Lab A: Using Python as a Calculator

2020 Summer — Calculus 1

Dr Matthew H Sunderland

☞ Launch Binder (https://mybinder.org/v2/gh/mattsunderland/pycalclab/master?filepath=calc1A.ipynb)
☞ Download Notebooks (https://github.com/mattsunderland/pycalclab)

Jupyter Notebooks

A1. Press **SHIFT+RETURN** to run a cell. Run a cell, and Jupyter displays the **last** output. Press **ESCAPE,r,y** to clear the output.
Try it! Run the following cell. Then clear the output.

In [ ]:
```
1 + 2 + 3
50-3
100*5
```

A2. Use a **comma** to display multiple outputs.
Try it! Run the following cell.

In [ ]:
```
25*3, 100-1
```

A3. The toolbar's "+" button adds a cell. The toolbar's scissors button deletes a cell.
Try it! Add some cells below. Then delete some of them.

A4. Click "Code" or "**Markdown**" on the toolbar to change cell mode. To render a markdown cell when you're finished editing it, press SHIFT+RETURN. To resume editing, **double click** the cell.
Try it! ADD a cell, CHANGE to markdown mode, TYPE something, RENDER it.

A5 Exercise. Math written "in LaTeX" can be put between dollar signs in markdown cells.
CHANGE the following cell to markdown mode, RENDER it.

In [ ]:
```
####  $\sqrt{b^2 - 4ac}$
```

Python arithmetic + - * / **

A6. Run the following.

In [ ]:
```
3 + 10*5,  5**2,  27/10
```

A7 Exercise. Based on A6, what does each of the 5 arithmetic operations do?
Type your answer here:

A8 Exercise. Do spaces around the 5 operations matter, or is it just style?
Type your answer here:

Python # and =

A9. Python will ignore the rest of a line after it sees #
Try it! Run the following, but first try to *predict* the outputs.

In [ ]:
```
# This is a comment
1 + 1 # This is also a comment
```

A10. Assign a value to a variable using the = symbol. Variable names must start with a letter. Jupyter does not automatically display anything when you assign a value; you have to **call the variable again** to get it to display its value.
Try it! Run the following, but first try to *predict* the outputs.

In [ ]:
```
a = 10
a
```

In [ ]:
```
b = 20
```

In [ ]:
```
a = 18
b = 21
c = a - b
c
```

A11. You can assign more than one variable at a time using **a comma.**
Try it! Run the following, but first try to *predict* the outputs.

In [ ]:
```
x, y = 100, 500
x
```

A12 Exercise. What would be the output of the commands

```
a,b,c = 3,4,5
a + b/c
```

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

A13. Assigning variables can help with complicated expressions. For example, we can do $\frac{(2-3)*-3}{-1+2}$ all at once or in steps.
Try it! Run the following.

In [ ]:
```
(2 - 3)*-3/(-1 + 2)
```

In [ ]:
```
top = (2 - 3)*-3
bottom = -1 + 2
top/bottom
```

A14 Exercise. Assign variables to help you compute $3 - \frac{3^2-2\cdot3}{2\cdot3-2}$

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Order of Operations

A15. We can use Python to show that $a - b * c$ equals $a - (b * c)$ but doesn't equal $(a - b) * c$.
Try it! Run the following cell.

In [ ]:
```
a - b*c,  a - (b*c),  (a - b)*c
```

A16 Exercise. One in each row is different from the rest.

```
Mark one with an x:     [ ]  a*(b - c)     [ ]  (a*b) - c     [ ]  a*b - c
Mark one with an x:     [ ]  a/b + c       [ ]  a/(b + c)     [ ]  (a/b) + c
Mark one with an x:     [ ]  (a + b)/c     [ ]  a + (b/c)     [ ]  a + b/c
```

A17 Exercise. One in each row is different from the rest.

```
Mark one with an x:     [ ]  a**(b*c)      [ ]  (a**b)*c      [ ]  a**b*c
Mark one with an x:     [ ]  a*(b**c)      [ ]  a*b**c        [ ]  (a*b)**c
Mark one with an x:     [ ]  a/b**c        [ ]  (a/b)**c      [ ]  a/(b**c)
Mark one with an x:     [ ]  a**b/c        [ ]  (a**b)/c      [ ]  a**(b/c)
```

A18 Exercise. We can illustrate the 2 operations in `a + b + c` using 2 sets of parentheses

```
((a + b) + c)
```

Illustrate the 4 operations in `7 - 3**2/9 + 4` using 4 sets of parentheses

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

A19 Exercise. One in each row is different from the rest.

```
Mark one with an x:     [ ]  3 - 3 - 3     [ ]  (3 - 3) - 3    [ ]  3 - (3 -
 3)
Mark one with an x:     [ ]  2**3**2       [ ]  (2**3)**2      [ ]  2**(3**2)
Mark one with an x:     [ ]  6/3/2         [ ]  (6/3)/2        [ ]  6/(3/2)
```

A20 Exercise. Assign `a,b,c = 4,5,8` and then evaluate $\frac{a^b - c/b}{c-a}$, $\frac{a^{c-b}}{c-b}$, $\frac{a^{3/2}}{b}$, $\frac{a-b(c-a)}{c-a}$

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Making python functions

A21. Run the following.

