**Strings & Printing**

*print(“Hello World”)*

*phrase = “Hello world” phrase.isupper()*

*print(phrase + “hello”) len(phrase)*

*phrase.lower()*

*phrase.upper() phrase[0]*

*phrase.index(“w”) 🡪 6*

*phrase.replace(“World”, “Anyone”)*

*print(str(5) + “ is my favorite number”)*

**Variables & Data Types**

*character\_name = “John”*

*character\_age = 35*

*print(“A man named” + character\_name + “,”)*

*print(“He was” + character\_age)*

**Numbers**

*num = -5*

*abs(num) 🡪 5 from math import \**

*max, min, pow, round, floor, ceil, sqrt*

**Input**

*name = input(“Enter your name: ”)*

*print(“Hello ” + name + “!”)*

**List**

*friends = [“Kevin”, 2, False]*

*friends = [“Kevin”, “Karen”, “Jim”]*

*print(friends);*

*print(friends[-1]) 🡪 Jim*

*print(friends[1:]) 🡪 [‘Karen’, ‘ Jim’]*

*friends = [“Kevin”, “Karen”, “Jim”, “Oscar”, “Toby”]*

*print(friends[1:3]) 🡪 [‘Karen’, ‘ Jim’]*

*friends = [“Kevin”, “Karen”, “Jim”, “Oscar”, “Toby”]*

*lucky\_numbers = [4,8,15,16,23,42]*

*friends.extend(lucky\_numbers) 🡪 Combines them*

*friends.extend(Creed) 🡪 Appends Creed*

*friends.insert(1, “Kelly”) 🡪 would add Kelly before Karen*

*friends.remove(“Jim”) 🡪 Removes Jim*

*friends.clear() 🡪 Clears list []*

*friends.pop() 🡪 Removes the last element from the list*

*friends.index(“Kevin”) 🡪 Tells the index of Kevin*

*friends.index(‘Mike”) 🡪 Error, mike not in the list*

*friends.count(“Jim”) 🡪 Says how many times Jim appears in the list*

*friends.sort() 🡪 Sorts the list in Accending order*

*friends2 = friends.copy 🡪 copies friends without ref*

**Tuples**

*coordinates = (4, 5)*

*coordinates[1] = 10 🡪 error, tuples are inmutable*

*print(coordinates[1])*

*coordinates = [(4, 5), (6, 7), (80, 34)] #List of Tuples*

**Functions/Def**

*def say\_hi(name, age):*

*print(“Hello ” + name + “, you are” + age )*

*say\_hi(“Mike”, “35”)*

**Return Statement** **Python Quick Review**

*def cube(num):*

*return num\*num\*num*

*cube(4) 🡪 64*

**If Statements**

*is\_male = True*

*is\_tall = False*

*if is\_male or is\_tall:*

*print(“You are a male or tall or both”)*

*elif is\_male and not(is\_tall):*

*print(“You are a short male”)*

*elif not(is\_male) and is\_tall:*

*print(“You are not a male but you are tall”)*

*else:*

*print(“You are not a male”)*

**Dictionaries**

*monthConversions = {*

*“Jan” : “January”, # left side doesn’t need to be a spring*

*“Feb” : “February”,*

*“Mar” : “March”,*

*“Apr” : “April”,*

*“May” : “May”,*

*“Jun” : “June”,*

*“Jul” : “July”,*

*“Aug” : “August”,*

*“Sep” : “September”,*

*“Oct” : “October”,*

*“Nov” : “November”,*

*“Dec” : “December”,*

*}*

*print(monthConversions[“Mar”])*

*print(monthConversions.get(“Mar”))*

*print(monthConversions.get(“Lov”, “Not a valid key”)) 🡪 Default*

**While Loop**

*i = 1;*

*while i <= 10:*

*print(i)*

*i += 1;*

**For loop**

*for letter in “Hello World”:*

*print(letter)*

*friends = [“Jim”, “Karen”, “Kevin”]*

*for friend in friends:*

*print(friend)*

*for index in range(10): #print out 0-10, not including 10*

*print(index)*

*for index in range(3,10) #print out 3-9*

*print(index)*

*for index in range(len(friends)):*

*print(friends[index])*

**2D Lists and Nested Loops**

*number\_grid = [*

*[1, 2, 3]*

*[4, 5, 6],*

*[7, 8, 9],*

*[0]*

*]*

*print(number\_grid[2][1]) 🡪 8*

*for row in number\_grid:*

*print(row)*

*for row in number\_grid:*

*for col in row:*

*print(col)*

**Translator Function Example**

*def translate(phrase):*

*translation = “”*

*for letter in phrase:*

*if letter.lower() in “aeiou”*

*if letter.isupper()*

*translation = translation + “G”*

*else:*

*translation = translation + “g”*

*else*

*translation = translation + letter*

*return translation*

**Comments**

*#Start comments like this*

*‘’’*

*Three quotes will be multiple line comments*

*‘’’*

**Try/Except**

*try:*

*value = 10/0;*

*number = int(input(“Enter a number: ”))*

*print(number)*

*except ZeroDivisionErro as errr:*

*print(err)*

*except ValueError:*

*print(“Invalid Input”)*

*except:*

*print(“Invalid Input”)*

**Reading Files**

*employee\_file = open(“employees.txt”, “r”) #r,w,a, r+*

*print(employee\_file.readable()) 🡪 True*

*print(employee\_file.readline()) 🡪 Reads the first line*

*print(employee\_file.readline()) 🡪 Reads the 2nd line*

*print(employee\_file.readlines()) 🡪 Puts each line as a list*

*print(employee\_file.readlines()[1]) 🡪 Reads 2nd line*

*for employee in employee\_file.readlines():*

*print(employee)*

*employee\_file.close()*

**Writing to Files**

*employee\_file = open(“employees.txt”, “a”)*

*employee\_file.write(“\nToby – Human Resources”);*

*employee\_file.close()*

*employee\_file = open(“employees.txt”, “w”)*

*employee\_file.write(“\nToby – Human Resources”);*

*employee\_file.close()*

* *Will overwrite the file*

*employee\_file = open(“employees1.txt”, “w”)*

* *Will create a new file*

**Classes & Objects**

*File name: Student.py*

*class Student:*

*def \_init\_(self, name, major, gpa, is\_on\_probation):*

*self.name = name*

*self.major = major*

*self.gpa = gpa*

*self.is\_on\_probation = is\_on\_probation*

*def on\_honor\_roll(self):*

*if self.gpa >= 3.5:*

*return True*

*else*

*return False*

*File name: app.py*

*from Student import Student #importanting student class from file*

*student1 = Student(“Jim”, “Business”, 3.1, False);*

*print(student1.name)*

*print(student1.on\_honor\_roll())*

**Inheritance**

*File name: Chef.py*

*class chef:*

*def make\_chicken(self):*

*Print(“Chef makes chicken”)*

*def make\_salad(self):*

*Print(“Chef makes salad”)*

*def make\_special\_dish(self):*

*Print(“Chef makes bbq ribs”)*

*File name: ChineseChef.py*

*from Chef import Chef*

*class ChineseChef(Chef):*

*def make\_special\_dish(self):*

*Print(“Chef makes orange chicken”)*

*def make\_fried\_rice(self):*

*Print(“Chef makes fried rice”)*

*File name: app.py*

*from Chef import Chef*

*from ChineseChef import ChineseChef*

*myChef = Chef()*

*myChef.make\_chicken()*

*myChineseChef = ChineseChef()*

*myChineseChef.make\_chicken()*