



Python

Browsing Directories Using **walk**

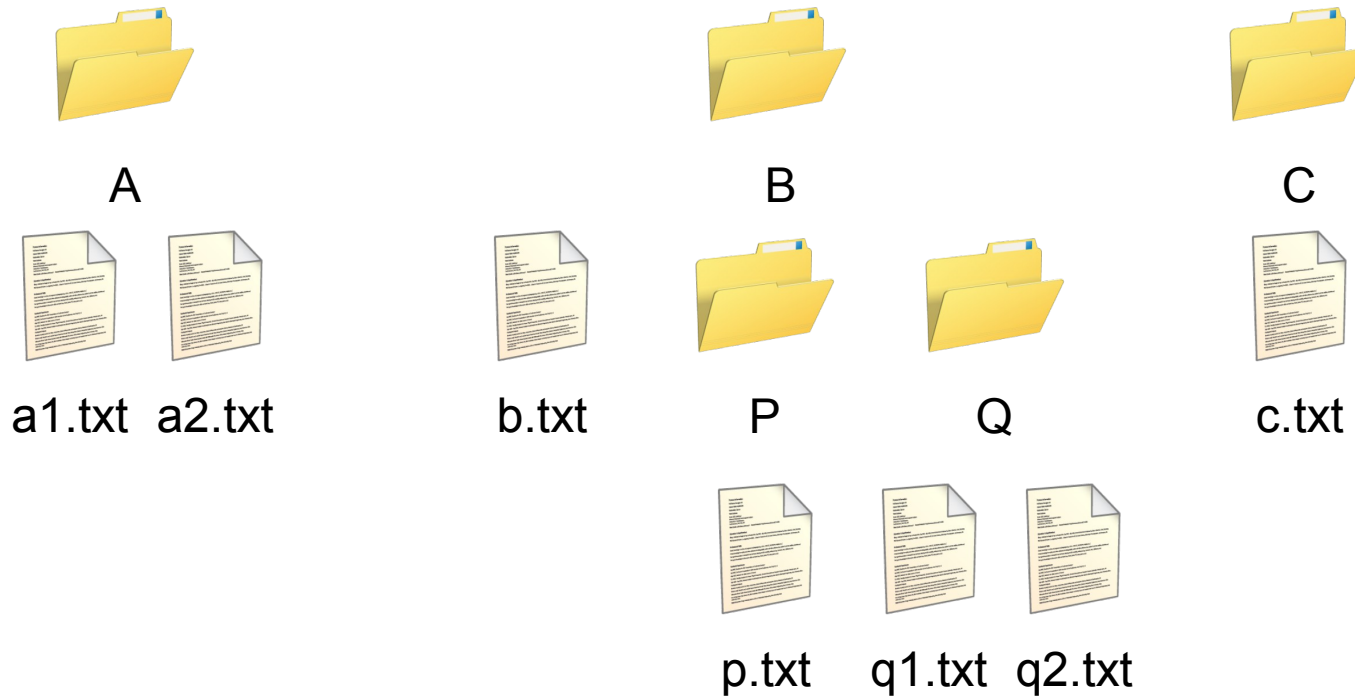


Copyright © Software Carpentry and The University of Edinburgh 2010-2011

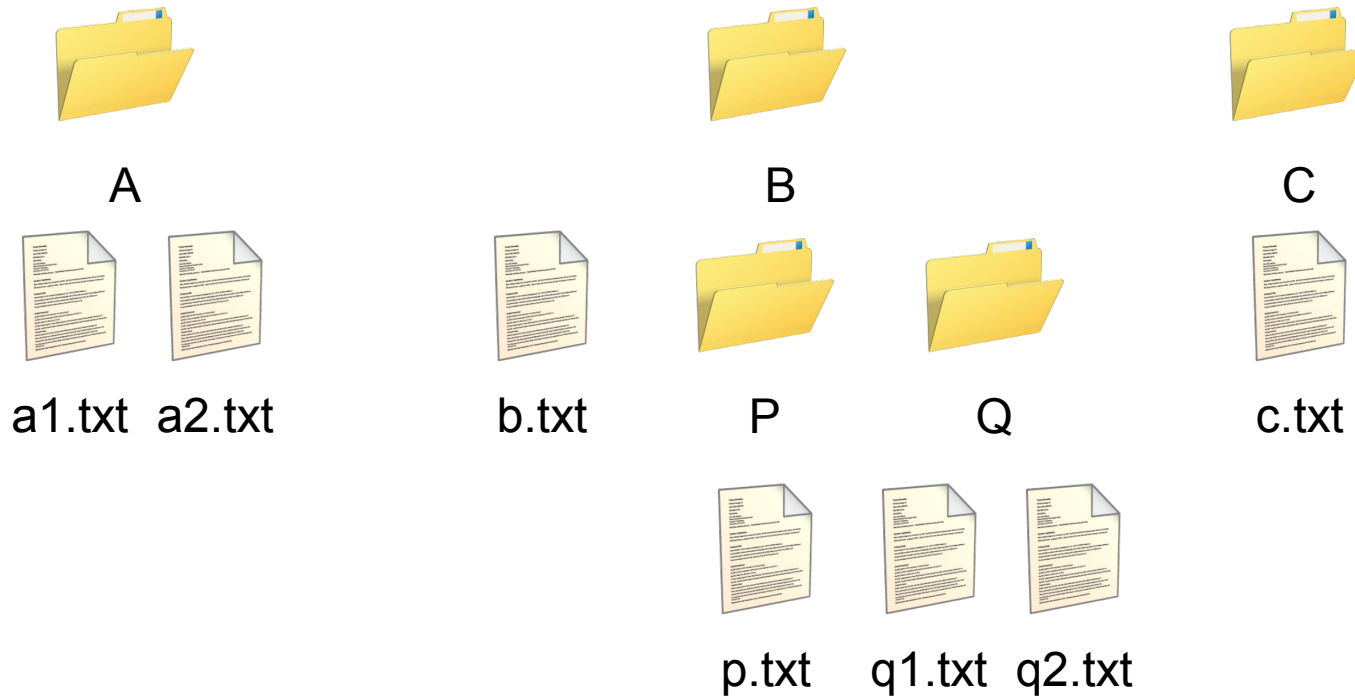
This work is licensed under the Creative Commons Attribution License

See <http://software-carpentry.org/license.html> for more information.

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



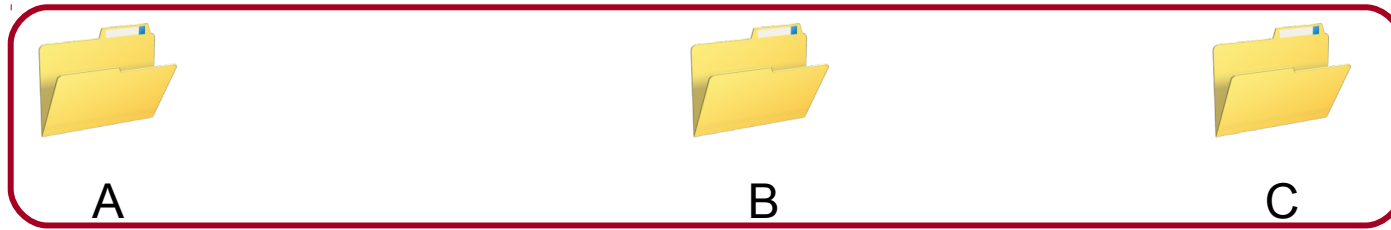
`walk('.',')`



`walk('.',')`

.





