Lab A: Using Python as a Calculator

mybinder.org/v2/gh/anniebmcc/pycalclab/master

2020 Summer — Calculus 1

Dr Matthew H Sunderland

Jupyter Notebooks

A1. **RUN** the following "code cell" (gray rectangle with `In[ ]` next to it), by CLICKING the code cell and pressing SHIFT+RETURN. Notice that only the last result will display.

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

```
Out[1]: 500
```

A2. **RUN** the following. As always, only the last result displays, but the last result has 2 parts because of the comma.

```
In [2]: 1 + 2 + 3
        50 - 3, 1000*1000
        100*5, 7*7
```

```
Out[2]: (500, 49)
```

A3. The "+" on the toolbar adds a code cell. The "scissors" deletes a cell.

Python arithmetic `+ - * / **`

A4. **RUN** the following.

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

```
Out[3]: (53, 25, 2.7)
```

A5. **EXERCISE.**

a) What does each of the 5 arithmetic operations do?

b) Do spaces around the 5 operations matter, or is it just style?

In [4]:
```
# TYPE YOUR ANSWERS BELOW
#
#  a) + is   addition
#     - is   subtraction
#     * is   multiplication
#     / is   division
#    ** is   exponentiation
#
#  b) No, spaces aroung + - * / ** don't matter
```

Python # and =

A6. **RUN** the following. You will notice python ignores everything after #

In [5]:
```
# This is a comment
1 + 1 # This is also a comment
```

Out[5]: 2

A7. **RUN** the following. Notice we assign variables using = Assignment itself does NOT produce output.

In [6]:
```
a = 10
a
```

Out[6]: 10

In [7]:
```
b = 20
```

In [8]:
```
a = 18
b = 21
c = a - b
c
```

Out[8]: -3

A8. **RUN** the following. Notice you can assign multiple variables at once with a comma.

In [9]:
```
x, y = 100, 500
x
```

Out[9]: 100

In [10]:
```
a,b,c = 3,4,5
a + b/c
```

Out[10]: 3.8

A9. **RUN** the following. See that we can compute $\frac{(2-3)*-3}{-1+2}$ all at once (1st cell below), or we can assign variables to help us (2nd cell below).

In [11]:
```
(2 - 3)*-3/(-1 + 2)
```

Out[11]: 3.0

In [12]:
```
top = (2 - 3)*-3
bottom = -1 + 2
top/bottom
```

Out[12]: 3.0

A10. **EXERCISE.** Assign variables to help you compute $3 - \frac{3^2-2\cdot3}{2\cdot3-2}$

In [13]:
```
# Type your answer below and press SHIFT+ENTER

top = 3**2 - 2*3
bottom = 2*3 - 2
3 - top/bottom
```

Out[13]: 2.25

Order of Operations

A11. **RUN** the following. Notice $a - b * c = a - (b * c)$, but they do not equal $(a - b) * c$.

In [14]:
```
a,b,c = 3,4,5

a - b*c,  a - (b*c),  (a - b)*c
```

Out[14]: (-17, -17, -5)

A12. **EXERCISE.** In each row, identify NON-equivalent choice. For example, the answer to (1) is $(a - b) * c$ because $a - b * c = a - (b * c)$

(1)	$a - b * c$	$a - (b * c)$	$(a - b) * c$
(2)	$a * (b - c)$	$(a * b) - c$	$a * b - c$
(3)	$a / b + c$	$a / (b + c)$	$(a / b) + c$
(4)	$(a + b) / c$	$a + (b / c)$	$a + b / c$
(5)	$a ** (b * c)$	$(a ** b) * c$	$a ** b * c$
(6)	$a * (b ** c)$	$a * b ** c$	$(a * b) ** c$
(7)	$a/b ** c$	$(a/b) ** c$	$a/(b ** c)$
(8)	$a ** b/c$	$(a ** b)/c$	$a ** (b/c)$
(9)	$(3 - 3) - 3$	$3 - 3 - 3$	$3 - (3 - 3)$
(10)	$(2 ** 3) ** 2$	$2 ** (3 ** 2)$	$2 ** 3 ** 2$
(11)	$6 / 3 / 2$	$6 / (3 / 2)$	$(6 / 3) / 2$

In [15]:
```
# TYPE YOUR ANSWERS BELOW.
#
#  (1)   (a - b)*c
#  (2)   a*(b - c)
#  (3)   a/(b + c)
#  (4)   (a + b)/c
#  (5)   a ** (b*c)
#  (6)   (a*b) ** c
#  (7)   (a/b) ** c
#  (8)   a ** (b/c)
#  (9)   3 - (3 - 3)
# (10)   (2 ** 3) ** 2
# (11)   6/(3/2)
```

A13. **RUN** the following example, where we add 2 sets of parentheses which show the order of the 2 operations.

