**Functions**

1. **Python Basics**
2. Finding the variable data type using "type" function

print(type(china\_exact))

1. Converting data types in python

int\_to\_str = str(china\_rounded)

str\_to\_int = int(int\_to\_str)

1. To add values to a list object, use the list.append() **method**

cities= []

cities.append("Albuquerque")

1. Available list.method in python

<https://docs.python.org/3/tutorial/datastructures.html#more-on-lists>

1. Opening and Reading files

A = open("story.txt", "r")

G = A.read()

1. The split() method

sample = "john,plastic,joe"

split\_list = sample.split(",")

print(split\_list)

# Here's another example.

string\_two = "How much wood\ncan a woodchuck chuck\nif a woodchuck\ncould chuck wood?"

split\_string\_two = string\_two.split('\n')

print(split\_string\_two)

1. Loops

For loops

ten\_rows = rows[0:10]

for rate in ten\_rows:

print(rate)

1. List of Lists

three\_rows = ["Albuquerque,749", "Anaheim,371", "Anchorage,828"]

final\_list = []

for row in three\_rows:

split\_list = row.split(',')

final\_list.append(split\_list)

print(final\_list)

1. Accessing elements in Lists of List

# Returns the first list's first element, 'Albuquerque'.

first\_list\_first\_value = final\_data[0][0]

1. Counter and Index

counter = 0

index = 0

for city in cities:

if city == "Washington":

index = counter

counter += 1

1. Highest value in List

highest = crime\_rates[0]

for cr in crime\_rates:

if cr > highest:

highest = cr

print(highest)

1. Convert numerical values

print(nested\_list[0:5])

numerical\_list = []

for cr in nested\_list:

f = cr[0]

h = float(cr[1])

i = [f, h]

numerical\_list.append(i)

print(numerical\_list[0:5])

1. IN statement

cat\_found = "cat" in animals

space\_monster\_found = "space\_monster" in animals

1. Counting with dictionaries

pantry = ["apple", "orange", "grape", "apple", "orange", "apple", "tomato", "potato", "grape"]

pantry\_counts = {}

for item in pantry:

if item in pantry\_counts:

pantry\_counts[item] = pantry\_counts[item] + 1

else:

pantry\_counts[item] = 1

weather\_counts = {}

for item in weather:

if item in weather\_counts:

weather\_counts[item] = weather\_counts[item] + 1

else:

weather\_counts[item] = 1

1. Functions

def return\_list(lista):

epmt\_list = []

for each in lista:

first = each[0]

epmt\_list.append(first)

return epmt\_list

movie\_names = return\_list(movie\_data)

print(movie\_names[0:5])

with IF statement

wonder\_woman = ['Wonder Woman','Patty Jenkins','Color',141,'Gal Gadot','English','USA',2017]

def is\_usa(input\_lst):

if input\_lst[6] == "USA":

return True

else:

return False

wonder\_woman\_usa = is\_usa(wonder\_woman)

with multiple arguments

def index\_equals\_str(input\_lst,index,input\_str):

if input\_lst[index] == input\_str:

return True

else:

return False

With optional parameter

def counter(input\_lst,header\_row = False):

num\_elt = 0

if header\_row == True:

input\_lst = input\_lst[1:len(input\_lst)]

for each in input\_lst:

num\_elt = num\_elt + 1

return num\_elt

wonder\_woman\_in\_color = index\_equals\_str(wonder\_woman,2,"Color")

print(wonder\_woman\_in\_color)

1. A

def clean\_text(text\_string, special\_characters):

cleaned\_string = text\_string

for string in special\_characters:

cleaned\_string = cleaned\_string.replace(string, "")

cleaned\_string = cleaned\_string.lower()

return(cleaned\_string)

def tokenize(text\_string, special\_characters, clean=False):

cleaned\_text = text\_string

if clean:

cleaned\_text = clean\_text(text\_string, special\_characters)

tokens = cleaned\_text.split(" ")

return(tokens)

final\_misspelled\_words = []

def spell\_check(vocabulary\_file, text\_file, special\_characters=[",",".","'",";","\n"]):

misspelled\_words = []

vocabulary = open(vocabulary\_file).read()

text = open(text\_file).read()

tokenized\_vocabulary = tokenize(vocabulary, special\_characters)

tokenized\_text = tokenize(text, special\_characters, True)

for ts in tokenized\_text:

if ts not in tokenized\_vocabulary and ts != '':

misspelled\_words.append(ts)

return(misspelled\_words)

final\_misspelled\_words = spell\_check(vocabulary\_file="dictionary.txt", text\_file="story.txt")

print(final\_misspelled\_words)