Lesson 00

CSC357 Machine Learning

13 January 2020

1. What happens when we run this Python code?

```
a = 3.14159265

type( a )

b = int( a )

type( b )

a == b

type( a == b )

c = "Donald Knuth"

type( c )

d = 'Edsger Dijkstra'

type( d )
```

2. What happens when we run this Python code?

```
a = [i for i in range(12)]

print( a )
```

3. What happens when we run this Python code?

```
b = [i for i in range(1948, 2020, 4)]

print( b )

print( b[0] )
```

```
print( b[:4])

print( b[5:])

print( b[6:8] )
```

4. What happens when we run this Python code?

```
c = [ int((i * (i + 1))/2) for i in range(8)]

print( c )
```

5. What happens when we run this Python code?

```
d = map( lambda x: x * x, [ 1, 2, 3, 4 ])

for n in d:
    print( n )
```

6. What happens when we run this Python code?

```
e = filter( lambda x: (x % 2) == 1, [0, 1, 2, 3, 4, 5, 6, 7] )

for n in e:
    print( n )
```

7. What happens when we run this Python code?

```
f = [ "simon" ]

print( f )

f.append( "garfunkel" )

print( f )

f[0] = "paul"
f[1] = "art"

print( f )
```

8. What happens when we run this Python code?

   Pay attention! Parentheses (not brackets) delimit the names in this example.

```
g = ( "john", "paul", "george", "ringo" )

print( len(g) )

# can you replace "john" with "lennon"?
```

9. What happens when we run this Python code?

   Pay attention! Brackets (not parentheses) delimit the numbers in this example.

```
h = [2, 3, 5, 7]
h.extend( [11, 13, 17, 19] )

print( h )
```

10. You might be tempted to write something like this:

```
a = [ 0, 1, 2, 3 ]
b = [ 4, 5, 6, 7 ]
c = a.extend( b )

print( c )
```

    - Did the interpreter identify an error?
    - Did you see what you expected to see?

11. What happens when we run this Python code?

```
g = { "mission": "apollo 8", "year": 1968 }

print( g )

g[ "commander" ] = "Borman"

print( g )
```

12. What happens when we run this Python code?

```
oak = { "genus": "Quercus", "species": "alba" }

for key in oak:
    print( key )

for key in oak:
    print( oak[key] )

for key in oak:
    print( key, oak[key] )
```

13. What happens when we run this Python code?

```
a = [ 1.61803, 2.71828, 3.14159 ]
b = ( 5, 2, 3, 1, 4 )
c = { "genus": "Quercus", "species": "alba" }

print( len(a) )
print( len(b) )
print( len(c) )
```

14. How do we slice a list or tuple? Look for examples in these exercises, then look on the Internet for a more complete explanation.

15. How do Python lists differ from Python tuples?

16. Can Python lists and tuples be heterogenous data structures?

17. Think of several other programming languages whose names are familiar to you.

    - Which of those languages also allow **lambda** expressions?
    - Which provide **filter** and **map**() functions?

18. Can we define lists of lists in Python? tuples of tuples? dictionaries that include values that are lists? What do you think? Can you begin to see in your mind's eye how we might represent data sets?

19. Find a definition of *tensor* on *mathworld.com*. You do not need to learn everything that you find in this article! The definition that we need is in the last sentence of the first paragraph.

20. We are going to build a model of housing prices. Think for a few minutes about how you might predict the prices of homes in a given neighborhood. What kinds of information about that neighborhood and the people and houses in that neighborhood might be helpful?

    Now, go to *Wikipedia*. Find the article titled *Categorical variable*. Can you suggest a categorical variable that we could build into our model of housing prices?