Yokesh .J

Exercises

# Answer the questions or complete the tasks outlined in bold below, use the speciﬁc method described if applicable.

\*\* What is 7 to the power of 4?\*\*

import math math.pow(7,4)

**2401.0**

# \*\* Split this string:\*\*

s = "Hi there Sam!"

## \*into a list. \*

s = "Hi there Sam!"

s.split()

**['Hi', 'there', 'Sam!']**

# \*\* Given the variables:\*\*

planet = "Earth" diameter = 12742

# \*\* Use .format() to print the following string: \*\*

The diameter of Earth is 12742 kilometers.

planet = "Earth" diameter = 12742

print("The diameter of {} is {} kilometers.".format(planet,diameter))

**The diameter of Earth is 12742 kilometers.**

# \*\* Given this nested list, use indexing to grab the word "hello" \*\*

lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]

lst[3][1][2][0]

**'hello'**

# \*\* Given this nest dictionary grab the word "hello". Be prepared, this will be annoying/tricky \*\*

d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}

d['k1'][3]['tricky'][3]['target'][3]

**'hello'**

# \*\* What is the main difference between a tuple and a list? \*\*

#The list is dynamic, whereas the tuple has static characteristics. #lists are mutable,whereas the tuples are immutable

# \*\* Create a function that grabs the email website domain from a string in the form: \*\*

[user@domain.com](mailto:user@domain.com)

**So for example, passing "**[**user@domain.com**](mailto:user@domain.com)**" would return: domain.com**

def domainGet(strDomain):

return strDomain.split('@')[1]

out = domainGet('user@domain.com')

print(out)

**domain.com**

domainGet('user@domain.com')

**'domain.com'**

# \*\* Create a basic function that returns True if the word 'dog' is contained in the input string. Don't worry about edge cases like a punctuation being attached to the word dog, but do account for

capitalization. \*\*

def findDog(st): print(st.lower())

return 'dog' in st.split()

findDog('is there a dog here?')

**is there a dog here? True**

# \*\* Create a function that counts the number of times the word "dog" occurs in a string. Again ignore edge cases. \*\*

def countDog(st): st.lower().split() return st.count('dog')

countDog('This dog runs faster than the other dog dude!')

**2**

Problem

\**You are driving a little too fast, and a police oficer stops you. Write a function to return one of 3*

*possible results: "No ticket", "Small ticket", or "Big Ticket". If your speed is 60 or less, the result is "No Ticket". If speed is between 61 and 80 inclusive, the result is "Small Ticket". If speed is 81 or more,*

## the result is "Big Ticket". Unless it is your birthday (encoded as a boolean value in the parameters of the function) -- on your birthday, your speed can be 5 higher in all cases. \*

def caught\_speeding(speed, is\_birthday):

if is\_birthday:

speeding = speed - 5 else:

speeding = speed

if speeding > 80: return 'Big Ticket'

elif speeding > 60: return 'Small Ticket'

else:

return 'No Ticket'

caught\_speeding(81,False)

**'Big Ticket'**

caught\_speeding(81,True)

**'Small Ticket'**

# Create an employee list with basic salary values(at least 5 values for 5 employees) and using a for loop retreive each employee salary and calculate total salary expenditure.

e=[1000,2000,3000,4000,5000]

tot=0

for i in range(0,len(e)): name=eval(input("Enter Empname:")) da=e[i]\*(15/100)

hra=e[i]\*(10/100) ta=e[i]\*(15/100) pf=e[i]\*(5/100) it=e[i]\*(5/100) gs=e[i]+da+hra+ta ded=pf+it

ns=gs-ded print(ns) tot=tot+ns

print("TOTAL:",tot)

**Enter Empname:"vj" 1300.0**

**Enter Empname:"vm" 2600.0**

**Enter Empname:"vn"**

**3900.0**

**Enter Empname:"vv" 5200.0**

**Enter Empname:"vvn" 6500.0**

**TOTAL: 19500.0**

# Create two dictionaries in Python:

First one to contain ﬁelds as Empid, Empname, Basicpay Second dictionary to contain ﬁelds as DeptName, DeptId.

# Combine both dictionaries.

d1={

"Empid":211, "Empname": "vijay", "Basicpay": 20000

} d2={

"DeptName":"CSE", "Deptid":211

}

d3= d1.copy() d3.update(d2) print(d3)

**{'Empid': 211, 'Empname': 'vijay', 'Basicpay': 20000, 'DeptName': 'CSE', 'Depti**

[Colab paid products](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links) - [Cancel contracts here](https://colab.research.google.com/cancel-subscription)