

Lecture 3: Python Intro

CS 167: Machine Learning

List Operations

Try out this Python code:

```
x = [1,2,3]
y = x
x[0] = "abc"
print(y)
z = x[1]
x[1] = "what"
print(x)
print(z)

x.append(70)
x.insert(2,90)
print(x)

print( x.index(90) )
print( len(x) )
print( x+x )
```

What happens if you try
something like
`x[10] = 'ok'`
for an index 10 that
doesn't exist?

Getting Started

Try out this Python code:

```
a = 10
b = 3.14
c = "machine learning"
d = 'hi'
e = 'x'
f = [1,2,3]
print(a,b,c,d,e,f)

aty = type(a)
bty = type(b)
cty = type(c)
dty = type(d)
ety = type(e)
fty = type(f)
print(aty,bty,cty,dty,ety,fty)
```

CS 167: Machine Learning

L03: Python Intro

2 / 9

Slicing

Try out this Python code:

```
my_list = [33, 55, 'hello', 'R2D2', 7.8, 'banana']
print( my_list[1:5] )
print( my_list[:4] )
print( my_list[4:] )
print( my_list[:] )

my_slice = my_list[1:5]
my_slice[2] = 'what'
print(my_slice)
print(my_list)

print(my_list[1:5] + 2*['Drake', 'University'])

my_list[:0] = ['Bulldogs']
print(my_list)
```

Lists, Tuples, and Dictionaries

```
#this all works
my_list = [42, 3.14, 'hello', ['ice', 'ice', 'baby']]
print(my_list[3][2])
print(my_list[2][2])
my_tuple = (100,200,'300')
print(my_tuple[1])
my_dictionary = {'x':42, 'y':3.14}
print(my_dictionary['x'])
Try these out one at a time
print(my_list[1][2])
my_tuple[1] = 0
print(my_dictionary[1])
```

if and for

```
x = 33
if x < 0:
    print("x is negative")
elif x % 2:
    print("x is positive and odd")
else:
    print("x is even and nonnegative")

numList = [3, 7, 9,17, 2]

for num in num_list:
    print(num)

print(range(10,20))
for n in range(10,20):
    print(n)
```

Exercise: Write a Python program that sums up all the numbers between 50 and 100

Split

Try this out

```
my_str = "I had breakfast with\
    my mother, the queen, and a talking giraffe."
first_split = my_str.split(" ")
second_split = my_str.split(",")
print(first_split)
print(second_split)

f = open("testFile.txt", "r")
line1 = f.readline()
print(line1)

for line in f:
    print(line)
```

Then try printing `line.rstrip()` instead.

Exercises

Exercise 0: Throw away the `titanic-exercise.csv` file we used last week. You will never use that version again. Then download `titanicFull.csv`.

Exercise 1: Read the `titanicFull.csv` file into a two dimensional list. When you print the list, it should look something like this:

```
[[ 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
 [ '1', '1', 'Dodge', 'Master. Washington', 'male', '4', '0', '2', '33638', '81.8583', 'A34', 'S'],
 [ '0', '3', 'Williams', 'Mr. Howard Hugh "Harry"', 'male', '', '0', '0', 'A/5 2466', '8.05', '', 'S'],
 [ '1', '1', 'Peuchen', 'Major. Arthur Godfrey', 'male', '52', '0', '0', '113786', '30.5', 'C104', 'S'],
 ...]
```

Exercise 2: Put the first row into a list named `header`

Exercise 3: Put the rest of the rows into a 2D list called `titanic_data`

More Exercises: Dealing with Missing Data

The titanic data is missing a lot of the passengers' ages. Many machine learning algorithms do not work (or work well) with missing data, so let's replace all the missing ages with the average age of the passengers that we do know about.

Exercise 4: Compute the average age of the passengers in `titanic_data`. Note that since these values are strings, you will need to convert them to floats in order to add them up.

```
float("23.5") #returns the float value 23.5
```

Exercise 5: Replace any missing ages in `titanic_data` with the average age.

Exercise 6: The Embarked column gives the port that the passengers embarked from (C = Cherbourg, Q = Queenstown, S = Southampton). There are missing values here too, so fix them.