1. Welcome 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

---

ACENET: Programming Workshop: Unix Shell, Version Control and Python

Session #1: Unix Shell

Workshop website: https://acenet-arc.github.io/2020-09-21-ACENET_SWC-P/

Lesson link: http://swcarpentry.github.io/shell-novice/

Setup link: http://swcarpentry.github.io/shell-novice/setup.html

Attendance (please write your full name below):
Afshan Khaleghi
Jamseena Parayil
Alexandre Caouette
Ali Ahmadi
Varun Kumar Tirupathi
Jennifer Anderson
Parth Shah
Sarah Clarke
Neline Labuscgane
Patrick Godin
Amar Anwar
Yunxiang Deng
Adnane Ait Nasser
Denise Clark
Kanagasabapathy Natarajan
Naaman Omar

Friday Chen
Marie Babineau
Baru Wesley
Jennifer Anderson
Maria Kilfoil

Exercise #1:

Using the filesystem diagram below, if pwd displays /Users/thing, what will ls -F ../backup display?
1. ./backup: No such file or directory
2. 2012-12-01 2013-01-08 2013-01-27
3. 2012-12-01/ 2013-01-08/ 2013-01-27/
4. original/ pnas_final/ pnas_sub/

Exercise #2:

When run in the molecules directory, which ls command(s) will produce this output?

```
ethane.pdb   methane.pdb
```

1. ls t*ane.pdb
2. ls t?ne.*
3. ls t??ne.pdb
4. ls ethane.

Answers: (put a + next to the answer you think is correct)
1.
2
3.+++++++++
4.

Exercise #3:
In our current directory, we want to find the 3 files which have the least number of lines. Which command listed below would work?

1. wc -l > sort -n > head -n 3
2. wc -l | sort -n | head -n 1-3
3. wc -l | head -n 3 | sort -n
4. wc -l | sort -n | head -n 3

Answers: (put a + next to the answer you think is correct)
1.
2.

3.
4.++++++++

Exercise #4: Variables in Loops

Feedback: Please tell us one thing that you liked about today's session (+) and one thing that you didn't like or that you think could be improved (-)
+example based teaching and logical flow of concepts
-time spent on simple concepts disproportionate to that spent on harder concepts
+good use of examples
-could have spent less time on simple concepts
+friendly, great explaination and entertaining all the doubts.
-a little too beginner oriented. I had just a basic knowledge of bash going into it.
+good speed which gave us enough time to use the instructions along with others
-Maybe a little more advance examples
+Good course material
-could have went a little quicker on the basic unix shell commands.
+good refreshment for an old user!
-
+well explained
-
+great use of examples + the following along aspect
-could have gone faster at the beginning, but still fine overall!
+
-

Preparation for Session #2: Version Control with Github
If you haven't already completed the pre-workshop survey on the Workshop website please do that before Wednesady.

Click on this link: https://github.com/
Create an account and have your username and password handy for our session on Wednesday.

Make sure you've complete the Github installation for your operating system on the Workshop website: https://acenet-arc.github.io/2020-09-21-ACENET_SWC-P/

* ACENET: Programming Workshop: Unix Shell, Version Control and Python
***Session #2: Version Control with Git**

Workshop website: https://acenet-arc.github.io/2020-09-21-ACENET_SWC-P/

Lesson link: http://swcarpentry.github.io/git-novice/

Setup link: Create an account at this link: https://github.com/

Attendance (please write your full name below):

Neline Labuschagne
Denise Clark
Afshan Khaleghi
Sarah Clarke
Marie Babineau
Naaman Omar
Jamseena Parayil
Alexandre Caouette
Adnane AIT NASSER
Jennifer Anderson
Shirine Jeradi

Useful commands:

```
git config --list
git config --help       OR      git config -h
git config --global user.name "Your name"
git config --global user.email "your.email@some.where"
git config --global core.autocrlf true          OR      core.autocrlf input
    ("true" is usually preferred on Windows, "input" is preferred on MacOS or Linux)
```

Getting out of vim: <esc>:q!

```
cd; cd Desktop
mkdir planets
cd planets
git init
git status                          ALWAYS USEFUL!
nano mars.txt                       ...add some text to mars.txt
git add mars.txt
git commit -m "Start notes on Mars as a base"
git log
git diff                            # changes between working directory and staging area
git diff --staged                   # changes between staging-area and last commit
git log --oneline --graph --decorate  # several options to show log entries in different ways
git remote -v
git remote add origin  <URL.git>
git push origin master              OR    git push <name-of-remote> <name-of-branch>
git pull origin master              OR    git pull <name-of-remote> <name-of-branch>
git clone <URL.git> <optional-local-name>
```

Typical work cycle:
* git pull
* modify contents
* git add
* git commit
* git push

Jennifer has made changes to the Python script that she has been working on for weeks, and the modifications she made this morning “broke” the script and it no longer runs. She has spent ~ 1hr trying to fix it, with no luck…
Luckily, she has been keeping track of her project’s versions using Git! Which commands below will let her recover the last committed version of her Python script called data_cruncher.py?