`walk('.', '')`



a1.txt a2.txt



b.txt



P



Q



c.txt



p.txt



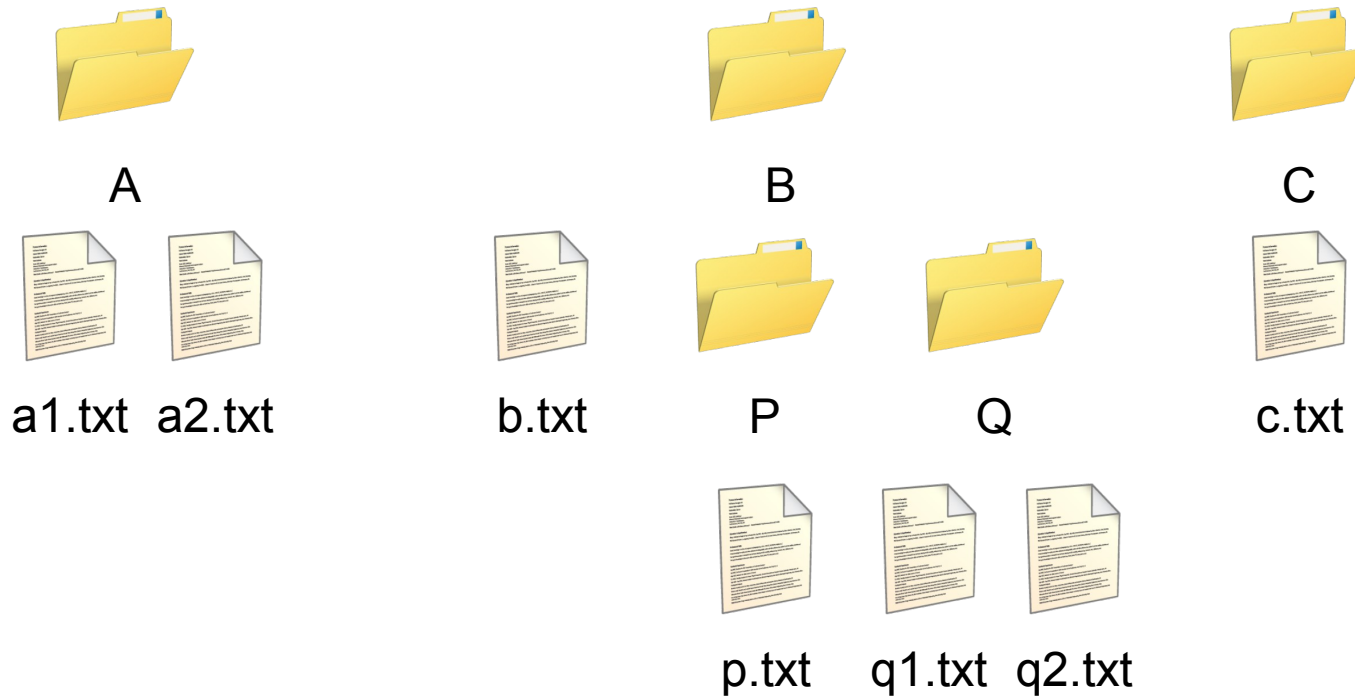
q1.txt



q2.txt

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



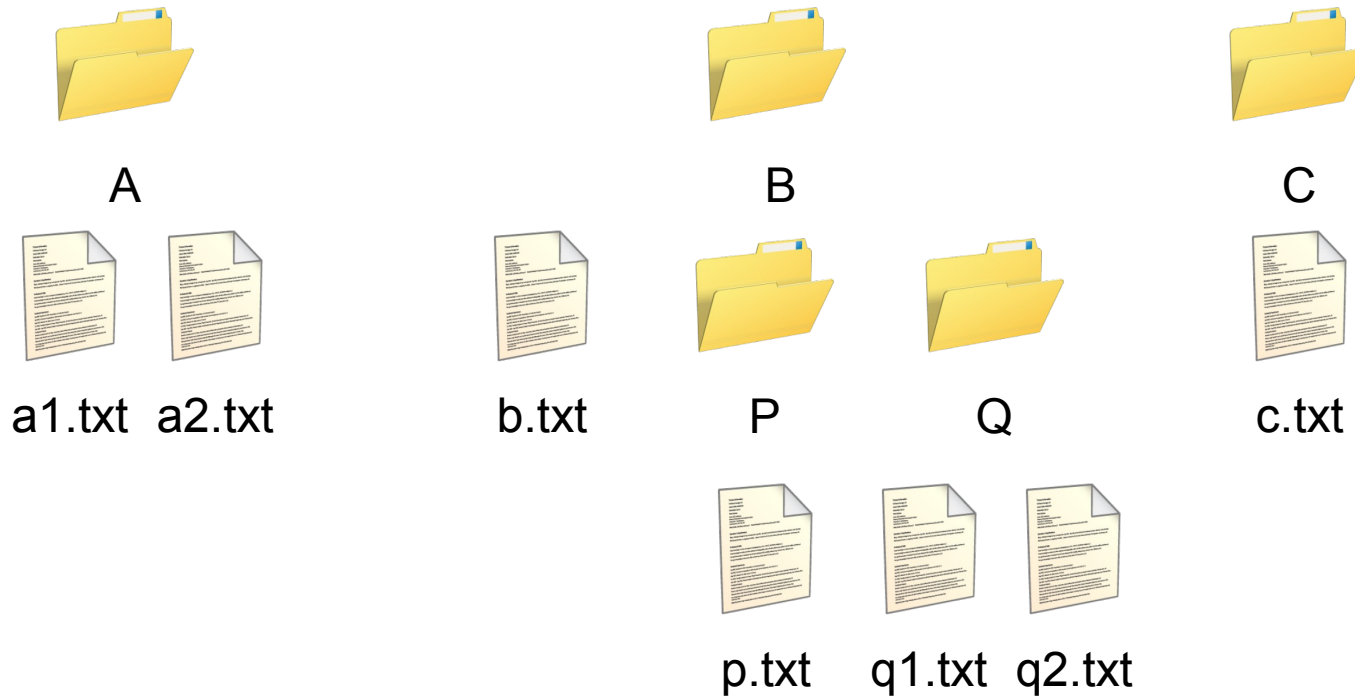


`walk('.', '')`

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

`[]`

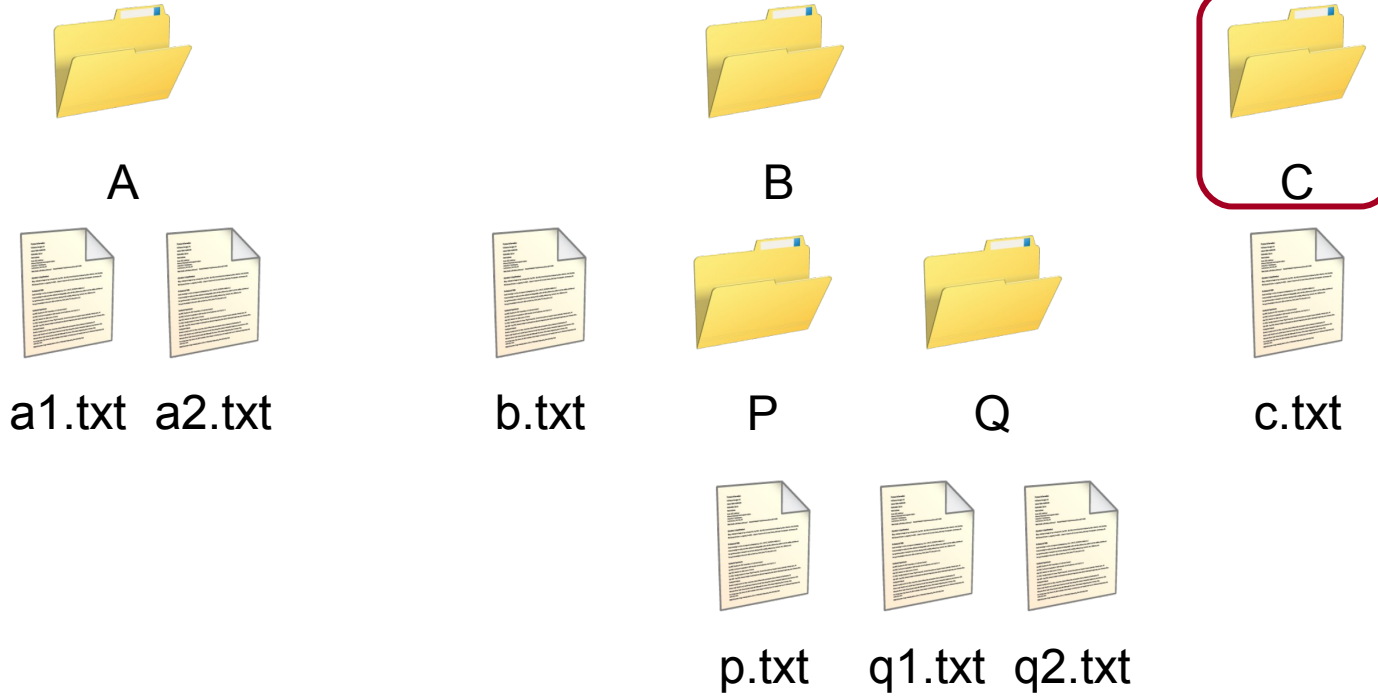




`walk('.', '')`

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

`[]`

