Operating on Directory Trees

```
In [46]: !ls brown
```

```
CONTENTS  ca41               cc10  ce18  cf24  cg18  cg60  ch27  cj39  ck01  cl14  cn26
README    ca42               cc11  ce19  cf25  cg19  cg61  ch28  cj40  ck02  cl15  cn27
ca01      ca43               cc12  ce20  cf26  cg20  cg62  ch29  cj41  ck03  cl16  cn28
ca02      ca44               cc13  ce21  cf27  cg21  cg63  ch30  cj42  ck04  cl17  cn29
ca03      categories.pickle  cc14  ce22  cf28  cg22  cg64  cj01  cj43  ck05  cl18  cp01
ca04      cats.txt           cc15  ce23  cf29  cg23  cg65  cj02  cj44  ck06  cl19  cp02
ca05      cb01               cc16  ce24  cf30  cg24  cg66  cj03  cj45  ck07  cl20  cp03
ca06      cb02               cc17  ce25  cf31  cg25  cg67  cj04  cj46  ck08  cl21  cp04
ca07      cb03               cd01  ce26  cf32  cg26  cg68  cj05  cj47  ck09  cl22  cp05
ca08      cb04               cd02  ce27  cf33  cg27  cg69  cj06  cj48  ck10  cl23  cp06
ca09      cb05               cd03  ce28  cf34  cg28  cg70  cj07  cj49  ck11  cl24  cp07
ca10      cb06               cd04  ce29  cf35  cg29  cg71  cj08  cj50  ck12  cm01  cp08
ca11      cb07               cd05  ce30  cf36  cg30  cg72  cj09  cj51  ck13  cm02  cp09
ca12      cb08               cd06  ce31  cf37  cg31  cg73  cj10  cj52  ck14  cm03  cp10
ca13      cb09               cd07  ce32  cf38  cg32  cg74  cj11  cj53  ck15  cm04  cp11
ca14      cb10               cd08  ce33  cf39  cg33  cg75  cj12  cj54  ck16  cm05  cp12
ca15      cb11               cd09  ce34  cf40  cg34  ch01  cj13  cj55  ck17  cm06  cp13
ca16      cb12               cd10  ce35  cf41  cg35  ch02  cj14  cj56  ck18  cn01  cp14
ca17      cb13               cd11  ce36  cf42  cg36  ch03  cj15  cj57  ck19  cn02  cp15
ca18      cb14               cd12  cf01  cf43  cg37  ch04  cj16  cj58  ck20  cn03  cp16
ca19      cb15               cd13  cf02  cf44  cg38  ch05  cj17  cj59  ck21  cn04  cp17
ca20      cb16               cd14  cf03  cf45  cg39  ch06  cj18  cj60  ck22  cn05  cp18
ca21      cb17               cd15  cf04  cf46  cg40  ch07  cj19  cj61  ck23  cn06  cp19
ca22      cb18               cd16  cf05  cf47  cg41  ch08  cj20  cj62  ck24  cn07  cp20
ca23      cb19               cd17  cf06  cf48  cg42  ch09  cj21  cj63  ck25  cn08  cp21
ca24      cb20               ce01  cf07  cg01  cg43  ch10  cj22  cj64  ck26  cn09  cp22
ca25      cb21               ce02  cf08  cg02  cg44  ch11  cj23  cj65  ck27  cn10  cp23
ca26      cb22               ce03  cf09  cg03  cg45  ch12  cj24  cj66  ck28  cn11  cp24
ca27      cb23               ce04  cf10  cg04  cg46  ch13  cj25  cj67  ck29  cn12  cp25
ca28      cb24               ce05  cf11  cg05  cg47  ch14  cj26  cj68  cl01  cn13  cp26
ca29      cb25               ce06  cf12  cg06  cg48  ch15  cj27  cj69  cl02  cn14  cp27
ca30      cb26               ce07  cf13  cg07  cg49  ch16  cj28  cj70  cl03  cn15  cp28
ca31      cb27               ce08  cf14  cg08  cg50  ch17  cj29  cj71  cl04  cn16  cp29
ca32      cc01               ce09  cf15  cg09  cg51  ch18  cj30  cj72  cl05  cn17  cr01
ca33      cc02               ce10  cf16  cg10  cg52  ch19  cj31  cj73  cl06  cn18  cr02
ca34      cc03               ce11  cf17  cg11  cg53  ch20  cj32  cj74  cl07  cn19  cr03
ca35      cc04               ce12  cf18  cg12  cg54  ch21  cj33  cj75  cl08  cn20  cr04
ca36      cc05               ce13  cf19  cg13  cg55  ch22  cj34  cj76  cl09  cn21  cr05
ca37      cc06               ce14  cf20  cg14  cg56  ch23  cj35  cj77  cl10  cn22  cr06
ca38      cc07               ce15  cf21  cg15  cg57  ch24  cj36  cj78  cl11  cn23  cr07
ca39      cc08               ce16  cf22  cg16  cg58  ch25  cj37  cj79  cl12  cn24  cr08
ca40      cc09               ce17  cf23  cg17  cg59  ch26  cj38  cj80  cl13  cn25  cr09
```

