Session 1: Python Fundamentals

Today we will

- Introduce Jupyter Notebooks
- Store data in variables
- Understand what types of data we can work with, and how to convert them.

Reference and Resource This lesson is adapted from Software Carpentry. (http://swcarpentry.github.io/python-novice-gapminder/design/)

Part 1. Introducing Jupyter Notebooks

CELLS - Markdown versus Code

This is a markdown cell. It renders text as HTML.

I can type in **bold**

- I can have bullet points

I can add LaTex like:

$\sqrt{2 + 3^8}$

and

$\sum_{i=1}^{N} 2^{-i} \approx 1$

Question 1. Do it yourself! Markdown

Your turn! Add something in markdown here and Shift+Enter to run the cell.

In [27]:

```
# This is a python code cell
# We will add and run python in code cells

x = 6 * 7 + 12
print(x)
```

```
54
```

In [2]:

```
# What if I use x again in a different cell?
x - 20
```

Out[2]:

```
34
```

Variables persist between cells once they have been run (executed)

Question 2. Do it yourself! Coding

In [17]:

```
#  INSTRUCTIONS:   Write a message in the quotes.
#  type Shift+Enter to run the cell.
message = ''


print(message)
```

Useful Shortcuts

See more under Help > Keyboard Shortcuts

There are 2 modes for a cell:

- Edit mode (blue box) and
- command mode (green box)

Toggle between them with ESC and Enter (Return)

Run a cell with Ctrl + Enter (Return)

Add a cell above with A Add a cell below with B

Part 2. Variables and Assignment

How can I store data in programs? Answer: Variables! They are names for values.

In [4]:

```
age = 36
first_name = 'Claire'
```

What's in a name? *Variable name conventions*

- Use only letters, digits, and underscores _
- Start with a letter (typically lower case)
- Variable names are case sensitive
- Use meaningful names!

The equals sign = assigns a value to the variable.

```
In [7]:  # Python function print prints things as text
         print(first_name)
         print(first_name,'is',age,'years old.')
```

```
Claire
Claire is 36 years old.
```

Variables must be created before they are used.

```
In [16]:  print(last_name)
          # What happens if I try to correct my error in the same cell?
          last_name='Pontbriand'
          print(last_name)
```

```
Pontbriand
```

Question 3.

```
In [39]:  # What will happen if I run this code?
          last_name = Pontbriand
          print(last_name)
```

```
-----------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
<ipython-input-39-7f978fa7f54d> in <module>()
      1 # What will happen if I run this code?
----> 2 last_name = Pontbriand
      3 print(last_name)

NameError: name 'Pontbriand' is not defined
```

Variables can be used in calculations.

Question 4. Outputs

In [12]:
```
# What will the output of the python code be?
age = age + 3
print('Age in three years:',age)
```

```
Age in three years: 39
```

Use an index to get a single character from a string

- Use square brackets for the position
- Indices are numbered from 0

In [18]:
```
atom_name = "helium"
print(atom_name[0])
print(atom_name[1])
print(atom_name[2])
print(atom_name[-1])
```

```
h
e
l
m
```

Use a slice to get a substring

- Take a slice with `[start:stop]` but mathematically it selects $[start : stop)$ indices

Question 5. Slicing strings

What will the output of the following slice be?

In [19]:
```
atom_name[3:5]
```

Out[19]:
```
'iu'
```

Question 6. Swapping Values

Given the code below, what is the value of the variable `swap` ?

In [28]:
```
x    = 1.0
y    = 3.0
swap = x
x    = y
y    = swap
```

In [29]:
```
print(swap)
```

```
1.0
```

Question 7 Challenge.

Assign a = 123. What happens if you try to get the second digit with a[1]?

In [30]:

```
a = 123
print(a[1])
```

```
---------------------------------------------------------------------------
----
TypeError                                 Traceback (most recent call l
ast)
<ipython-input-30-c761901e9428> in <module>()
      1 a = 123
----> 2 print(a[1])

TypeError: 'int' object is not subscriptable
```

In [31]:

```
a = str(123)
print(a[1])
```

```
2
```

Part 3. Data Types and Conversion

Data Types:

- integers (`int`) represent positive or negative whole numbers like 3 or -512
- floating point numbers (`float`) represent real numbers like 3.12159 or -2.5
- character strings (`str`) are text
  - written with single or double quotes (matching)
  - quotations aren't printed when the string is displayed

In [54]:

```
# Find the type with function type()
print(type(52))
print(type(age))
print(type(first_name))
print(type(3.14))

# notice we are nesting functions.
```

```
<class 'int'>
<class 'int'>
<class 'str'>
<class 'float'>
```

Data types control what operations (or methods) can be performed on a value.

```
In [40]: print(5-3)
         print('hello'-'h')
```

```
2

---------------------------------------------------------------------------
----
TypeError                                 Traceback (most recent call l
ast)
<ipython-input-40-149e748127f6> in <module>()
      1 print(5-3)
----> 2 print('hello'-'h')

TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

You can use the + and * operators on strings.

```
In [43]: print(type(first_name))
         full_name = first_name + ' ' + 'Pontbriand'
         print(full_name)
```

```
<class 'str'>
Claire Pontbriand
```

Strings have length (`len()`), but numbers don't.

```
In [46]: print(len(full_name))
         print(len(3.1415))
```

```
17

---------------------------------------------------------------------------
----
TypeError                                 Traceback (most recent call l
ast)
<ipython-input-46-775fd20633b7> in <module>()
      1 print(len(full_name))
----> 2 print(len(3.1415))

TypeError: object of type 'float' has no len()
```

We must convert numbers to strings or vice versa when operating on them. Consistency is key!

```
In [49]: # print(1+'2')
         print(1+int('2'))
         print(str(1)+'2')
```

```
3
12
```

We can mix integers and floats freely in operations.

In [51]:

```
print('half is', 1/2.0)
print('three squared is', 3.0 ** 2)
```

```
half is 0.5
three squared is 9.0
```

Variables only change value when something is assigned to them. They are **not** like spreadsheets where a cell can depend on another and update automatically.

In [53]:

```
first  = 1
second = 5 * first
first  = 2
print('first is', first, 'and second is', second)
```

```
first is 2 and second is 5
```

Question 8. Choose a type (int, float, str)

Time elapsed from the start of the year until now in days.

Question 9. Choose a type

Serial cod of a piece of lab equipment

Question 10. Choose a type

A lab specimen's age

Division Types with numbers

- // operator performs integer floor division
- / operator performs floating point division
- % modulo operator returns the remainder from integer division

In [56]:

```
print(5//3)
print(5/3)
print(5%3)
```

```
1
1.6666666666666667
2
```

Challenge Project

Work with your group come back, and vote your answers!

In [61]:

```
first  = 1.0
second = "1"
third  = "1.1"
```

Which of the following will return the floating point number `2.0` ?

In [65]:

```
# first + float(second)          # choice a
# float(second) + float(third)   # choice b
# first + int(third)             # choice c
# first + int(float(third))      # choice d
# int(first) + int(float(third)) # choice e
# 2.0 * second                   # choice f
```

Out[65]:

```
2.0
```

Follow along with materials and sessions at [https://deisdata.github.io (https://deisdata.github.io)](https://deisdata.github.io)