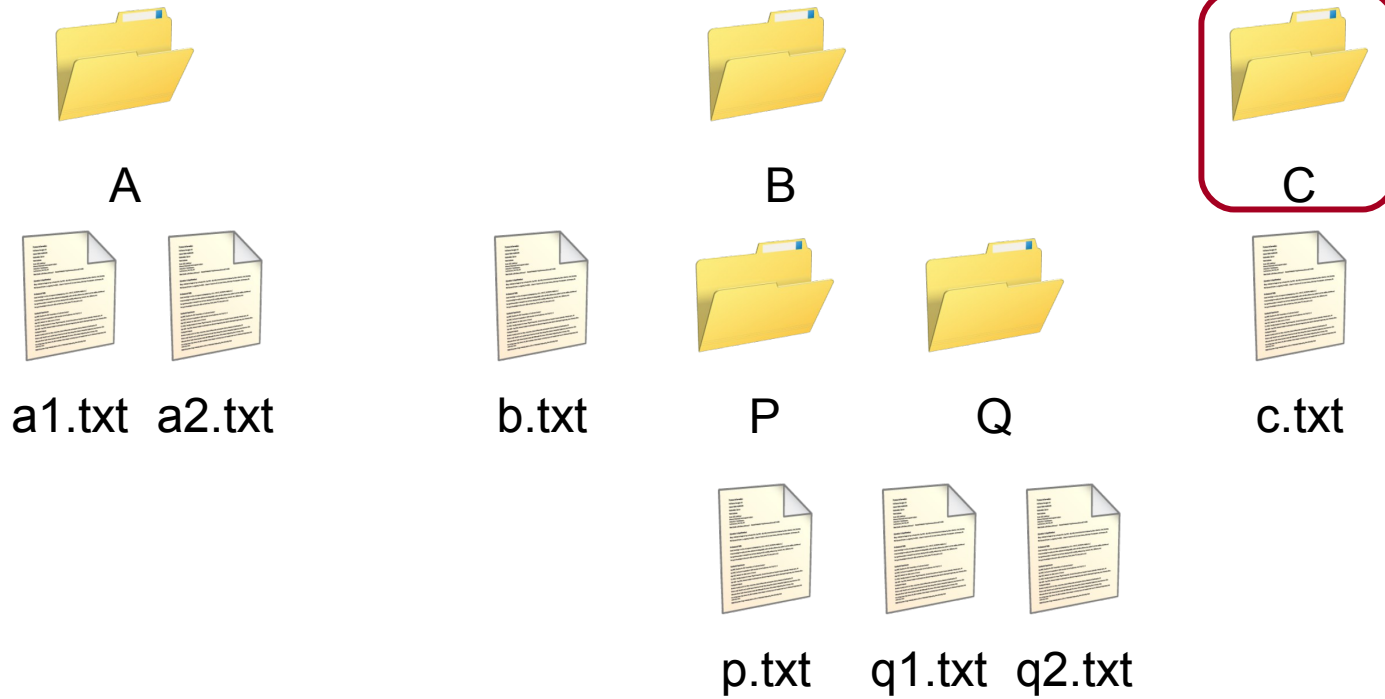


`walk('.')`

`walk('./C')`

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

`[]`

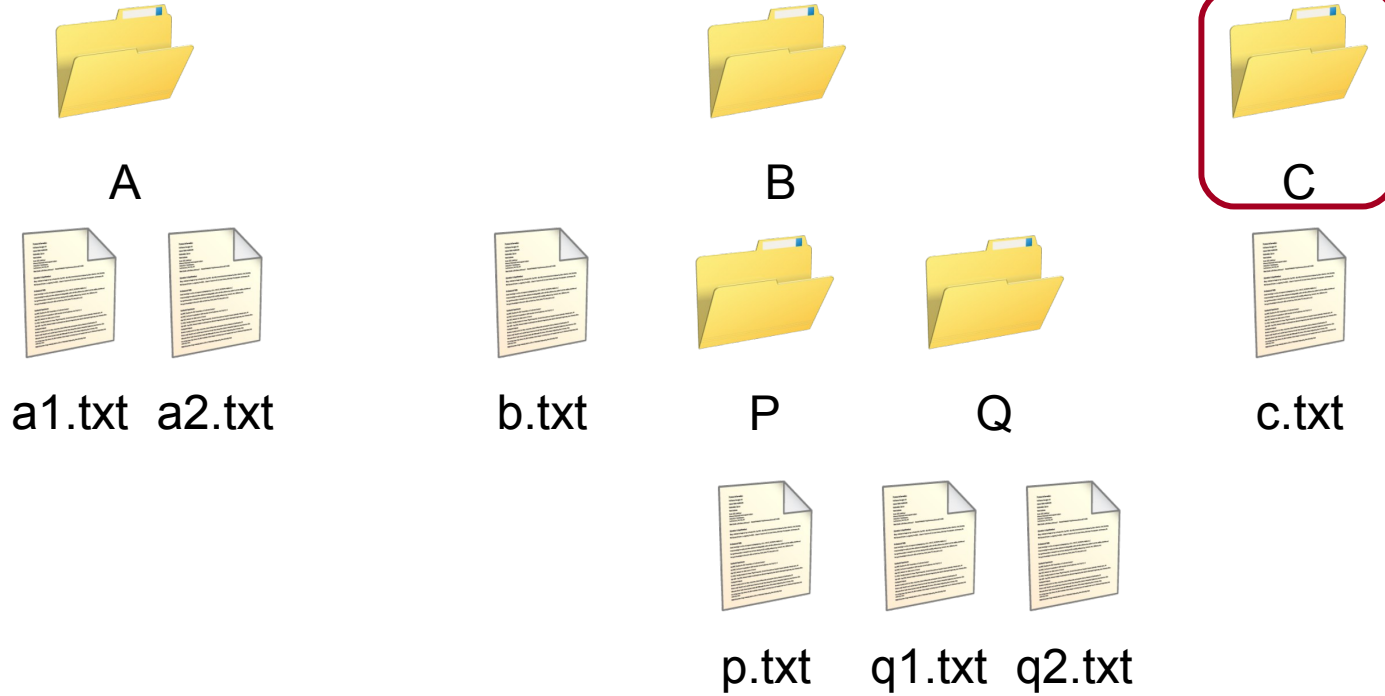


`walk('.')`

`walk('./C')`

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

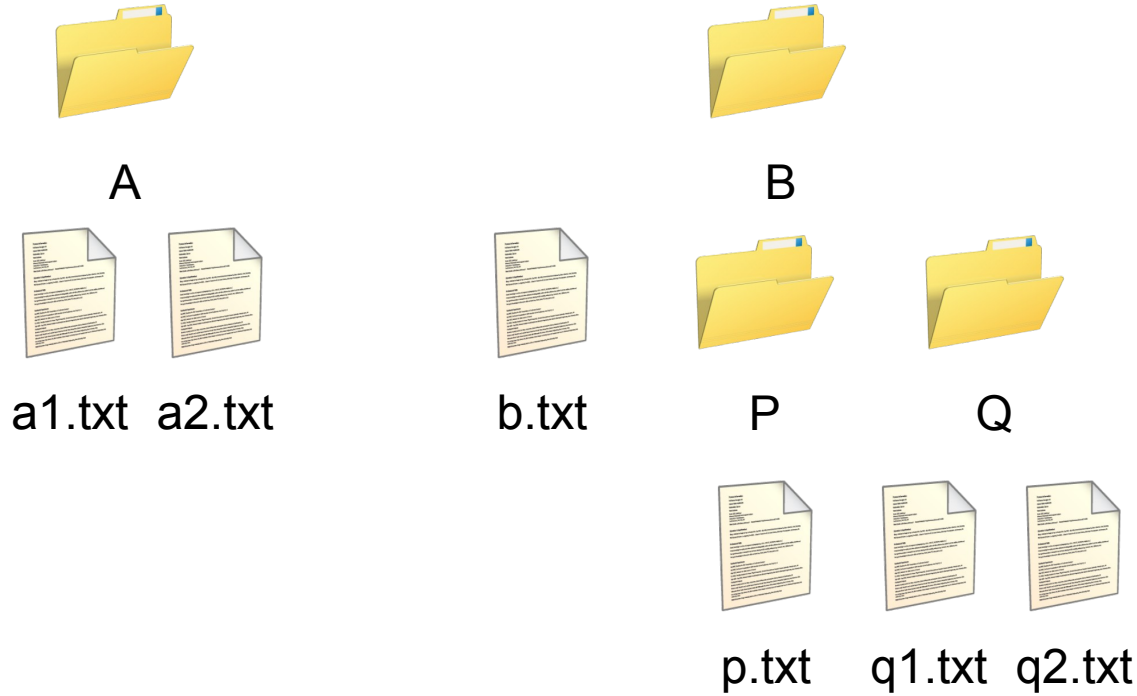
`[]`



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

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

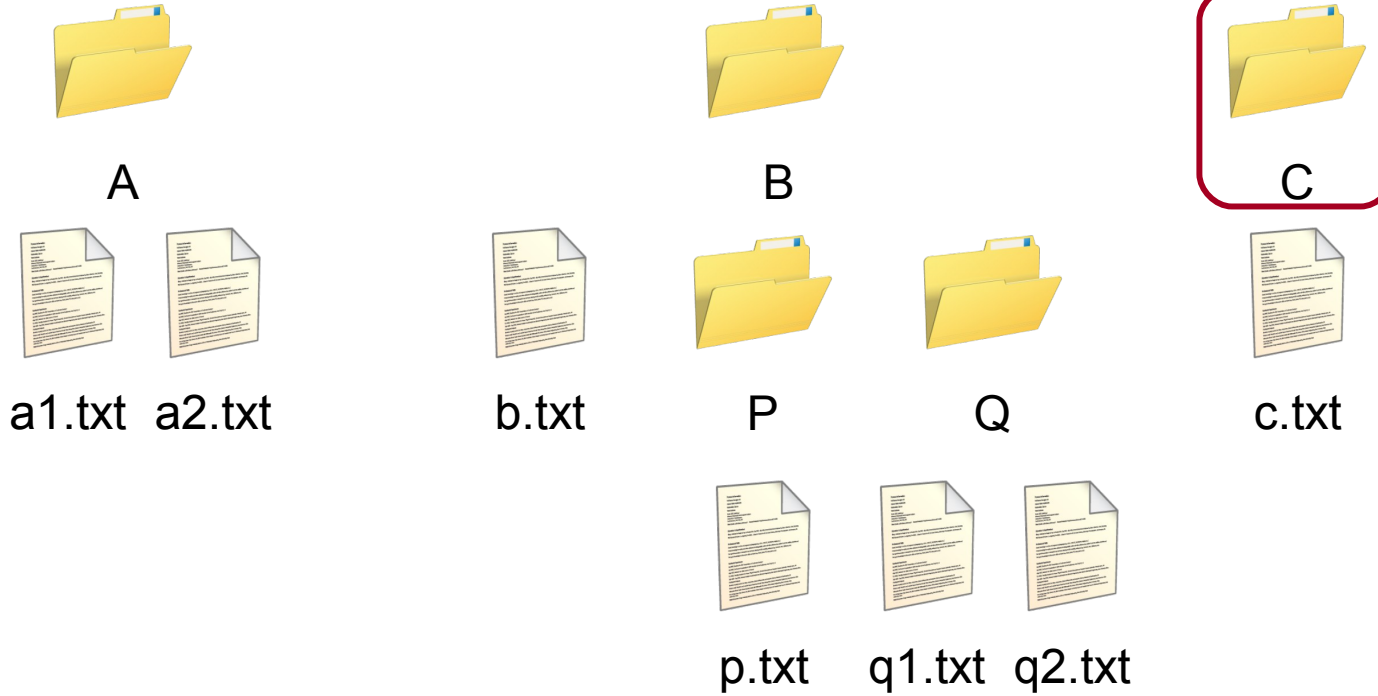
[]