Let's look at operating on directory trees, a fairly common operation when dealing with files.

It's common to want to search through a directory tree of files for matches. These days, `grep` has a built-in option for that, but let's see whether we can write that in some other (and more flexible) way.

```
In [15]: !grep -r nuclear brown/. | wc
```

```
    107    3132   28944
```

The first thing people tend to do is look at the `find` command and see its `-exec` option; they then write something like this command. Do not use this kind of command; `-exec` is rarely the right thing to use because it is quite inefficient, because it is limited in what you can do with it, and because the syntax and quoting can get tricky.

```
In [44]: !find brown/. -type f -exec grep nuclear '{}' \; | wc # DO NOT USE
```

```
    107    3099   27553
```

A better way of dealing with this is the `xargs` command. It takes a partial command as its arguments, reads a list of file names on its standard input, and then applies the command to all those file names. It can do this in parallel (and there are even distributed versions of it).

```
In [16]: !find brown/. | xargs grep nuclear | wc
```

```
    107    3132   28944
```

To deal properly with file names containing spaces, you need to use one of the following two commands (look at the manual pages to see why that works). The latter is probably better behaved, since most UNIX commands expect line-oriented inputs, not null terminated inputs.

```
In [42]: !find brown/. -print0 | xargs -0 grep nuclear | wc
```

```
    107    3132   28944
```

```
In [45]: !find brown/. | xargs -d '\n' grep nuclear | wc   # THIS REALLY SHOULD BE THE DEFAULT
```

```
    107    3132   28944
```

The `-l` option to `grep` tells it only to list the names of matching files. So, if we want to know the number of matching files (instead of the number of matching lines), we use this command:

```
In [36]: !find brown/. | xargs grep -l nuclear | sed 5q
```

```
brown/./cj72
brown/./cj74
brown/./cb21
brown/./ch21
brown/./cg03
```

```
In [37]: !find brown/. | xargs grep -l nuclear | wc
```

```
     35      35     455
```

Since the output of `find` is just a list of lines, we can apply filters to it as well, for example searching for specific file names, file name extensions, or other conditions. So, if we want to look for the term `nuclear` only in the `ch` files of the Brown corpus, we can use this command:

```
In [39]: !find brown/. | fgrep brown/./ch | xargs grep -l nuclear | wc
```

```
      3       3      39
```

We can even put another grep in between there to filter things:

```
In [41]: !find brown/. | xargs grep -l Kennedy | xargs grep -l nuclear | wc
```

```
     11      11     143
```

Finally, let's add our little `sed` script back in to format the output.

```
In [47]: !find brown/. | xargs grep -l Kennedy | xargs grep -h nuclear | sed 's/\/[^ ]*//g;s/^\s//' | head
```

```
Until Moscow resumed nuclear testing last September 1 , the US and UK had released more than
twice as much radiation into the atmosphere as the Russians , and the fallout from the earlier
blasts is still coming down .
On October 19 , after the Soviets had detonated at least 20 nuclear devices , Ambassador
Stevenson warned the UN General Assembly that this country , in `` self protection '' , might
have to resume above-ground tests .
Now , of course , that the Russians are the nuclear villains , radiation is a nastier word than
it was in the mid , when the US was testing in the atmosphere .
After a nuclear blast , one bureaucrat suggested in those halcyon days , about all you had to
do was haul out the broom and sweep off your sidewalks and roof .
Can thermonuclear war be set off by accident ? ?
`` E '' stands for `` execution '' -- the moment a `` go order '' would unleash an American
nuclear strike .
Work is under way to see whether new restraining devices should be installed on all nuclear
weapons .
Only the President is permitted to authorize the use of nuclear weapons .
The President cannot personally remove the safety devices from every nuclear trigger .
However , the system is designed , ingeniously and hopefully , so that no one man could
initiate a thermonuclear war .
sed: couldn't flush stdout: Broken pipe
```

```
In [ ]:
```