

to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try <https://etherpad.wikimedia.org>).

Users are expected to follow our code of conduct: [https://docs.carpentries.org/topic\\_folders/policies/code-of-conduct.html](https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html)

All content is publicly available under the Creative Commons Attribution License:  
<https://creativecommons.org/licenses/by/4.0/>

-----

Welcome everyone to the Software Carpentry 2020 Workshop at the Earlham Institute!

Your instructors for these 2 days are Nicola, Toni and Alice.

Code of Conduct: [https://docs.carpentries.org/topic\\_folders/policies/code-of-conduct.html](https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html)

This course uses material produced by Software Carpentry: <https://software-carpentry.org/about/>

Workshop page/Setup Instructions: <https://tgac.github.io/2020-01-21-earlham/>

Please interrupt at any moment if you have any question, or ask them here!

## **The Unix Shell**

First lesson: <https://swcarpentry.github.io/shell-novice/>

How to set aliases in Bash for Windows: <https://stackoverflow.com/questions/37104273/how-to-set-aliases-in-the-git-bash-for-windows>

## **Version Control with Git**

Second lesson: <https://swcarpentry.github.io/git-novice/>

GitHub: <https://github.com>

## **Programming with Python**

<https://swcarpentry.github.io/python-novice-inflammation/index.html>

Why list/array indexing starts at 0: [https://en.wikipedia.org/wiki/Zero-based\\_numbering](https://en.wikipedia.org/wiki/Zero-based_numbering)

question on the shell:

<https://swcarpentry.github.io/shell-novice/07-find/index.html>

exercise tracking a species: why do you need the

`cut -d : -f 2`

piece in your code?

A: Because ``grep -r`` returns lines in the form ``filename:match``, and the ``cut`` splits such lines on the ``:`` character and returns the second field, i.e. the match.

meaning that if you just do the grep command and return it on the screen, you will get:

filename:match

filename:match

filename:match,

etc???

because in my case (windows) that was not what I got....(I just got the match (in this case:

```
$ grep -w rabbit -r animal-counts/animals.txt
```

2012-11-

05,rabbit,22

2012-11-

06,rabbit,19

2012-11-

07,rabbit,16

That's probably because after the "-r" flag you are passing a file instead of a folder, while in the example exercise you are passing "." which would be the same as "animal-counts":

```
$ grep -w rabbit -r
```

animals.txt:2012-11-05,rabbit,22

animals.txt:2012-11-06,rabbit,19

animals.txt:2012-11-07,rabbit,16

```
grep -w rabbit -r animals.txt
```

2012-11-05,rabbit,22

2012-11-06,rabbit,19

2012-11-07,rabbit,16

thanks!