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

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

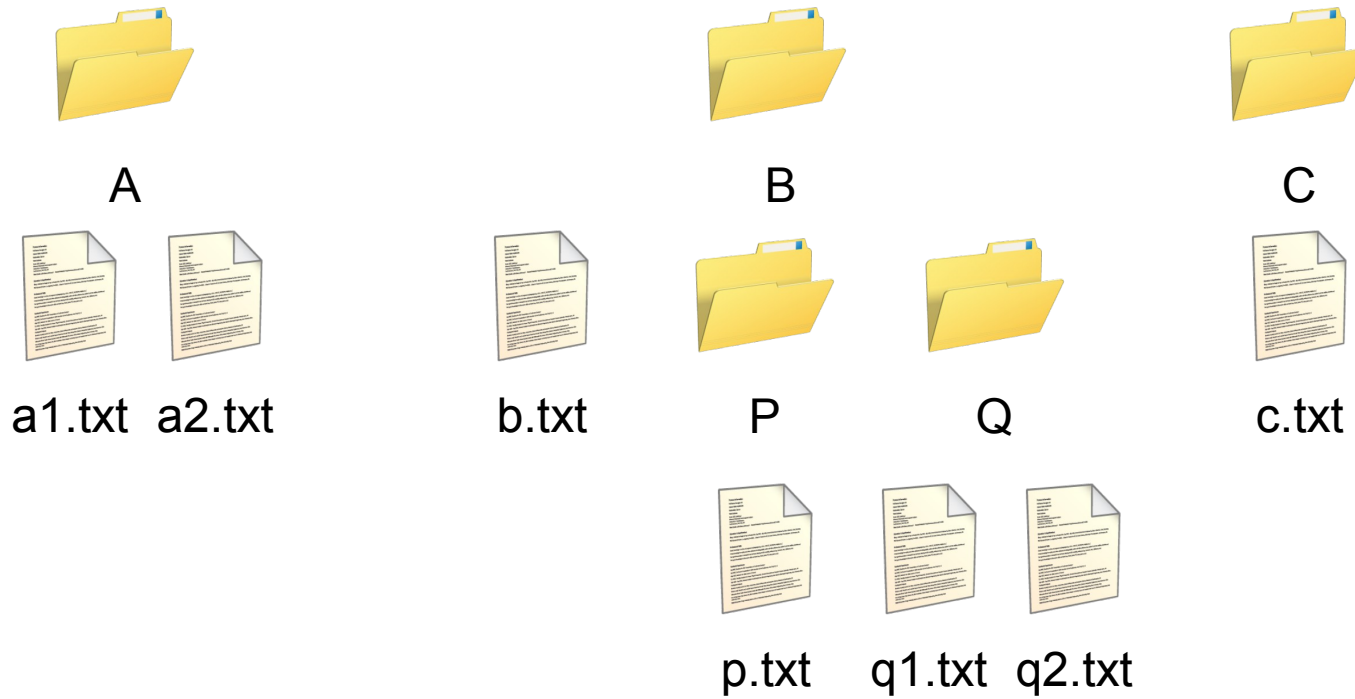
[]
['c.txt']



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

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

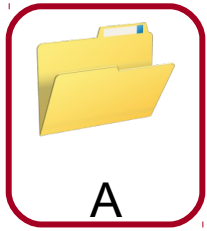
[]
['c.txt']



walk('.', '.')

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

[]
 ['c.txt']



A



a1.txt a2.txt



b.txt



B



P



Q



p.txt



q1.txt



q2.txt



C



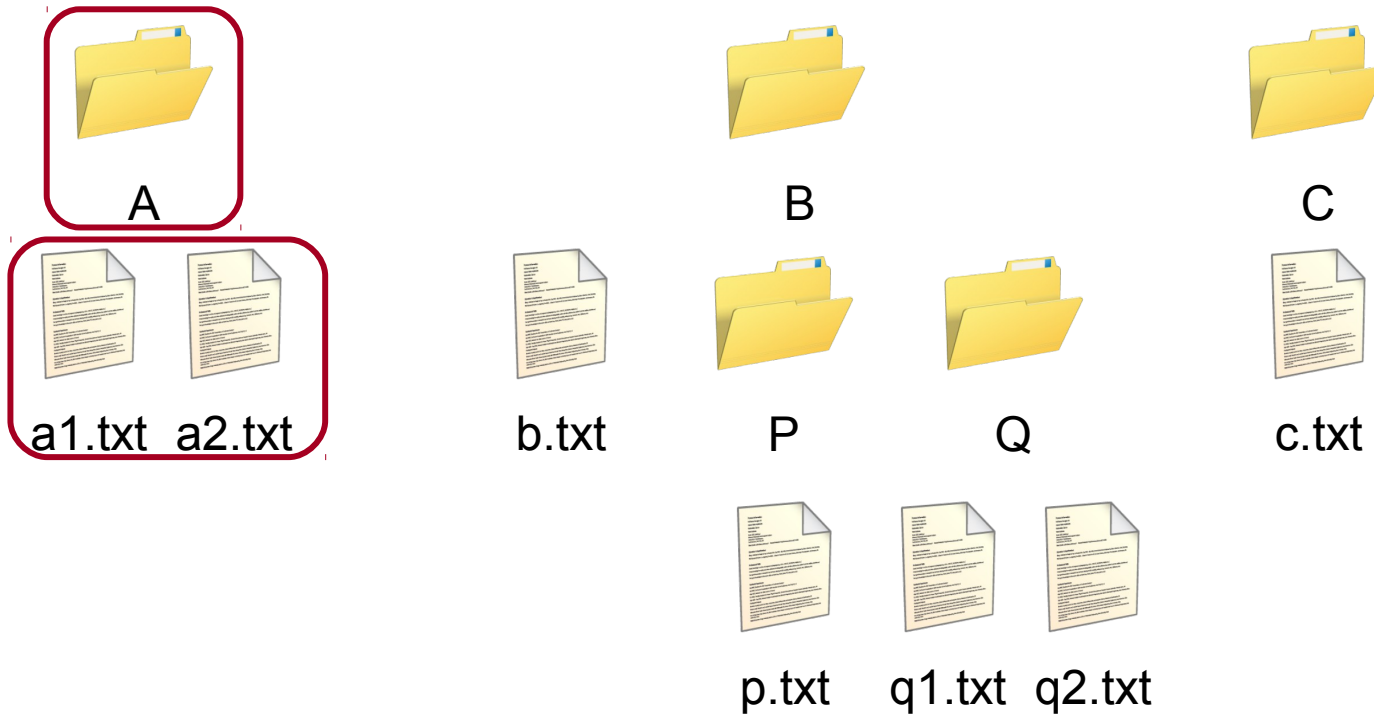
c.txt

`walk('.')`

`walk('./A')`

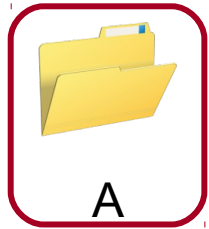
`.` ['C', 'A', 'B']
`./C` []

`[]`
`['c.txt']`



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

<code>.</code>	<code>['C', 'A', 'B']</code>	<code>[]</code>
<code>./C</code>	<code>[]</code>	<code>['c.txt']</code>
<code>./A</code>	<code>[]</code>	<code>['a1.txt', 'a2.txt']</code>



