fpath=os.path.join('usr', 'bin', 'spam')
print(type(fpath))
```

```
<class 'str'>
```

In [11]:
```
#Program to create strings for filenames with specific path.
myFiles = ['accounts.txt', 'details.csv', 'invite.docx']
for filename in myFiles:
    print(os.path.join('C:','Users','asweigart', filename))
```

```
C:Users\asweigart\accounts.txt
C:Users\asweigart\details.csv
C:Users\asweigart\invite.docx
```

The Current Working Directory

In [12]:
```
import os
print(os.getcwd())
```

```
C:\Users\Dell
```

In [13]:
```
os.chdir('C:\\t')
os.getcwd()
```

Out[13]:
```
'C:\\t'
```

In [14]:
```
os.chdir('C:\\Users\\Dell')
os.getcwd()
```

Out[14]:
```
'C:\\Users\\Dell'
```

In [15]:
```
os.chdir('C:\\ThisFolderDoesNotExist')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-15-b3529b3e26d4> in <module>
----> 1 os.chdir('C:\\ThisFolderDoesNotExist')

FileNotFoundError: [WinError 2] The system cannot find the file specified: 'C:\\ThisFolderDoesNotExist'
```

Creating New Folders with os.makedirs()

In [ ]:
```
import os
os.makedirs('C:\\delicious\\walnut\\waffles')
```

Usage of os Module

In [17]:
```
#return the file size in bytes
os.path.getsize('C:\\Windows\\System32\\calc.exe')
```

Out[17]:
```
27648
```

In [18]:
```
os.listdir('C:\\Windows\\System')
```

Out[18]:
```
['Speech']
```

In [60]:
```
totalSize = 0
for filename in os.listdir('C:\\Windows\\System32'):
    totalSize = totalSize + os.path.getsize(os.path.join('C:\\Windows\\System32', filename))
print(totalSize)
```

```
2289522528
```

In [20]:
```
os.path.exists('C:\\Windows')
```

Out[20]: True

In [21]:
```
os.path.exists('C:\\some_made_up_folder')
```

Out[21]: False

In [22]:
```
os.path.isdir('C:\\Windows\\System32')
```

Out[22]: True

In [23]:
```
os.path.isfile('C:\\Windows\\System32')
```

Out[23]: False

In [24]:
```
os.path.isdir('C:\\Windows\\System32\\calc.exe')
```

Out[24]: False

In [25]:
```
os.path.isfile('C:\\Windows\\System32\\calc.exe')
```

Out[25]: True

In [26]:
```
os.getcwd()
```

Out[26]: 'C:\\Users\\Dell'

In [27]:
```

os.path.exists('D:\\')
```

Out[27]: False

In [28]:
```
#Relative path: . represents current working directory
os.path.abspath('.')  # relative -->absolute
```

Out[28]: 'C:\\Users\\Dell'

In [29]:
```
os.path.abspath('.\\Scripts')# relative one to absolute path c:\users\dell\scripts
```

Out[29]: 'C:\\Users\\Dell\\Scripts'

In [30]:
```
os.path.isabs('.') #return False, arg is relative path
```

Out[30]: False

In [31]:
```
os.path.isabs(os.path.abspath('.'))
```

Out[31]: True

In [32]:
```
path = 'C:\\Windows\\System32\\calc.exe'
os.path.basename(path)
```

Out[32]: 'calc.exe'

In [33]:
```
os.path.dirname(path)
```

Out[33]: 'C:\\Windows\\System32'

In [34]:
```
calcFilePath = 'C:\\Windows\\System32\\calc.exe'
os.path.split(calcFilePath)  #tuple of dir name and base name
```

Out[34]:
```
('C:\\Windows\\System32', 'calc.exe')
```

In [35]:
```
os.path.sep
```

Out[35]:
```
'\\'
```

In [36]:
```
calcFilePath.split(os.path.sep)
```

Out[36]:
```
['C:', 'Windows', 'System32', 'calc.exe']
```

In [37]:
```
calcFilePath.split('\\')  #split returns a list of strings splited by default is space
```

Out[37]:
```
['C:', 'Windows', 'System32', 'calc.exe']
```

Reading and Writing Files

In [43]:
```
#To open in read mode, second arg='r'-- default is read mode
#returns file object
helloFile = open('C:\\Users\\Dell\\guests.txt')
type(helloFile)
```

Out[43]:
```
_io.TextIOWrapper
```

In [39]:
```
#Entire Contents of a file as a single string value
helloContent = helloFile.read()
```

In [40]:
```
type(helloContent)
```

Out[40]:
```
str
```

In [41]:
```
print(helloContent)
```

