Python

Browsing Directories Using `walk`
```python
>>> from os import walk
>>> tree = walk('..')
```
Python

Browsing Directories Using `walk`
Browsing Directories Using `walk`
Python

Browsing Directories Using `walk`
A

a1.txt  a2.txt

B

b.txt

P

P

Q

q1.txt  q2.txt

c.txt

.  ['C', 'A', 'B']  []
Python

Browsing Directories Using `walk`
Browsing Directories Using `walk`

```
walk('.',)
walk('./C')
```

- File list: a1.txt, a2.txt, b.txt, p.txt, q1.txt, q2.txt, c.txt
- Directory list: ['C', 'A', 'B']
Python

Browsing Directories Using

```
walk('.
walk('./C')
```

To navigate through directories, you can use the `walk` function provided by the `os` module in Python. This function helps in exploring the directory tree, and it takes a directory path as input.

In the example provided:

- `A` contains `a1.txt` and `a2.txt`
- `B` contains `b.txt`
- `C` contains `c.txt`, `p.txt`, `q1.txt`, `q2.txt`,
  - The directory `./C` contains files `p.txt`, `q1.txt`, and `q2.txt`
  - The directory `C` contains files `p.txt`, `q1.txt`, and `q2.txt`

The code `walk('.',) walk('./C')` would explore all the directories and files starting from the current directory and then from `./C` directory respectively.
Browsing Directories Using `walk`

```python
walk('..')
wakl('./C')
```
A
a1.txt  a2.txt

B
b.txt

C

P

Q
p.txt  q1.txt  q2.txt
c.txt

walk('.')
walk('./C')

[['C', 'A', 'B'], []]
[['C', 'C', 'A', 'B'], ['c.txt']]

Python Browsing Directories Using walk
Browsing Directories Using `walk`

```
walk('..')
wakl('./C')
```

```
.   [ 'C', 'A', 'B' ]
./C  []
    [ 'c.txt' ]
```

- A
  - a1.txt
  - a2.txt
- B
  - b.txt
- C
  - c.txt
  - p.txt
  - q1.txt
  - q2.txt
- P
- Q
Python

Browsing Directories Using `walk`
Browsing Directories Using `walk`

```
walk('.')
walk('./A')
```

```
[ 'C', 'A', 'B' ]
[
]
[ 'c.txt'
]
```
Python

Browsing Directories Using `walk`

```python
walk('..')
walk('./A')
```

```
.      ['C', 'A', 'B']     []
./C    []                   ['c.txt']
./A    []                   ['a1.txt', 'a2.txt']
```
```python
walk('.')
walk('./A')
```

```
['C', 'A', 'B']
['c.txt']
['a1.txt', 'a2.txt']
```
Python

Browsing Directories Using `walk`
Browsing Directories Using `walk`

```
walk('..')
walk('./B')
```

```
.  ['C', 'A', 'B']  []
./C  []  ['c.txt']
./A  []  ['a1.txt', 'a2.txt']
```
Browsing Directories Using

Python

walk('...')
wakl('.../B')

A

B

C

a1.txt  a2.txt

b.txt

P  Q

p.txt  q1.txt  q2.txt

c.txt

.  ['C', 'A', 'B']  []
./C  []  ['c.txt']
./A  []  ['a1.txt', 'a2.txt']
./B  ['P', 'Q']  ['b.txt']
Browsing Directories Using `walk`

```python
walk('.
walk('./B')
```

```bash
./B

.'
['C', 'A', 'B']

./C
[]

./A
[]

./B
['P', 'Q']
```
Python

Browsing Directories Using `walk`

```
walk('..')
wwalk('./B')
wwalk('./B/P')
```

```
.
./C
./A
./B

['C', 'A', 'B']

['c.txt']

['a1.txt', 'a2.txt']

['b.txt']
```
Browsing Directories Using `walk`

```python
walk('./')
walk('./B')
walk('./B/P')
```
Browsing Directories Using `walk`

```python
walk('.')
walk('=./B')
```

```
.    ['C', 'A', 'B']    []
./C  []                  ['c.txt']
./A  []                  ['a1.txt', 'a2.txt']
./B  ['P', 'Q']          ['b.txt']
./B/P []                  ['p.txt']
```
Browsing Directories Using `walk`

```python
walk('..')
walk('./B')
walk('./B/Q')
```
Python

Browsing Directories Using `walk`

```python
cwd = '.
walk(cwd)
walk('./B')
```

```
.      ['C', 'A', 'B']         []
./C    []                     ['c.txt']
./A    []                     ['a1.txt', 'a2.txt']
./B    ['P', 'Q']             ['b.txt']
./B/P  []                     ['p.txt']
./B/Q  []                     ['q1.txt', 'q2.txt']
```
Python

Browsing Directories Using walk

```
walk('.')
```

```
.      ['C', 'A', 'B']      []
./C    []                    ['c.txt']
./A    []                    ['a1.txt', 'a2.txt']
./B    ['P', 'Q']            ['b.txt']
./B/P  []                    ['p.txt']
./B/Q  []                    ['q1.txt', 'q2.txt']
```
Browsing Directories Using `walk()`

```
walk('.')

.
./C
./A
./B
./B/P
./B/Q

['C', 'A', 'B']
[
]
[
]
['c.txt']
['a1.txt', 'a2.txt']
['b.txt']
['p.txt']
['q1.txt', 'q2.txt']
```
```python
>>> from os import walk
>>> tree = walk('..')
```
```python
>>> from os import walk
>>> tree = walk('..')
```

`walk` returns a list of tuples.
>>> from os import walk
>>> tree = walk('..')