A



a1.txt a2.txt



b.txt



B



P



Q



p.txt



q1.txt



q2.txt



C



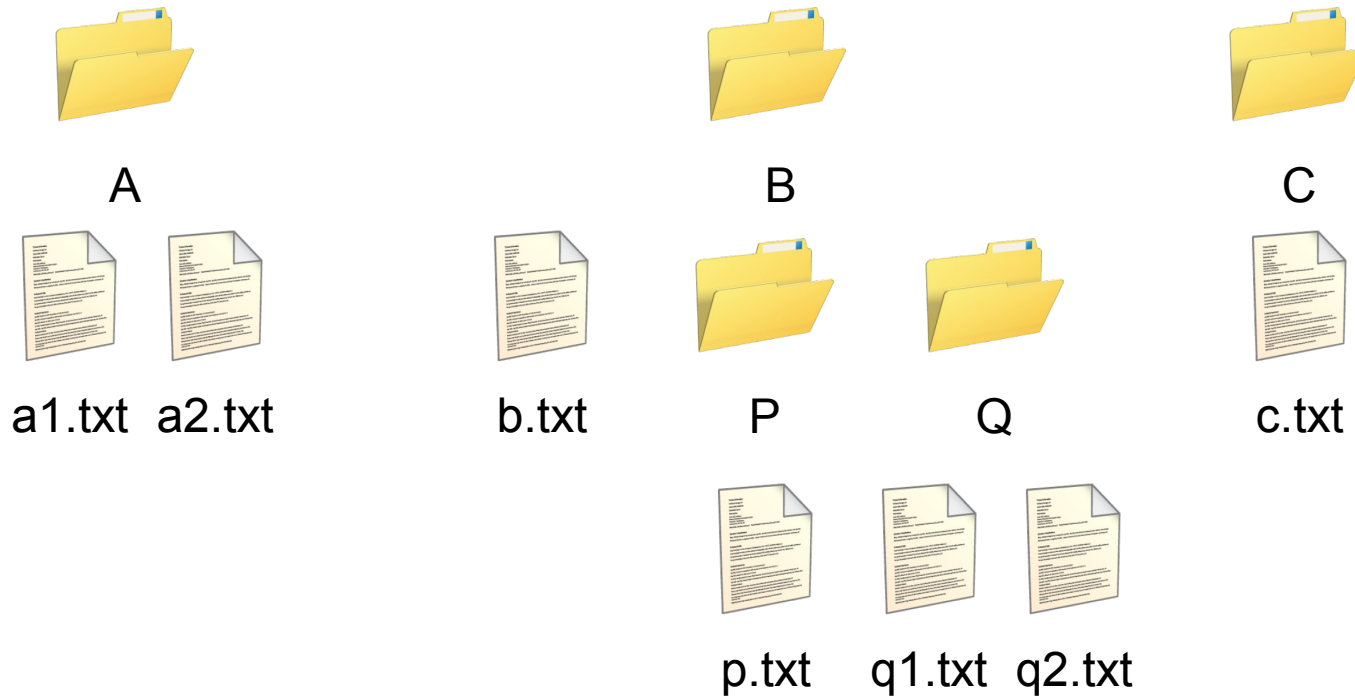
c.txt

`walk('.')`

`walk('./A')`

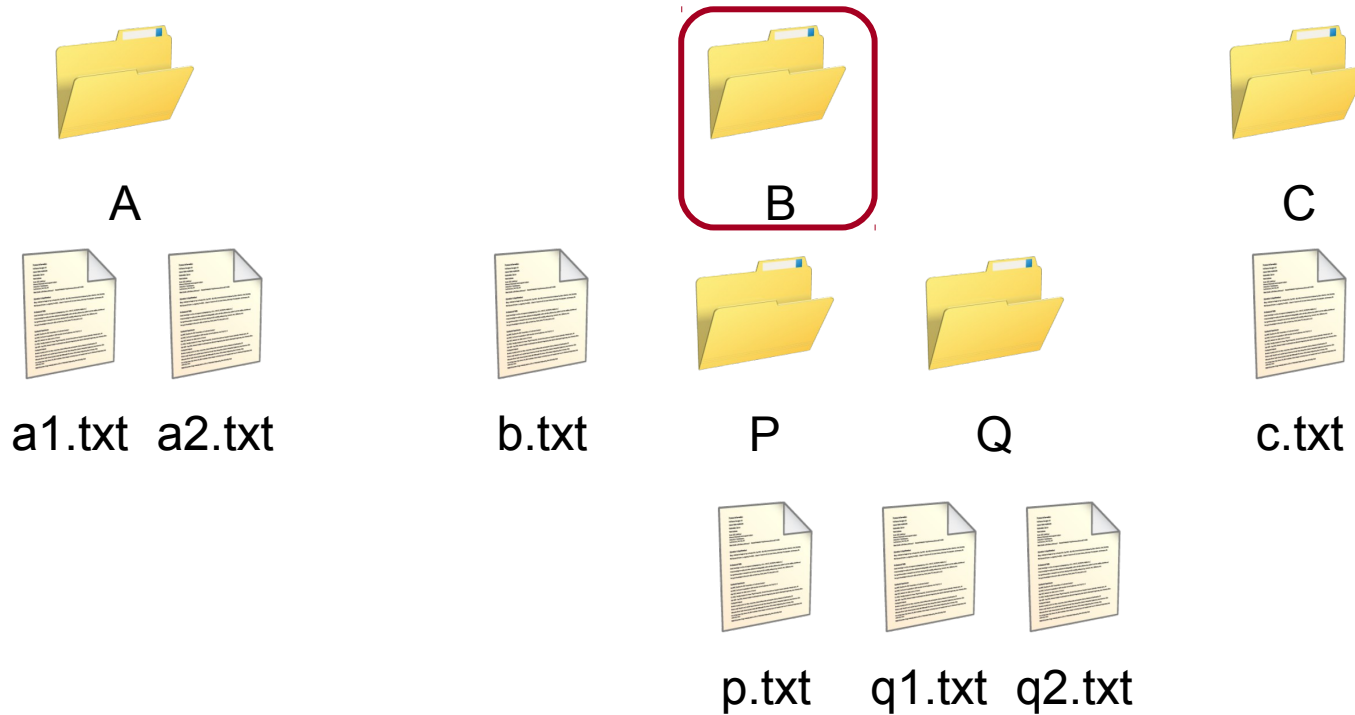
.	['C', 'A', 'B']
./C	[]
./A	[]

.	[]
./C	['c.txt']
./A	['a1.txt', 'a2.txt']



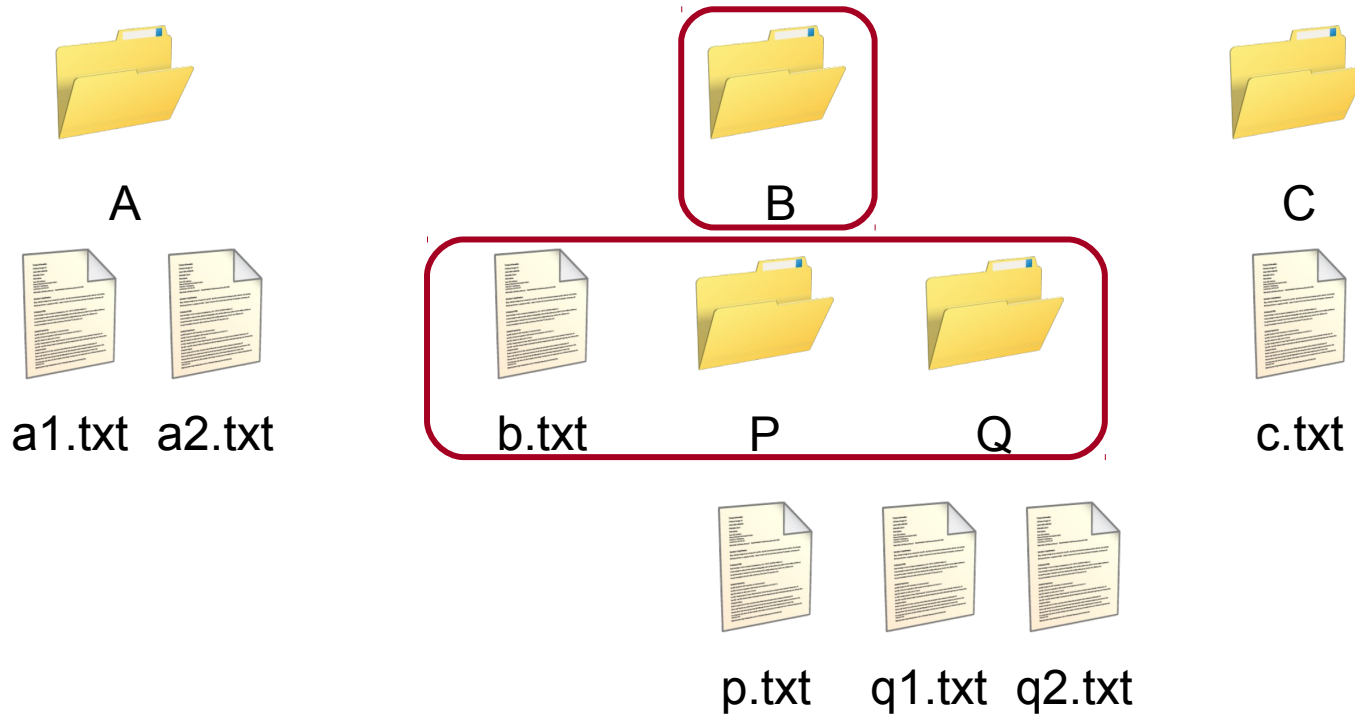
`walk('.', '.')`

<code>.</code>	<code>['C', 'A', 'B']</code>	<code>[]</code>
<code>./C</code>	<code>[]</code>	<code>['c.txt']</code>
<code>./A</code>	<code>[]</code>	<code>['a1.txt', 'a2.txt']</code>