```
Mamatha
Sreelatha
Santhosh
Deeksha
Meghashree
V Semester
```

In [44]:
```
#returns list of strings, where each string represents each line
helloFile.readlines()
```

Out[44]:
```
['Mamatha\n',
 'Sreelatha\n',
 'Santhosh\n',
 'Deeksha\n',
 'Meghashree\n',
 'V Semester']
```

In [45]:
```
baconFile = open('bacon.txt', 'w')
baconFile.write('Hello world!\n')
baconFile.close()
```

In [46]:
```
baconFile = open('bacon.txt', 'a')
baconFile.write('Bacon is not a vegetable.')
baconFile.close()
```

In [47]:
```
baconFile = open('bacon.txt')
content = baconFile.readlines()
baconFile.close()
print(content)
```

```
['Hello world!\n', 'Bacon is not a vegetable.']
```

Usage of shelve module

In [48]:
```
import shelve

shelfFile = shelve.open('mydata')

#data
cats = ['Zophie', 'Pooka', 'Simon']

shelfFile['cats'] = cats
shelfFile['grade'] ='A'
shelfFile['names']=('Mamatha','Priya','Ramu')

shelfFile.close()
```

In [49]:
```
shelfFile = shelve.open('mydata')
type(shelfFile)
```

Out[49]:
```
shelve.DbfilenameShelf
```

In [50]:
```
shelfFile['a']
#shelfFile.close()
```

Out[50]:
```
1
```

In [51]:
```
list(shelfFile.keys())
```

Out[51]:
```
['cats', 'a', 'grade', 'names']
```

In [52]:
```
list(shelfFile.values())
```

Out[52]:
```
[['Zophie', 'Pooka', 'Simon'], 1, 'A', ('Mamatha', 'Priya', 'Ramu')]
```

In [53]:
```
import pprint
cats = [{'name': 'Zophie', 'desc': 'chubby'}, {'name': 'Pooka', 'desc': 'fluffy'}]
pprint.pformat(cats)
```

Out[53]:
```
"[{'desc': 'chubby', 'name': 'Zophie'}, {'desc': 'fluffy', 'name': 'Pooka'}]"
```

In [54]:
```
import pprint
fileObj=open('myCats.py','w')
fileObj.write("cats=" + pprint.pformat(cats) + '\n')
fileObj.close()
```

In [55]:
```
import myCats
myCats.cats
```

Out[55]:
```
[{'desc': 'chubby', 'name': 'Zophie'}, {'desc': 'fluffy', 'name': 'Pooka'}]
```

In [9]:
```
print(myCats.cats[0])
print(myCats.cats[0]['name'])
print(myCats.name)
```

```
{'desc': 'chubby', 'name': 'Zophie'}
Zophie
Mamatha
```

MultiClipBoard Program

In [ ]:
```
import shelve, pyperclip, sys
mcbShelf = shelve.open('mcb')
# TODO: Save clipboard content.
if len(sys.argv) == 3 and sys.argv[1].lower() == 'save':
    mcbShelf[sys.argv[2]] = pyperclip.paste()
elif len(sys.argv) == 2:
    # List keywords and load content.
    if sys.argv[1].lower() == 'list':
        pyperclip.copy(str(list(mcbShelf.keys())))
    elif sys.argv[1] in mcbShelf:
        pyperclip.copy(mcbShelf[sys.argv[1]])
mcbShelf.close()
```