In [ ]:
```
def f(x):
    return x**2

f(7)
```

A22 Exercise. Make the function $P(x) = x^2 - 2x + 1$ and find $P(P(7))$.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Numpy builtin functions via `%pylab`

Python	Math notation	Meaning
`abs(x)`	$\|x\|$	absolute value
`sqrt(x)`	$\sqrt{x}$	square root
`exp(x)`	e^x	exponential function
`log(x)`	$\ln x$	natural logarithm
`sin(x)`	$\sin x$	sine
`arcsin(x)`	$\sin^{-1} x$	inverse sine

A23. Anaconda includes standard data science packages for python. **Execute the command `%pylab`** to load the packages `numpy` and `matplotlib`. The package `numpy` comes with many builtin math functions (which work with arrays, see A27), including those listed in the table above.
Try it! Run the following.

In [ ]:
```
%pylab

sqrt(49)
```

In [ ]:
```
pi, exp(1)
```

In [ ]:
```
x = pi/3
tan(x),  sin(x)**2 + cos(x)**2
```

A24. Type a question mark `?` after a function to pull up documentation. *Try it! Run the following.*

In [ ]:
```
sqrt?
```

A25 Exercise. Evaluate $\sin 40^\circ, \sin^2 65^\circ, e^{(10-8.5)/3}, \arcsin(\sin(3\pi/4))$ Note. Python uses radians for all angle measurements, so you need to convert any degrees to radians.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Numpy Arrays with `r_[ ]`

A26. Use the function `r_[ ]` to make an array of numbers of your choice. The function `r_[ ]` is part of the numpy package and is automatically loaded when you execute the Jupyter command `%pylab`. An **array** is a list of numbers that you can do arithmetic on and apply functions to.
Note. If you get `NameError: name 'r_' is not defined` go back and run A27.

In [ ]:
```
x = r_[0, pi/6, pi/4]
cos(x)
```

In [ ]:
```
x = r_[1,2,5,10]
x**2
```

A27 Exercise. Use `r_[ ]` to store the numbers 2,3,5 in an array named `x`. Find `x+x` and `x*x`.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

Numpy Arrays with `r_[a:b:stride]`

A28. Use `r_[a:b]` list integers **from a up to but *not* including b.** Omit the a and the default is 0.
Try it! Run the following.

In [ ]:
```
r_[-10:10]
```

In [ ]:
```
r_[:5]
```

A29 Exercise. Use `r_[a:b]` to make the array 1,2,3,4,5,6,7,8,9?

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

A30. Use `r_[a:b:stride]` to space out your numbers by the amount `stride`
Try it! Run the following.

In [ ]:
```
r_[0:100:2]
```

A31 Exercise. Use `r_[a:b:stride]` to make the arrays $1, 3, 5, \ldots, 99$ and then $10, 20, 30, \ldots, 100$.

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

In [ ]:
```

```

Numpy arrays with `linspace(a,b,n)`

A32. Use `linspace(a,b,n)` to make a list of numbers **n numbers from a to b inclusive.** This is useful when we want to generate a lot of evenly-spaced numbers but don't want to have to figure out the stride length. If you omit the n then the default n is 50.
Try it! Run the following.

In [ ]:
```
linspace(0,10,6)
```

In [ ]:
```
linspace(0,10)
```

A33 Exercise. Use `linspace(a,b,n)` to make the array $1, 1.5, 2, 2.5, 3, 3.5, 4$

In [ ]:
```
# Type your answer below and press SHIFT+ENTER
```

A34 Exercise. What is the last value output by the command `linspace(0,pi)`

```
In [ ]: # Type your answer below and press SHIFT+ENTER
```

Celsius and Fahrenheit

A35. Convert between Celsius and Fahrenheit using $F = 9/5C + 32$ and $C = 5/9(F - 32)$.

A36 Exercise. Convert room temperature $68^\circ F$ and average body temperature $98.6^\circ F$ to Celsius.

```
In [ ]: # Type your answer below and press SHIFT+ENTER
```

A37. Put two arrays `x` and `y` into a 2D array using `r_[[x,y]]`. Transpose a 2D array using `.T` Try it! Run the following.

```
In [ ]: x = r_[:10]
        y = x**2
        r_[[x,y]]
```

```
In [ ]: x = r_[:10]
        y = x**2
        r_[[x,y]].T
```

A38 Exercise. Use `r_` to make an array of Fahrenheit values `x` $= -100, -80, -60, \ldots, 100$ and the corresponding array of Celsius values `y`. Put them together in a 2D array and transpose.

```
In [ ]: # Type your answer below and press SHIFT+ENTER
```

Double-click here to see the answer.

Downloading Anaconda (free, open-source)

A40 Homework. Download Anaconda to your own computer from www.anaconda.com/products/individual

After installing, go to the program **Anaconda Navigator**, find JupyterLab, click "Launch." To start a new .ipynb notebook, go to File > New > Notebook > Python 3. To open a .ipynb notebook, click the folder icon on the far left and then navigate to the file.