Download (right-click, save target as ...) this page as a jupyterlab notebook from:

Laboratory 4.1

Laboratory 4.1: Data Structures

Medrano, Giovanni

R11521018

ENGR 1330 Laboratory 4.1 - In-Lab

In [2]:
```
# Preamble script block to identify host, user, and kernel
import sys
! hostname
! whoami
print(sys.executable)
print(sys.version)
print(sys.version_info)
```

```
DESKTOP-6HAS1BN
desktop-6has1bn\medra
C:\Users\medra\anaconda3\python.exe
3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
sys.version_info(major=3, minor=8, micro=5, releaselevel='final', serial=0)
```

Data Structures: List

A list is a collection of data that are somehow related. It is a convenient way to refer to a collection of similar things by a single name, and using an index (like a subscript in math) to identify a particular item.

In engineering and data science we use lists a lot - we often call them vectors, arrays, matrices and such, but they are ultimately just lists.

To declare a list you can write the list name and assign it values. The square brackets are used to identify that the variable is a list. Like:

```
MyList = [7,11,5,9,13,66,99,223]
```

One can also declare a null list and use the `append()` method to fill it as needed.

```
MyOtherList = [ ]
```

Python indices start at ZERO. Alot of other lnguages start at ONE. Its just the convention.

The first element in a list has an index of 0, the second an index of 1, and so on. We access the contents of a list by referring to its name and index. For example

```
MyList[3] has a value of the number 9.
```

In [3]:
```
MyOtherList = [] #Create an empty list
print(MyOtherList)
MyOtherList.append(765) #Add one item to the list
print(MyOtherList)

MyList = [7,11,5,9,13,66,99,223] #Define a list
print(MyList)

sublist = MyList[3:6] #slice a sublist
print("sublist is: ", sublist)

mysum = sum(sublist) #sum the numbers in the sublist
print("Sum: ", mysum)

mylength = len(sublist) #get the length of the sublist
print("Length: ", mylength)
```

```
[]
[765]
[7, 11, 5, 9, 13, 66, 99, 223]
sublist is:  [9, 13, 66]
Sum:  88
Length:  3
```

Data Structures: Special List | Tuple

A tuple is a special kind of list where the values cannot be changed after the list is created. It is useful for list-like things that are static - like days in a week, or months of a year. You declare a tuple like a list, except use round brackets instead of square brackets.

```
MyTupleName =
("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")
```

Data Structures: Special List | Dictionary

A dictionary is a special kind of list where the items are related data PAIRS. The second item could itself be a list, so a dictionary would be a meaningful way to build a database in Python.

To declare a dictionary using `curly` brackets

```
MyPetsNamesAndMass = { "Dusty":7.8 , "Aspen":6.3, "Merrimee":0.03}
```

To declare a dictionary using the `dict()` method

```
MyPetsNamesAndMassToo = dict(Dusty = 7.8 , Aspen = 6.3, Merrimee = 0.03)
```

Some examples follow:

In [4]:
```
MyTupleName = ("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")
print(MyTupleName)
```

```
('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec')
```

In [5]:
```
MyPetsNamesAndMass = { "Dusty":7.8 , "Aspen":6.3, "Merrimee":0.03}
print(MyPetsNamesAndMass)
MyPetsNamesAndMassToo = dict(Dusty = 7.8 , Aspen = 6.3, Merrimee = 0.04)
print(MyPetsNamesAndMassToo)
```

```
{'Dusty': 7.8, 'Aspen': 6.3, 'Merrimee': 0.03}
{'Dusty': 7.8, 'Aspen': 6.3, 'Merrimee': 0.04}
```

In [6]:
```
# Tuples
MyTupleName = ("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")
# Access a Tuple
print ("5th element of the tuple:", MyTupleName[4])
# Dictionary
MyPetsNamesAndMass = { "Dusty":7.8 , "Aspen":6.3, "Merrimee":0.03}
# Access the Dictionary
print ("Aspen's mass = ", MyPetsNamesAndMass["Aspen"])
# Change a value in a dictionary
print ("Merrimee's mass" , MyPetsNamesAndMass["Merrimee"])
MyPetsNamesAndMass["Merrimee"] = 0.01
print ("Merrimee's mass" , MyPetsNamesAndMass["Merrimee"], "She lost weight !")
# Alternate dictionary
MyPetsNamesAndMassToo = dict(Dusty = 7.8 , Aspen = 6.3, Merrimee = 0.03)
print ("Merrimee's mass" , MyPetsNamesAndMassToo["Merrimee"])
# Attempt to change a Tuple
#MyTupleName[3]=("Fred") # Activate this line and see what happens!
```

```
5th element of the tuple: May
Aspen's mass =  6.3
Merrimee's mass 0.03
Merrimee's mass 0.01 She lost weight !
Merrimee's mass 0.03
```

Example: Nested Dictionary

From the dictionary below, print "Pandemic" and "Tokyo":

In [7]:
```
FD = {"Quentin":"Tarantino","2020":[2020,"COVID",19,"Pandemic"],"Bond":["James","Gun",(
print(FD)
```

```
{'Quentin': 'Tarantino', '2020': [2020, 'COVID', 19, 'Pandemic'], 'Bond': ['James', 'Gu
n', ('Paris', 'Tokyo', 'London')]}
```

In [8]:
```
FD['2020'][3]
```

Out[8]:
```
'Pandemic'
```

In [9]:
```
FD['Bond'][2][1]
```

```
Out[9]: 'Tokyo'
```

```
In [10]: FD['Bond']
         FD['Bond'][2]
         FD['Bond'][2][0]
```

```
Out[10]: 'Paris'
```

Readings

Here are some great reads on this topic:

- **"Common Python Data Structures (Guide)"** by **Dan Bader** available at *https://realpython.com/python-data-structures/
- **"Data Structures You Need To Learn In Python"** by **Akash** available at *https://www.edureka.co/blog/data-structures-in-python/
- **"Data Structures in Python— A Brief Introduction"** by **Sowmya Krishnan** available at *https://towardsdatascience.com/data-structures-in-python-a-brief-introduction-b4135d7a9b7d
- **"Everything you Should Know About Data Structures in Python"** by **ANIRUDDHA BHANDARI** available at *https://www.analyticsvidhya.com/blog/2020/06/data-structures-python/
- **"Conditional Statements in Python"** by **John Sturtz** available at *https://realpython.com/python-conditional-statements/
- **"Python If Statement explained with examples"** by **CHAITANYA SINGH** available at *https://beginnersbook.com/2018/01/python-if-statement-example/

Here are some great videos on these topics:

- **"Python: Data Structures - Lists, Tuples, Sets & Dictionaries tutorial"** by **Joe James** available at *https://www.youtube.com/watch?v=R-HLU9Fl5ug&t=92s
- **"Python Tutorial for Beginners 5: Dictionaries - Working with Key-Value Pairs"** by **Corey Schafer** available at *https://www.youtube.com/watch?v=daefaLgNkw0
- **"How to Use If Else Statements in Python (Python Tutorial #2)"** by **CS Dojo** available at *https://www.youtube.com/watch?v=AWek49wXGzI
- **"Python If Statements | Python Tutorial #10"** by **Amigoscode** available at *https://www.youtube.com/watch?v=wKQRmXR3jhc