In [16]:
```
1 + 3/5
```

Out[16]: 1.6

In [17]:
```
(1 + (3/5))
```

Out[17]: 1.6

A14. **EXERCISE.** Add 4 sets of parentheses, which show the order of the 4 operations.

In [18]:
```
7 - 3 ** 2/9 + 4
```

Out[18]: 10.0

In [19]:
```
# Type your answer below and press SHIFT+ENTER

((7 - ((3 ** 2)/9)) + 4)
```

Out[19]: 10.0

A15. **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 [20]:
```
# Type your answer below and press SHIFT+ENTER

a,b,c = 4,5,8

(a**b - c/b)/(c-a),  a**(c-b)/(c-b),  a**(3/2)/b,  (a - b*(c-a))/(c-a)
```

Out[20]: (255.6, 21.333333333333332, 1.6, -4.0)

Making python functions

A16. **RUN** the following.

In [21]:
```
def g(x):
    return x**2

g(7)
```

Out[21]: 49

In [22]:
```
def h(n): return n + 100

h(7)
```

Out[22]: 107

A17. **EXERCISE.** Make the function $P(x) = x^2 - 2x + 1$ and find $P(P(7))$.

In [23]:
```
# Type your answer below and press SHIFT+ENTER

def P(x):
    return x**2 - 2*x + 1

P(P(7))
```

Out[23]: 1225

Built-in `%pylab` functions

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
`radians(x)`		converts degrees to radians

A18. **RUN** the code cells below. The command `%pylab` only needs to be run once per lab; it loads "built-in functions" (from python packages numpy and matplotlib).

In [24]:
```
%pylab

sqrt(49)
```

```
Using matplotlib backend: MacOSX
Populating the interactive namespace from numpy and matplotlib
```

Out[24]:
```
7.0
```

In [25]:
```
pi, exp(1), sin(pi/2)
```

Out[25]:
```
(3.141592653589793, 2.718281828459045, 1.0)
```

A19. **EXERCISE.** Evaluate

1. $\sin 40^\circ$
2. $\sin^2 65^\circ$
3. $e^{(10-8.5)/3}$
4. $\arcsin(\sin(3\pi/4))$

Note. Python uses radians for all angle measurements, so you need to convert any degrees to radians.

In [26]:
```
# Type your answer below and press SHIFT+ENTER

sin(radians(40)),  sin(radians(65))**2,  exp((10-8.5)/3),  arcsin(sin(3*pi/
4))
```

Out[26]:
```
(0.6427876096865393,
 0.8213938048432696,
 1.6487212707001282,
 0.7853981633974484)
```

Making an array with `r_[ ]`

A20. **RUN** the following. (If you get an error, go back and run A17.) The function `r_[ ]` can make an array of numbers of your choice. We will need arrays for graphing (Lab B).

```
In [27]: x = r_[2,3,4,5,10]
         x**3

Out[27]: array([   8,   27,   64,  125, 1000])
```

A21. **EXERCISE.** Use `r_[ ]` to store the numbers 2,3,5,7,11 in an array named `x` . Find `x*x` .

```
In [28]: # Type your answer below and press SHIFT+ENTER

         x = r_[2,3,5,7,11]
         x*x

Out[28]: array([  4,   9,  25,  49, 121])
```

Making an array with `r_[a:b:stride]`

A22. **RUN** the following. In general, `r_[a:b]` will list integers from a up to but *not* including b. A missing a is the same as 0.

```
In [29]: r_[5:10]

Out[29]: array([5, 6, 7, 8, 9])
```

```
In [30]: r_[:5]

Out[30]: array([0, 1, 2, 3, 4])
```

A23. **EXERCISE.** Use `r_[a:b]` to make the array 1,2,3,4,5,6,7,8,9

```
In [31]: # Type your answer below and press SHIFT+ENTER

         r_[1:10]

Out[31]: array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

A24. **RUN** the following. In general, `r_[a:b:stride]` spaces out your numbers by the amount `stride` .