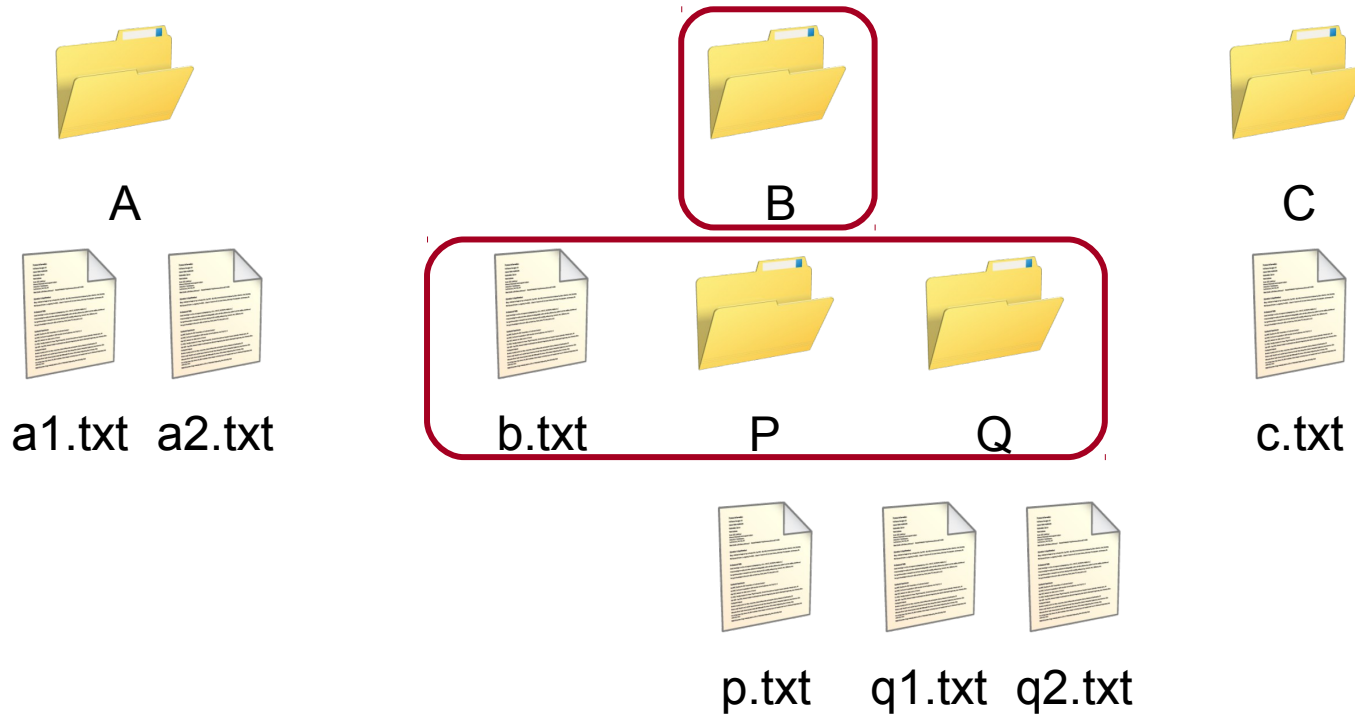
`walk('.')`
`walk('./B')`

<code>.</code>	<code>['C', 'A', 'B']</code>	<code>[]</code>
<code>./C</code>	<code>[]</code>	<code>['c.txt']</code>
<code>./A</code>	<code>[]</code>	<code>['a1.txt', 'a2.txt']</code>



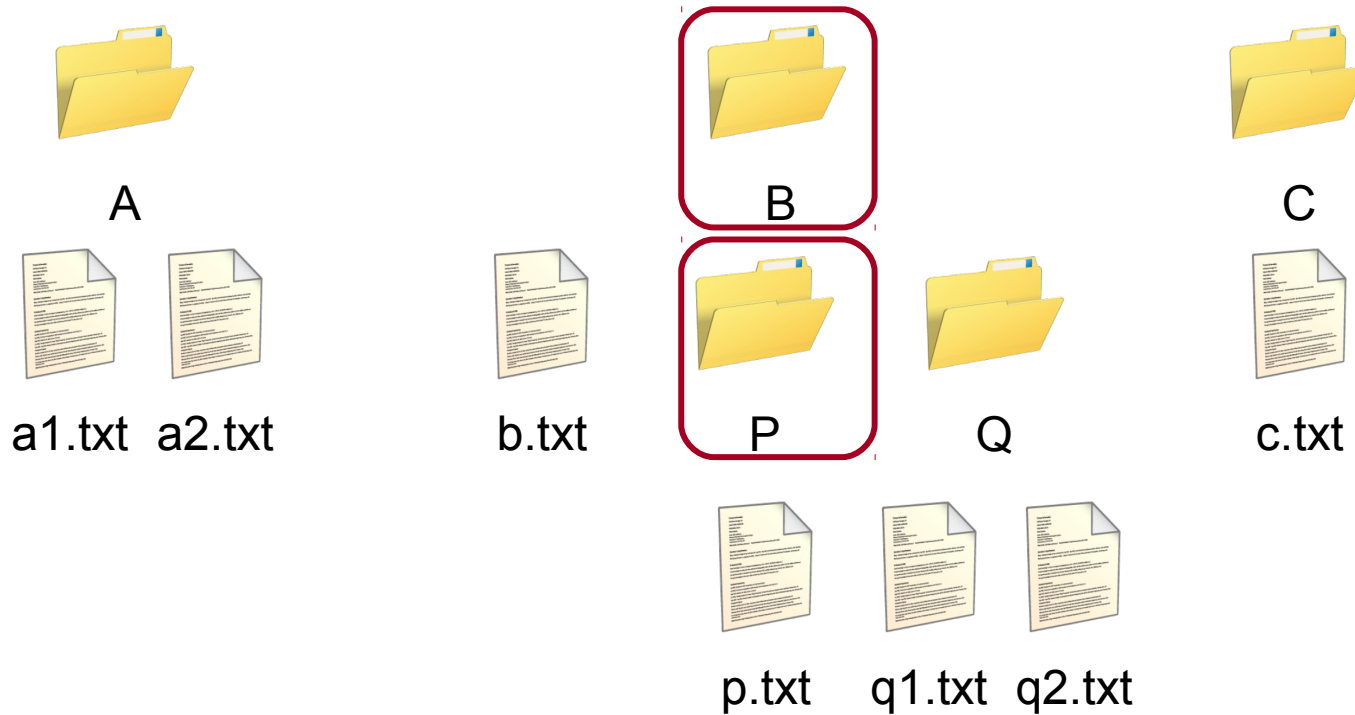
`walk('.')`
`walk('./B')`

<code>.</code>	<code>['C', 'A', 'B']</code>	<code>[]</code>
<code>./C</code>	<code>[]</code>	<code>['c.txt']</code>
<code>./A</code>	<code>[]</code>	<code>['a1.txt', 'a2.txt']</code>
<code>./B</code>	<code>['P', 'Q']</code>	<code>['b.txt']</code>



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

.	['C', 'A', 'B']	[]
./C	[]	['c.txt']
./A	[]	['a1.txt', 'a2.txt']
./B	['P', 'Q']	['b.txt']