1. $ git checkout HEAD
2. $ git checkout HEAD data_cruncher.py
3. $ git checkout HEAD~1 data_cruncher.py
4. $ git checkout <unique ID of last commit> data_cruncher.py
5. Both 2 and 4

Answers:
1)
2)++
3)
4)
5)++++++

https://xkcd.com/1296/

https://xkcd.com/1597/

http://phdcomics.com/comics.php?n=1531

https://docs.github.com/en/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#generating-a-new-ssh-key

Preparation for Session #3: Python (part 1)
Follow the instructions to install Python for your Operating System on the workshop website:
https://acenet-arc.github.io/2020-09-21-ACENET_SWC-P/

Feedback: Please tell us one thing that you liked about today's session (+) and one thing that you didn't like or that you think could be improved (-)
+Very thorough explanation of github
-Use more interactive questions by asking more self-assessment exercises
+
-It would be nice to summarize a bloc at the end (before starting a new one)
+It was quiet usefull and practical
-it would have been way better if there were more exercises since there is so much to learn and the time is limited
+reassurances that it takes time to master were appreciated
-more discussion about applications of github, and possible alternatives (are there any?)
+

-

* ACENET: Programming Workshop: Unix Shell, Version Control and Python
*Session #3: Python (part 1)

Workshop website: https://acenet-arc.github.io/2020-09-21-ACENET_SWC-P/

Lesson link: http://swcarpentry.github.io/python-novice-gapminder/

Attendance (please write your full name below):
Adnane Ait Nasser
Alexandre Caouette
Neline Labuschagne
Sarah Clarke
Afshan Khaleghi
Shirine Jeradi
Marie Babineau
Naaman Omar
Denise Clark
Varun Tirupathi
Patrick Godin
Ali Ahmadi
Jennifer Anderson
Maria Kilfoil

Exercise #1:

print(x, y, swap)

 # Command # Value of x # Value of y # Value of swap #
x = 1.0 # # # #
y = 3.0 # # # #
swap = x # # # #
x = y # # # #
y = swap # # # #

Exercise #2:

Fractions
What type of value is 3.4? How can you find out?

Exercise #3:
Automatic Type Conversion
What type of value is 3.25 + 4?

Exercise #4:
Choose a Type
What type of value (integer, floating point number, or character string) would you use to represent each of the following? Try to come up with more than one good answer for each problem. For example, in # 1, when would counting days with a floating point variable make more sense than using an integer?

Number of days since the start of the year.
Time elapsed from the start of the year until now in days.
Serial number of a piece of lab equipment.
A lab specimen's age
Current population of a city.
Average population of a city over time.

EXCERCISE 5
Which of the following will return the floating point number 2.0? Note: there may be more than one right answer.
first = 1.0second = "1"third = "1.1"

1. first + float(second)
2. float(second) + float(third)
3. first + int(third)
4. first + int(float(third))
5. int(first) + int(float(third))
6. 2.0 * second

http://swcarpentry.github.io/python-novice-gapminder/04-built-in/index.html

Excercise 6

Given this:
print('string to list:', list('tin'))**print**('list to string:', ''.join(['g', 'o', 'l', 'd']))
['t', 'i', 'n']
'gold'

1. What does list('some string') do?
2. What does '-'.join(['x', 'y', 'z']) generate?

Excercise 7

Reversing a String
Fill in the blanks in the program below so that it prints "nit" (the reverse of the original character string "tin").

original = "tin"

```
result = ____
for char in original:
    result = ____
print(result)
```

Excercies 8

Practice Accumulating
Fill in the blanks in each of the programs below to produce the indicated result.

```
# Total length of the strings in the list: ["red", "green", "blue"] => 12total = 0for word in ["red", "green", "blue"]:
    ____ = ____ + len(word)print(total)

# List of word lengths: ["red", "green", "blue"] => [3, 5, 4]lengths = ____for word in ["red", "green", "blue"]:
    lengths.____(____)print(lengths)

# Concatenate all words: ["red", "green", "blue"] => "redgreenblue"words = ["red", "green", "blue"]result = ____for ____ in ____:
    ____print(result)


# Create acronym: ["red", "green", "blue"] => "RGB"
# write the whole thing
```