```
In [32]: r_[0:100:2]

Out[32]: array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32,
                34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66,
                68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94, 96, 98])
```

A25. **EXERCISE.** Use `r_[a:b:stride]` to make the array $1, 3, 5, \ldots, 99$

```
In [33]: # Type your answer below and press SHIFT+ENTER

         r_[1:100:2]
```

```
Out[33]: array([ 1,  3,  5,  7,  9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33,
                35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67,
                69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99])
```

Making an array with `linspace(a,b,n)`

A26. **RUN** the following. Observe that `linspace(a,b,n)` lists n numbers from a to b inclusive. This is useful for generating a lot of evenly-spaced numbers, such as when graphing (Lab B). Observe that `linspace(a,b)` lists 50 numbers from a to b inclusive.

```
In [34]: linspace(0,10,6)
```

```
Out[34]: array([ 0.,  2.,  4.,  6.,  8., 10.])
```

```
In [35]: linspace(0,10)
```

```
Out[35]: array([ 0.        ,  0.20408163,  0.40816327,  0.6122449 ,  0.81632653,
                 1.02040816,  1.2244898 ,  1.42857143,  1.63265306,  1.83673469,
                 2.04081633,  2.24489796,  2.44897959,  2.65306122,  2.85714286,
                 3.06122449,  3.26530612,  3.46938776,  3.67346939,  3.87755102,
                 4.08163265,  4.28571429,  4.48979592,  4.69387755,  4.89795918,
                 5.10204082,  5.30612245,  5.51020408,  5.71428571,  5.91836735,
                 6.12244898,  6.32653061,  6.53061224,  6.73469388,  6.93877551,
                 7.14285714,  7.34693878,  7.55102041,  7.75510204,  7.95918367,
                 8.16326531,  8.36734694,  8.57142857,  8.7755102 ,  8.97959184,
                 9.18367347,  9.3877551 ,  9.59183673,  9.79591837, 10.        ])
```

A27. **EXERCISE.** Use `linspace(a,b,n)` to make the array $1, 1.5, 2, 2.5, 3, 3.5, 4$

```
In [36]: # Type your answer below and press SHIFT+ENTER

         linspace(1,4,7)
```

```
Out[36]: array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. ])
```

A28. **EXERCISE.**
Convert average body temperature $98.6^\circ F$ to Celsius using $C = 5/9(F - 32)$.

```
In [37]: # Type your answer below and press SHIFT+ENTER

         5/9*(98.6 - 32)
```

```
Out[37]: 37.0
```

A29. **RUN** the following.
Notice that `x` and `y` are arrays,
`c_[x,y]` puts them into a table.

In [38]:
```
x = r_[:10]
y = x**2
c_[x,y]
```

Out[38]:
```
array([[ 0,  0],
       [ 1,  1],
       [ 2,  4],
       [ 3,  9],
       [ 4, 16],
       [ 5, 25],
       [ 6, 36],
       [ 7, 49],
       [ 8, 64],
       [ 9, 81]])
```

A30. **EXERCISE.**
Use `r_` to make an array of Fahrenheit values `x` $= -100, -80, -60, \ldots, 100.$
Make the corresponding array of Celsius values `y`
Use `c_` to put `x` and `y` into a table.

In [39]:
```
# Type your answer below and press SHIFT+ENTER

x = r_[-100:101:20]
y = 5/9*(x - 32)
c_[x,y]
```

Out[39]:
```
array([[-100.        ,  -73.33333333],
       [ -80.        ,  -62.22222222],
       [ -60.        ,  -51.11111111],
       [ -40.        ,  -40.        ],
       [ -20.        ,  -28.88888889],
       [   0.        ,  -17.77777778],
       [  20.        ,   -6.66666667],
       [  40.        ,    4.44444444],
       [  60.        ,   15.55555556],
       [  80.        ,   26.66666667],
       [ 100.        ,   37.77777778]])
```