Function= reusable code

Create a function in python

Def= define, name has to be related to the purpose of the function

Def greeting ():

Indentation is very important in python (use 4 spaces or a tab, but choose one and stick with it)

2 types of variables (global and local)

name= local variable (part of the function)

e = “!” 🡪 global variable (out of the scope of the function)

ground rule is consistency- have to choose the way you do it and stick w it

lists are mutable (mutable= you can change the list)

data type= classification of the data, so the computer knows who we attempt to use the data in the future

if you want to create a stream you have to use quotation marks

doesn’t matter how many spaces you have between two lines

you have to put the rules in the list as you want them to be followed (in the right order)

“if” and “else” are conditions

elif = if else (more layers)

“:” at the end of the line mean “Then”

two == because we already used one = earlier so now we need = to mean something diff.

means “equal”

errors are good in python, very information, read them carefully

can’t convert two different data types, so we have to take one and convert it into another (take integers and convert it into stream and vice versa)

TAX & TIP CODE

1st do a pseudo code

step by step instruction

create a new function- def name of function (

take input from user & store it 🡪 variable= cost of meal

int(input(pls\_enter\_cost\_of\_meal\_)

then we need 2 more variables (tax rate & tip rate) & multiply to find out what needs to be paid

*def* tax\_and\_tip():

tax\_rate = 0.05

tip\_rate = 0.18

cost\_of\_meal = *int*(input(“Pls enter the bost:”))

tip = cost\_of\_meal \* tip\_rate

tax = cost\_of\_meal \* tax\_rate

total = cost\_of\_meal + tip + tax

print(“Tip is ${}, Tax is ${}, and Total is ${}”.format(tip,tax,total)

range function – 1st has a start point and 2nd is a stop point (101 make it stop at 100))

for number in range (1, 101)

print (number) 🡪 shoots out all the numbers

<http://zingpython.com/>

fizzbuzz test

create list with square brackets

[ ]

list = [1,2,3,4,5,6,9]

banana

“keep your code dry” = don’t repeat yourself

counter🡪 take a variable and assign a value (i.e. 0)

counter = counter + 1

^same code as count += 1

1.5 hour classes