Solution

Feedback: Please tell us one thing that you liked about today's session (+) and one thing that you didn't like or that you think could be improved (-)
+It was a great introduction and althougt the speed could have increases in the bigining it gave me enough time to catch up and explore a bit extra
-I would like to have more complicated examples
+excellent in terms of speed and clarity
-I would prefer a command line environment
+
-

* **ACENET: Programming Workshop: Unix Shell, Version Control and Python**
***Session #4: Python (part 2)**

Workshop website: https://acenet-arc.github.io/2020-09-21-ACENET_SWC-P/

Lesson link: http://swcarpentry.github.io/python-novice-gapminder/

Attendance (please write your full name below):

Adnane Ait Nasser
Denise Clark

Afshan Khaleghi
Alexandre Caouette
Jennifer Anderson
Marie Babineau
Jamseena Parayil
Sarah Clarke
Neline LAbuschagne
Naaman Omar

Question 1
Read the data in gapminder_gdp_americas.csv (which should be in the same directory as gapminder_gdp_oceania.csv) into a variable called americas and display its describe function, display Transpose and information.

Assume Pandas has been imported into your notebook and the Gapminder GDP data for Europe has been loaded:
import pandas as pd
df = pd.read_csv('data/gapminder_gdp_europe.csv', index_col='country')

Question 2
Write an expression to find the Per Capita GDP of Serbia in 2007.

Question 3
Assume Pandas has been imported and the Gapminder GDP data for Europe has been loaded. Write an expression to select each of the following:

1. GDP per capita for all countries in 1982.
2. GDP per capita for Denmark for all years.
3. GDP per capita for all countries for years *after* 1985.
4. GDP per capita for each country in 2007 as a multiple of GDP per capita for that country in 1952.

Question 4
Fill in the blanks below to plot the minimum GDP per capita over time for all the countries in Europe. Modify it again to plot the maximum GDP per capita over time for Europe.
data_europe = pd.read_csv('data/gapminder_gdp_europe.csv', index_col='country')data_europe.____.plot(label='min')data_europe.____plt.legend(loc='best')plt.xticks(rotation=90)

Question 5
Modify the example in the notes to create a scatter plot showing the relationship between the minimum and maximum GDP per capita among the countries in Asia for each year in the data set. What relationship do you see (if any)?
data_asia = pd.read_csv('data/gapminder_gdp_asia.csv', index_col='country')data_asia.describe().T.plot(kind='scatter', x='min', y='max')

Whenever you are generating plots to go into a paper or a presentation, there are a few things you can do to make sure that everyone can understand your plots.

- Always make sure your text is large enough to read. Use the fontsize parameter in xlabel, ylabel, title, and legend, and tick_params with labelsize to increase the text size of the numbers on your axes.
- Similarly, you should make your graph elements easy to see. Use s to increase the size of your scatterplot markers and linewidth to increase the sizes of your plot lines.
- Using color (and nothing else) to distinguish between different plot elements will make your plots unreadable to anyone who is colorblind, or who happens to have a black-and-white office printer. For lines, the linestyle parameter lets you use different types of lines. For scatterplots, marker lets you change the shape of your points. If you're unsure about your colors, you can use Coblis or Color Oracle to simulate what your plots would look like to those with colorblindness.

Which of these files is *not* matched by the expression glob.glob('*as*.csv')?

1. data/gapminder_gdp_africa.csv
2. data/gapminder_gdp_americas.csv
3. data/gapminder_gdp_asia.csv
4. 1 and 2 are not matched.

1. Read this short program and try to predict what it does.
2. Run it: how accurate was your prediction?
3. Refactor the program to make it more readable. Remember to run it after each change to ensure its behavior hasn't changed.
4. Compare your rewrite with your neighbor's. What did you do the same? What did you do differently, and why?

```
n = 10s = 'et cetera'print(s)i = 0while i < n:
    # print('at', j)    new = ''
    for j in range(len(s)):
        left = j-1
        right = (j+1)%len(s)
        if s[left]==s[right]: new += '-'
        else: new += '*'
    s=''.join(new)
    print(s)
    i += 1
```

Feedback: Please tell us one thing that you liked about today's session (+) and one thing that you didn't like or that you think could be improved (-)
+Going through examples and give us time to work on it was a great practise. It was a great class
-the only downside to that was that not everone were intereste in working on those and i felt a bit rushed working on some. I'd rather feel like it is must exersice at least a couple of harder ones
+very useful introduction. I appreciate there is much more to learn now.
-suggest what to read before the class. would sometime help to understand the purpose of a script
+Great introduction to all the concepts!
-
+
-
+
-
+
-
+
-
+
-
+
-
+
-
+
-
+
-