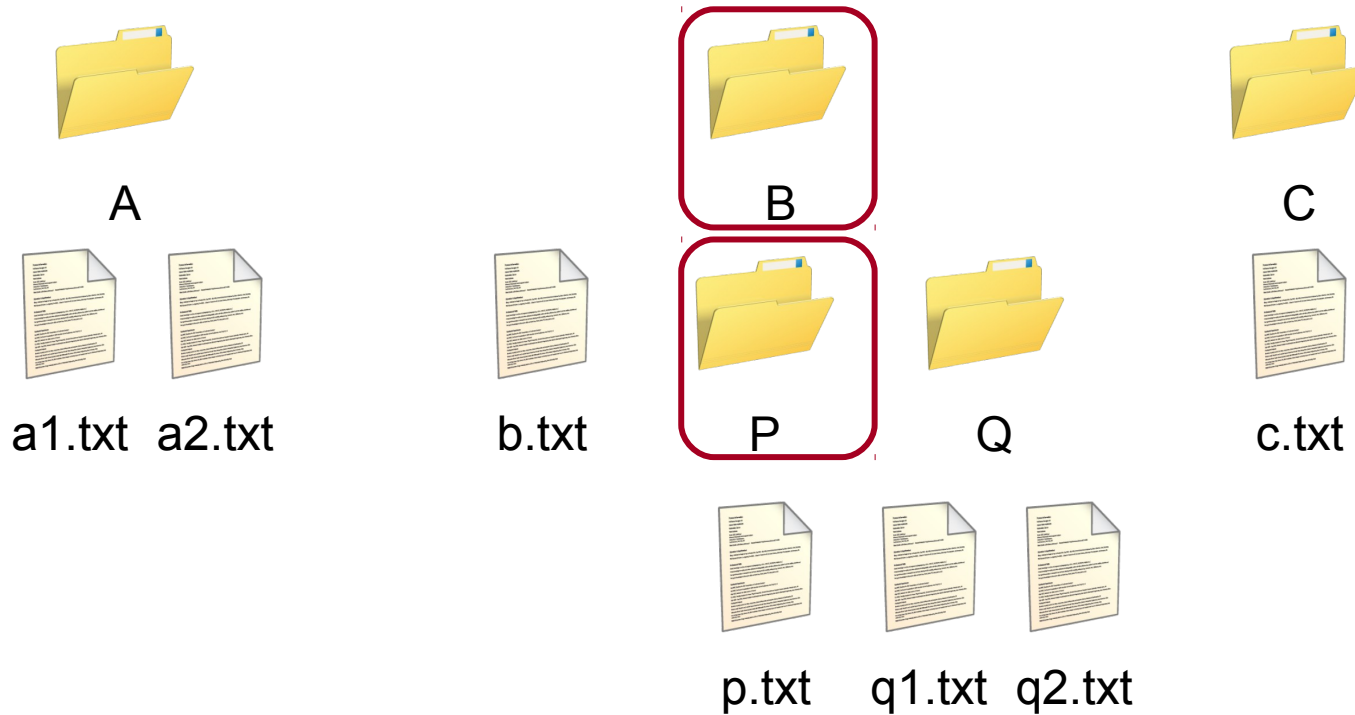


```

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

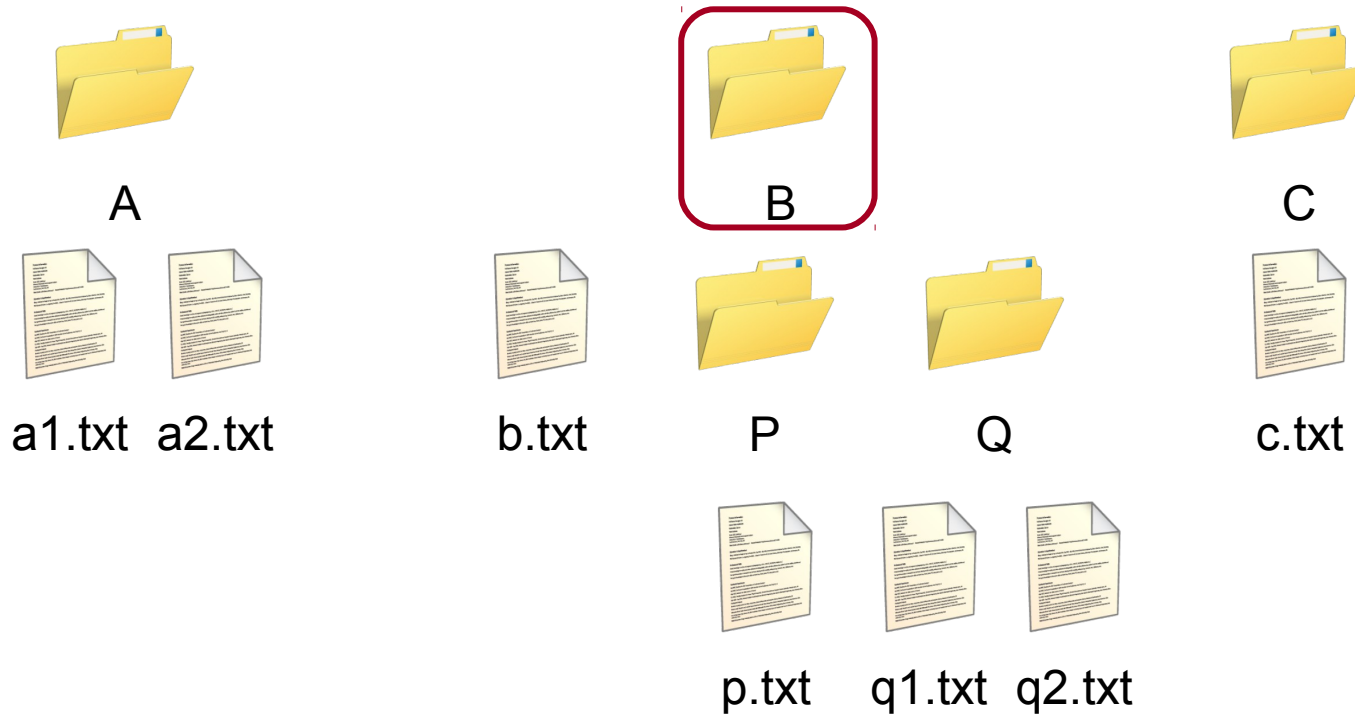
```

.	['C', 'A', 'B']	[]
./C	[]	['c.txt']
./A	[]	['a1.txt', 'a2.txt']
./B	['P', 'Q']	['b.txt']



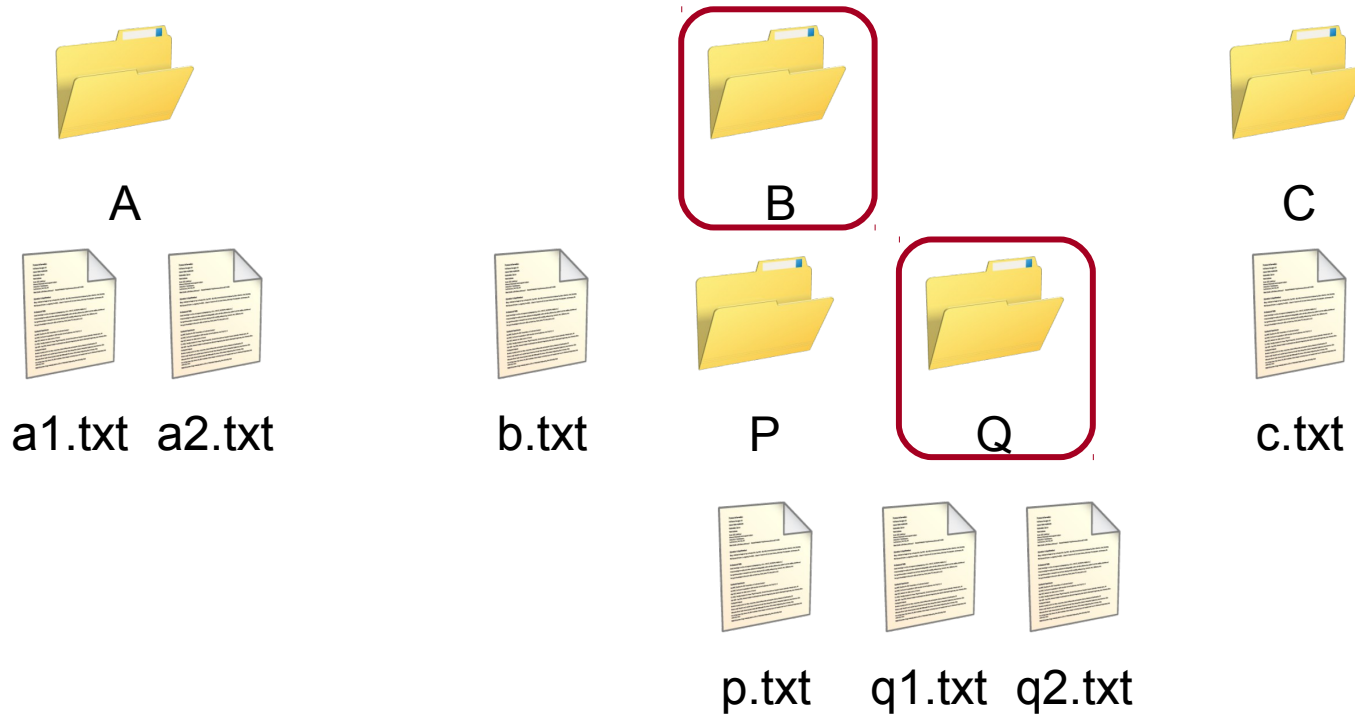
```
walk('.')
walk('./B')
walk('./B/P')
```