>>> for dir,subdirs,files in tree:
...     print "%s %s %s" %(dir,subdirs,files)
...
```python
>>> from os import walk
>>> tree = walk('.

>>> for dir,subdirs,files in tree:
...     print "%s   %s   %s" %(dir,subdirs,files)
...
.   ['C', 'A', 'B']   []
./C   []   ['c.txt']
./A   []   ['a1.txt', 'a2.txt']
./B   ['P', 'Q']   ['b.txt']
./B/P  []   ['p.txt']
./B/Q  []   ['q1.txt' 'q2.txt']
```
>>> from os import walk
>>> tree = walk('.

Each tuple contains a directory

>>> for dir, subdirs, files in tree:
...     print "%s   %s   %s" %(dir, subdirs, files)
...
.    ['C', 'A', 'B']    []
./C    []    ['c.txt']
./A    []    ['a1.txt', 'a2.txt']
./B    ['P', 'Q']    ['b.txt']
./B/P    []    ['p.txt']
./B/Q    []    ['q1.txt', 'q2.txt']
```python
>>> from os import walk
>>> tree = walk('.

Each tuple contains a directory, its subdirectories

>>> for dir, subdirs, files in tree:
...     print "%s   %s   %s" %(dir, subdirs, files)
...
. [ 'C', 'A', 'B' ] []
./C [] ['c.txt']
./A [] ['a1.txt', 'a2.txt']
./B ['P', 'Q'] ['b.txt']
./B/P [] ['p.txt']
./B/Q [] ['q1.txt' 'q2.txt']
```
>>> from os import walk
>>> tree = walk('.

Each tuple contains a directory, its subdirectories, and its files

>>> for dir,subdirs,files in tree:
...    print "%s   %s   %s" %(dir,subdirs,files)
...
...    ['C', 'A', 'B']   []
...    ./C   []   ['c.txt']
...    ./A   []   ['a1.txt', 'a2.txt']
...    ./B   ['P', 'Q']   ['b.txt']
...    ./B/P   []   ['p.txt']
...    ./B/Q   []   ['q1.txt', 'q2.txt']
```python
>>> from os import walk
>>> tree = walk('..

>>> for dir, subdirs, files in tree:
...    print "%s %s %s" % (dir, subdirs, files)
...
.  ['C', 'A', 'B'] []
./C [] ['c.txt']
./A [] ['a1.txt', 'a2.txt']
./B ['P', 'Q'] ['b.txt']
./B/P [] ['p.txt']
./B/Q [] ['q1.txt', 'q2.txt']
```

walk’s input is used as a prefix for each directory name
>>> tree = walk(getcwd())
```python
>>> tree = walk(getcwd())
>>> for dir, subdirs, files in tree:
...     print "%s   %s   %s" %(dir, subdirs, files)
...
/user/vlad   ['C', 'A', 'B']   []
/user/vlad/C   []   ['c.txt']
/user/vlad/A   []   ['a1.txt', 'a2.txt']
/user/vlad/B   ['P', 'Q']   ['b.txt']
/user/vlad/B/P   []   ['p.txt']
/user/vlad/B/Q   []   ['q1.txt', 'q2.txt']
```
>>> tree = walk(getcwd())

>>> for dir, subdirs, files in tree:
...     print "%s %s %s" %(dir, subdirs, files)
...
/user/vlad    ['C', 'A', 'B']    []
/user/vlad/C   []               ['c.txt']
/user/vlad/A   []               ['a1.txt', 'a2.txt']
/user/vlad/B   ['P', 'Q']       ['b.txt']
/user/vlad/B/P []               ['p.txt']
/user/vlad/B/Q []               ['q1.txt', 'q2.txt']
>>> tree = walk(getcwd(), topdown=False)
>>> tree = walk(getcwd(), topdown=False)

>>> for dir, subdirs, files in tree:
    ...    print "%s    %s    %s" % (dir, subdirs, files)
    ...

/user/vlad/C   []   ['c.txt']
/user/vlad/A   []   ['a1.txt', 'a2.txt']
/user/vlad/B/P  []   ['p.txt']
/user/vlad/B/Q  []   ['q1.txt', 'q2.txt']
/user/vlad/B    ['P', 'Q']   ['b.txt']
/user/vlad   ['C', 'A', 'B']   []
```python
>>> tree = walk(getcwd(), topdown=False)
>>> for dir, subdirs, files in tree:
...     print "%s   %s   %s" %(dir, subdirs, files)
...
/user/vlad/C   []   ['c.txt']
/user/vlad/A   []   ['a1.txt', 'a2.txt']
/user/vlad/B/P  []   ['p.txt']
/user/vlad/B/Q  []   ['q1.txt', 'q2.txt']
/user/vlad/B   ['P', 'Q']   ['b.txt']
/user/vlad   ['C', 'A', 'B']   []
```

P and Q are before B
```python
>>> tree = walk(getcwd(), topdown=False)

>>> for dir,subdirs,files in tree:
...     print "%s   %s   %s" %(dir,subdirs,files)
...
/user/vlad/C   []   ['c.txt']
/user/vlad/A   []   ['a1.txt', 'a2.txt']
/user/vlad/B/P  []   ['p.txt']
/user/vlad/B/Q  []   ['q1.txt', 'q2.txt']
/user/vlad/B   ['P', 'Q']   ['b.txt']
/user/vlad    ['C', 'A', 'B']   []
```

A, B and C are before the original directory.
<table>
<thead>
<tr>
<th>os</th>
<th>Miscellaneous operating system interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>walk</td>
<td>Recursively explore directory contents</td>
</tr>
</tbody>
</table>