

# Python

## Browsing Directories Using walk



Copyright © Software Carpentry and The University of Edinburgh 2010-2011 This work is licensed under the Creative Commons Attribution License See http://software-carpentry.org/license.html for more information.

```
>>> from os import walk
>>> tree = walk('.')
```





a1.txt a2.txt



В









c.txt



P





q1.txt q2.txt p.txt





a1.txt a2.txt





В





c.txt



walk('.')



P

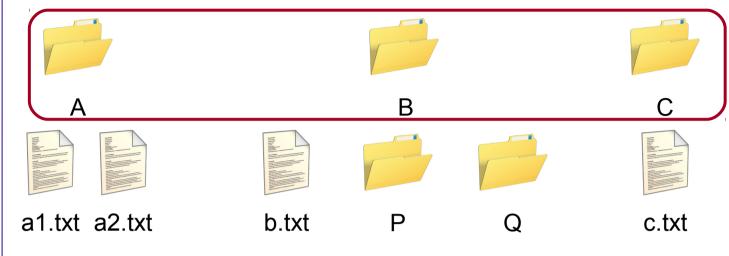


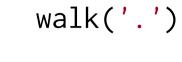


q1.txt q2.txt p.txt















p.txt q1.txt q2.txt

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







a1.txt a2.txt



P



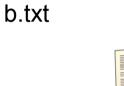




c.txt



walk('.')







p.txt

q1.txt q2.txt

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











a1.txt a2.txt



В



C



c.txt



b.txt



Р



Walter State of the Control of the C

p.txt

q1.txt q2.txt

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





a1.txt a2.txt





b.txt



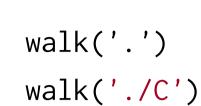
В



P



c.txt









q1.txt q2.txt p.txt

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





a1.txt a2.txt



В



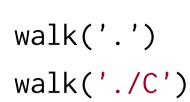
b.txt







c.txt





P



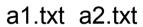


p.txt q1.txt q2.txt

./C











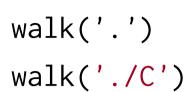
b.txt



В



c.txt





P





q1.txt q2.txt p.txt





a1.txt a2.txt





b.txt

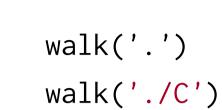


Р





c.txt











a1.txt a2.txt



В



b.txt



c.txt





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



P







Α



a1.txt a2.txt



В



C



c.txt



b.txt



C



P





p.txt q1.txt q2.txt





a1.txt a2.txt



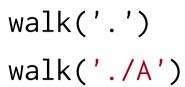
P







c.txt





b.txt

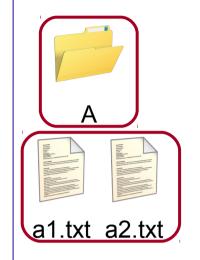






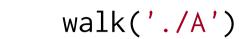


p.txt q1.txt q2.txt

















b.txt

Р

Q

c.txt







p.txt q1.txt q2.txt

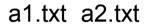
```
. ['C', 'A', 'B']
./C []
./A []
```

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

walk('./A')















b.txt





c.txt



Р











a1.txt a2.txt







c.txt



b.txt







Р





walk('./B')





a1.txt a2.txt



b.txt





Р













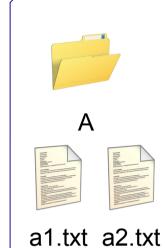
c.txt

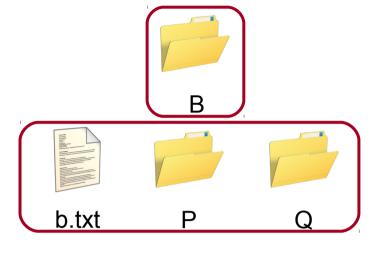


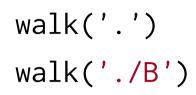




```
['c.txt']
['a1.txt', 'a2.txt']
```







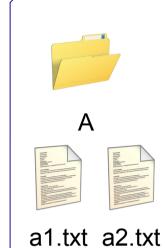


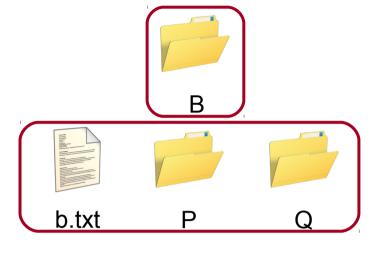


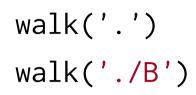


c.txt

p.txt q1.txt q2.txt













c.txt

p.txt q1.txt q2.txt



Α



a1.txt a2.txt





Р



C



c.txt

walk('.')







p.txt q1.txt q2.txt



Α



a1.txt a2.txt



В



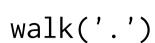
Р



C



c.txt









p.txt q1.txt q2.txt

```
[]
['c.txt']
['a1.txt', 'a2.txt']
['b.txt']
['p.txt']
```

walk('./B')





a1.txt a2.txt









c.txt



Р

b.txt





```
['C', 'A', 'B']
./C
./A []
./B ['P', 'Q']
./B/P []
```

```
['c.txt']
['a1.txt', 'a2.txt']
['b.txt']
['p.txt']
```



Α



a1.txt a2.txt



B



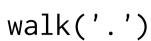
Р



C



c.txt



walk('./B')







p.txt q1.txt q2.txt

```
[]
['c.txt']
['a1.txt', 'a2.txt']
['b.txt']
['p.txt']
['q1.txt' 'q2.txt']
```

walk('./B')













Р









c.txt







p.txt q1.txt q2.txt

```
['C', 'A', 'B']
./C
./A []
./B ['P', 'Q']
./B/P []
./B/Q []
```

```
['c.txt']
['a1.txt', 'a2.txt']
['b.txt']
['p.txt']
['q1.txt' 'q2.txt']
```







a1.txt a2.txt



В



C



c.txt







p.txt q1.txt q2.txt

```
. ['C', 'A', 'B']
./C []
./A []
./B ['P', 'Q']
./B/P []
./B/Q []
```

```
LJ
['c.txt']
['a1.txt', 'a2.txt']
['b.txt']
['p.txt']
['q1.txt' 'q2.txt']
```







a1.txt a2.txt



В







b.txt





c.txt



Р





p.txt q1.txt q2.txt

```
['C', 'A', 'B']
./C
./A []
./B ['P', 'Q']
./B/P []
./B/Q []
```

```
['c.txt']
['a1.txt', 'a2.txt']
['b.txt']
['p.txt']
['q1.txt' 'q2.txt']
```

```
>>> from os import walk
>>> tree = walk('.')
```

```
>>> from os import 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)
...
```

```
>>> 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']
```

```
>>> 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']
```

```
>>> 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']
                               walk's input is used as a
./B/P [] ['p.txt']
./B/Q [] ['q1.txt' 'q2.txt']
                               prefix for each directory name
```

>>> tree = walk(getcwd())



```
>>> 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/B ['C', 'A', 'B'] []
```

```
>>> 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/B ['C', 'A', 'B'] []
```

A, B and C are before the original directory



os Miscellaneous operating system interfaces

walk Recursively explore directory contents



#### created by

### Mike Jackson and Greg Wilson

May 2011



Copyright © Software Carpentry and The University of Edinburgh 2010-2011 This work is licensed under the Creative Commons Attribution License See http://software-carpentry.org/license.html for more information.