**Functions**

1. **Python Basics**
2. Square root function

import math

root = math.sqrt(99)

1. Floor Function to round off to integer

import math

flr = math.floor(89.9)

1. Variables Within Modules

import math

print(math.pi)

a = math.sqrt(math.pi)

b = math.ceil(math.pi)

c = math.floor(math.pi)

1. Csv reader function

import csv

f = open("nfl.csv", 'r')

csvreader = csv.reader(f)

nfl = list(csvreader)

1. Classes and objects

class Dataset:

def \_\_init\_\_(self):

self.type = "csv"

class Dataset:

def \_\_init\_\_(self, data):

self.data = data

f = open("nfl.csv", 'r')

csvreader = csv.reader(f)

nfl\_data = list(csvreader)

nfl\_dataset = Dataset(nfl\_data)

dataset\_data = nfl\_dataset.data

1. Enumerate function in modules

class Dataset:

def \_\_init\_\_(self, data):

self.header = data[0]

self.data = data[1:]

def column(self, label):

if label not in self.header:

return None

index = 0

for idx, element in enumerate(self.header):

if label == element:

index = idx

column = []

for row in self.data:

column.append(row[index])

return column

nfl\_dataset = Dataset(nfl\_data)

year\_column = nfl\_dataset.column('year')

player\_column = nfl\_dataset.column('player')

1. Set() function for unique elements

unique\_animals = set(["Dog", "Cat", "Hippo", "Dog", "Cat", "Dog", "Dog", "Cat"])

print(unique\_animals)

1. Try and except block

try:

int('')

except Exception:

print("There was an error")

try:

int('')

except Exception:

pass

1. Add a column in dataset

for row in legislators:

y = row[2].split('-')[0]

try:

birth\_year = int(y)

except Exception:

birth\_year = 0

row.append(birth\_year)

print(legislators[0:10])

1. Filling missing values in column with last value

last\_value = 1

for row in legislators:

if row[7] == 0:

row[7] = last\_value

last\_value = row[7]

1. Enumerate function to use multiple list in tandem

animals = ["Dog", "Tiger", "SuperLion", "Cow", "Panda"]

viciousness = [1, 5, 10, 10, 1]

for i, animal in enumerate(animals):

print("Animal")

print(animal)

print("Viciousness")

print(viciousness[i])

things = [["apple", "monkey"], ["orange", "dog"], ["banana", "cat"]]

trees = ["cedar", "maple", "fig"]

for i, thing in enumerate(things):

thing.append(trees[i])

1. Short for loops

apple\_prices = [100, 101, 102, 105]

apple\_prices\_doubled = [apple\*2 for apple in apple\_prices]

apple\_prices\_lowered = [apple-100 for apple in apple\_prices]

1. Highest value with None

values = [-50, -80, -100]

max\_value = None

for i in values:

if max\_value is None or i > max\_value:

max\_value = i

max\_value = None

for k in name\_counts:

count = name\_counts[k]

if max\_value is None or count > max\_value:

max\_value = count

1. Items() function for finding highest items

highest\_male\_count = None

for name in male\_name\_counts:

count = male\_name\_counts[name]

if highest\_male\_count is None or count > highest\_male\_count:

highest\_male\_count = count

top\_male\_names = []

for male, count in male\_name\_counts.items():

if count == highest\_male\_count:

top\_male\_names.append(male)