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



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

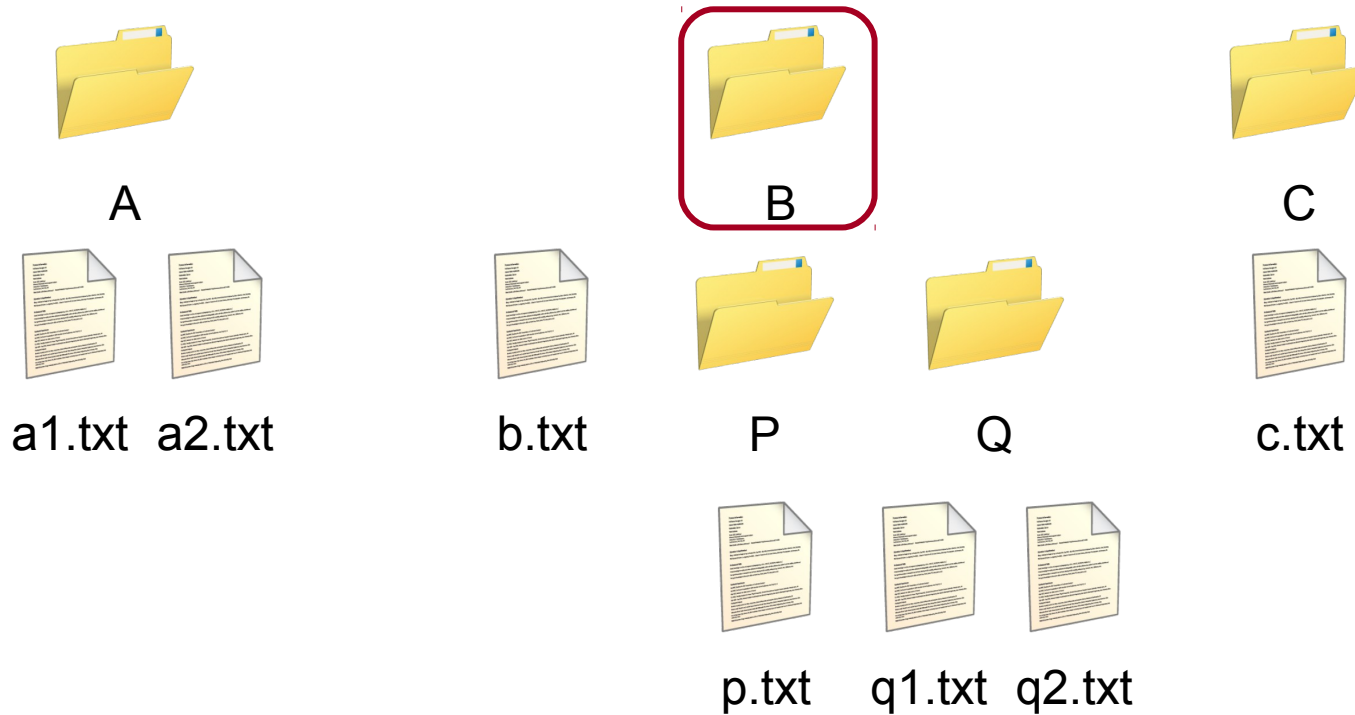
<code>.</code>	<code>['C', 'A', 'B']</code>	<code>[]</code>
<code>./C</code>	<code>[]</code>	<code>['c.txt']</code>
<code>./A</code>	<code>[]</code>	<code>['a1.txt', 'a2.txt']</code>
<code>./B</code>	<code>['P', 'Q']</code>	<code>['b.txt']</code>
<code>./B/P</code>	<code>[]</code>	<code>['p.txt']</code>



```

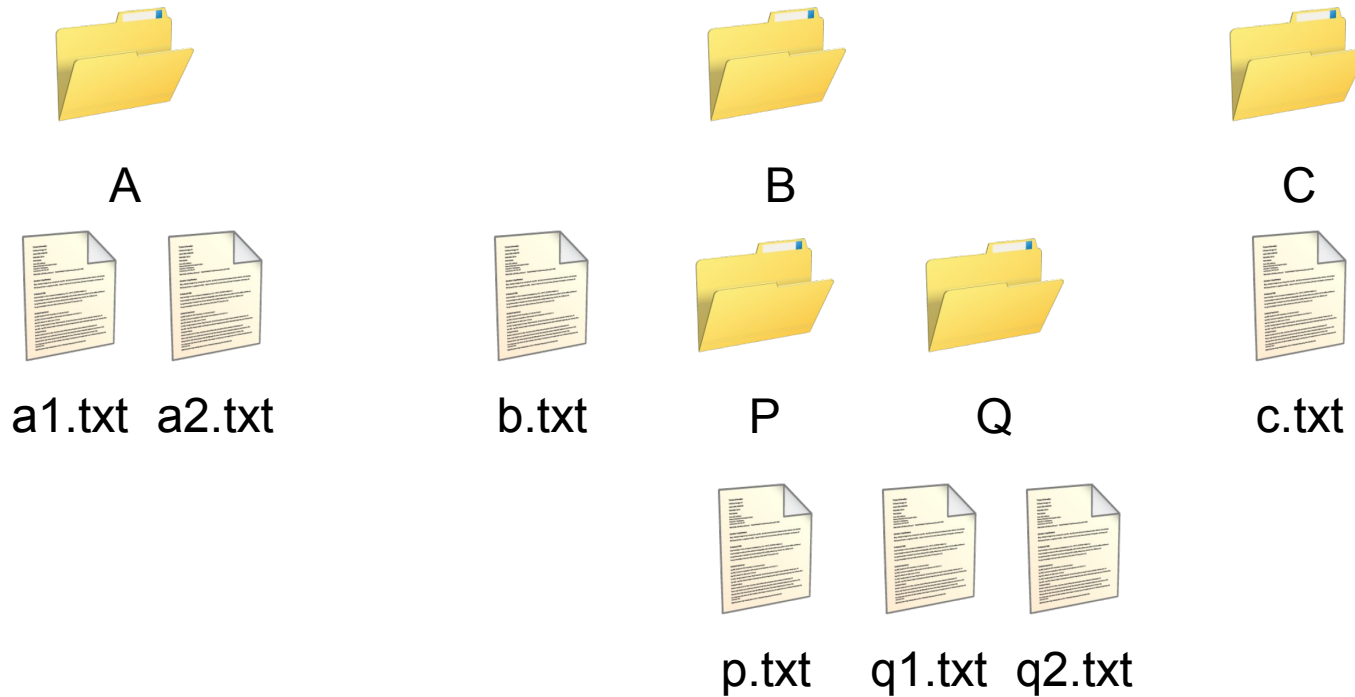
walk('.')
walk('./B')
walk('./B/Q')
  
```

.	['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']



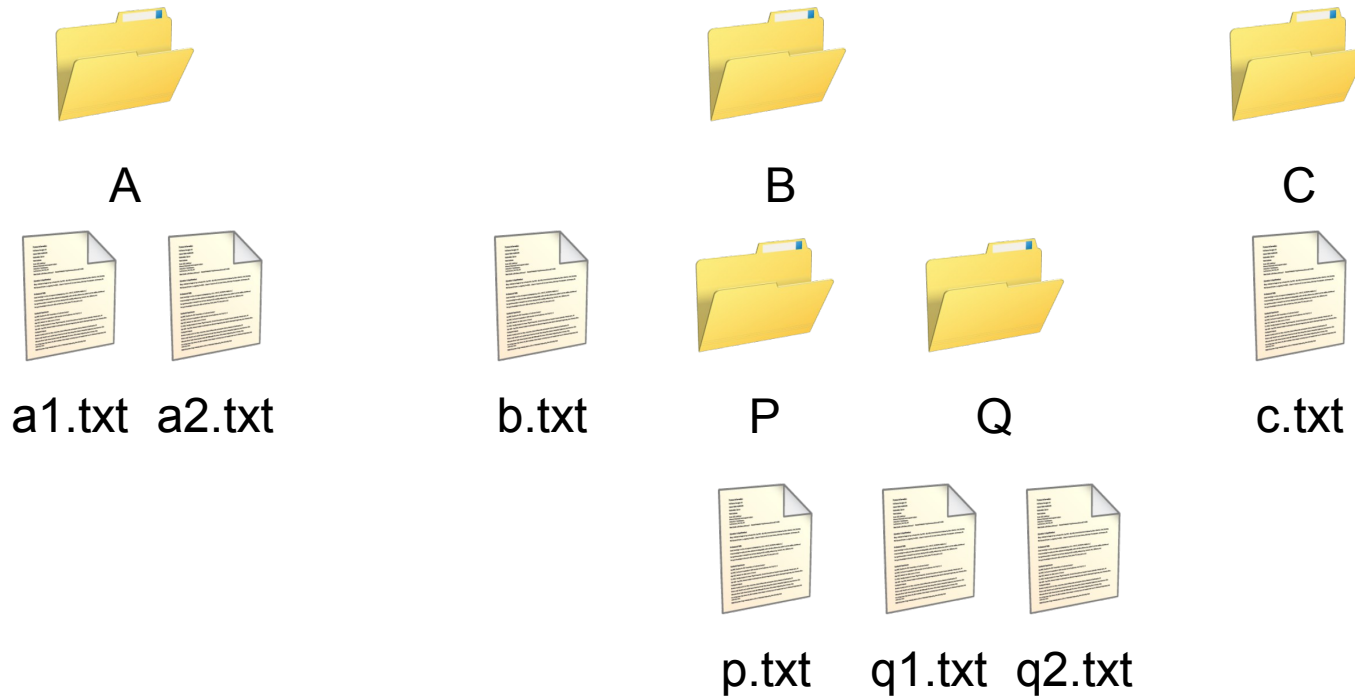
`walk('.')`
`walk('./B')`

<code>.</code>	<code>['C', 'A', 'B']</code>	<code>[]</code>
<code>./C</code>	<code>[]</code>	<code>['c.txt']</code>
<code>./A</code>	<code>[]</code>	<code>['a1.txt', 'a2.txt']</code>
<code>./B</code>	<code>['P', 'Q']</code>	<code>['b.txt']</code>
<code>./B/P</code>	<code>[]</code>	<code>['p.txt']</code>
<code>./B/Q</code>	<code>[]</code>	<code>['q1.txt', 'q2.txt']</code>



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

<code>.</code>	<code>['C', 'A', 'B']</code>	<code>[]</code>
<code>./C</code>	<code>[]</code>	<code>['c.txt']</code>
<code>./A</code>	<code>[]</code>	<code>['a1.txt', 'a2.txt']</code>
<code>./B</code>	<code>['P', 'Q']</code>	<code>['b.txt']</code>
<code>./B/P</code>	<code>[]</code>	<code>['p.txt']</code>
<code>./B/Q</code>	<code>[]</code>	<code>['q1.txt', 'q2.txt']</code>



`walk('.')`

<code>.</code>	<code>['C', 'A', 'B']</code>	<code>[]</code>
<code>./C</code>	<code>[]</code>	<code>['c.txt']</code>
<code>./A</code>	<code>[]</code>	<code>['a1.txt', 'a2.txt']</code>
<code>./B</code>	<code>['P', 'Q']</code>	<code>['b.txt']</code>
<code>./B/P</code>	<code>[]</code>	<code>['p.txt']</code>
<code>./B/Q</code>	<code>[]</code>	<code>['q1.txt', 'q2.txt']</code>

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

```
>>> 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)
...
```

```
>>> 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']
```

```
./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())
```

```
>>> 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']
```

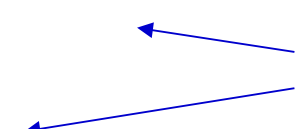
```
>>> tree = walk(getcwd(), topdown=False)
```

```
>>> 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']  []
```



```
>>> 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



```
>>> 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

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.