**Introduction**

Today in studio, you will complete several tasks that will help you to learn how Python3 works. You should be prepared to engage with your classmates for the entire studio time. The main goals for today are:

1. To practice learning about different built-in Python functions and functions in NumPy.
2. Practice writing your own functions.

My expectations for you today are:

1. Stay engaged with the material for the entire time (take a quick stretch break as needed).
2. Focus more of what you will learn through this process rather than on finishing in time.
3. Remember the things you learned about teamwork, communication and listening from ENGR 100 and 102 in Fall and Winter.
4. Be respectful of your classmates’ input and strive to be part of an inclusive culture in this learning space.

The TAs and ULAs will guide you through today’s work. Please follow along with the

instructions below and fill out the bolded sections in this document. You will also need to access other documents as indicated. Feel free to ask lots of questions if anything is unclear. An evaluation rubric for today’s studio is included below.

You will show your completed document to a TA/ULA at the end of the studio session.

Thank you for your hard work and engagement. Be sure you tell a TA/ULA that you are ready to leave in order to receive credit for today’s studio.

**Evaluation**

**Topic Description Points**

|  |  |  |
| --- | --- | --- |
| **Topic** | **Description** | **Points** |
| **Part 1:** | Exploring built-in Python functions and NumPy functions. | **3 pts** |
| **Part 2:** | Converting your old code into functions. And running your function by calling it with proper inputs and outputs. | **7 pts** |

**Part 1: (30 min)**

1. Explore the Python3 documentation website and fill in the table below for the following functions. What does the function do, what are the function inputs needed and what are the function outputs needed? Also note any functions you could have used to make your programming life so far easier and how?
   1. <https://docs.python.org/3/library/functions.html>

|  |  |  |  |
| --- | --- | --- | --- |
| Function name: | What does it do? | Needed input | Expected output |
| int() | Return an integer object constructed from a number or string x, or return 0 if no arguments are given | A string or a number x | An integer object |
| sum() | Sums start and the items of an iterable from left to right and returns the total. | Numbers and a start value that cannot be string | Numbers (sums of numbers being inputted) |
| len() | Return the length (the number of items) of an object | string, bytes, tuple, list, or range or a dictionary, set, or frozen set | Integer |
| max() | Return the largest item in an iterable or the largest of two or more arguments. | More than one items or iterable | Biggest number |
| range() | an immutable sequence type | Number for start or end | Sequence of number |
| bool() | Return a Boolean value | X: statement that will go through process of truth table | True or false |
| complex() | Return a complex number: | real+imag\*1j, or string, or number | Complex number |
| min() | Return the smallest item in an iterable or the smallest of two or more arguments. | Iterbale or more than one items | Minimum number of the iterable |
| round() | Return number rounded to ndigits precision after the decimal point. | A number and another number for the decimal place | A number rounded to the number of decimal place of input |
| float() | Return a floating point number constructed from a number or string x. | a number or string x | floating point of the number of string with number in it |
| input() | If the prompt argument is present, it is written to standard output without a trailing newline | A prompt by the writer, and input by the user | Input of the user |

1. Explore the NumPy documentation website and fill in the table below for the following functions. What does the function do, what are the function inputs needed and what are the function outputs needed? Also note any functions you could have used to make your programming life so far easier and how?
   1. <https://numpy.org/doc/stable/reference/index.html>

|  |  |  |  |
| --- | --- | --- | --- |
| Function name: | What does it do? | Needed input | Expected output |
| numpy.sqrt() | Return the non-negative square-root of an array, element-wise. | Array of non-negative numbers | Array of square-rooted numbers |
| numpy.log() | Natural logarithm, element-wise. | array of numbers | Array of natural logged numbers |
| numpy.log10() | Return the base 10 logarithm of the input array, element-wise. | Array of numbers | base 10 logarithm of the input array |
| numpy.cos() | Cosine element-wise. | Array of numbers | Array of cosine of input values |
| numpy.random() | Random values in a given shape. | Set of two numbers for dimension of random numbers | Random numbers according to the input dimension |
| numpy.mean() | Returns the average of the array elements | Array of numbers | mean of the array |
| numpy.average() | Returns the average of the array elements | Array of numbers | Average of the array |
| numpy.trapz() | Integrate along the given axis using the composite trapezoidal rule | X,Y array like  Dx: scalar  Axis: int | float or nd array |
| numpy.isclose() | Returns a boolean array where two arrays are element-wise equal within a tolerance. | a, b:array\_like  rtol:float  atoll:float  equal\_nan:bool | a boolean array |
| numpy.linspace() | Return evenly spaced numbers over a specified interval. | Start array  Stop array  Num: int  Endpt: Boolean  Retstep: Boolean  dtype | Nd array  Step: float |
| numpy.unique() | find the unique elements of an array. | array | unique elements of an array |

**Part 2: (80 min)**

You are going to turn some of your past HW and Studio code into functions and then call the functions to perform the computation from those assignments.

The format for writing a function is as follows:

def function\_name(input1, input2, …):

your code

…

…

return output1, output2, …

The format for calling function and storing its outputs as variables is as follows:

output1, output2, … = function\_name(input1, intput2, …)

**Part A:** From HW1 – write a function that outputs the maximum and the actual power from a wind turbine given its blade dimensions, its efficiency, and the wind speed.

Recall: To compute the maximum and actual power output for your wind turbine, you need the following math formulas:

* , which represents the cross-sectional area of a circle (π = 3.14159265).
* Pmax = 0.5ρAv3, which calculates the maximum available power given the wind speed in m/s (ν), cross-sectional area of the blades in m2 (A), and the density of the air (ρ = 1.2 kg/m3).
* Once you have the maximum available power, computing the actual amount of power (not the maximum) produced by the wind turbine is a matter of determining the amount of power based on the operating efficiency.

Test your function with the following:

|  |  |  |
| --- | --- | --- |
| Inputs | Max Power | Actual Power |
| Wind speed = 0.5 m/s  Radius = 2 m  Efficiency: 98% | 0.94 W | 0.92 W |
| Wind speed = 15 m/s  Radius = 2 m  Efficiency = 98% | 25447 W | 24938 W |
| Wind speed = 0.5 m/s  Radius = 3 m  Efficiency: 60% | 2.12 W | 1.27 W |
| Wind speed = 15 m/s  Radius = 3 m  Efficiency = 69% | 57255 W | 34353 W |

**Part B:** From Studio 5 – write a function that calculate the necessary orifice diameter given a pond’s depth and design flow discharge rate condition so that the pond does not overflow. You do not need to use a while loop, you can solve for D directly for your function computation.

Recall: The equations for estimating the pipe diameter (D) are as follows.

is the area of the drain outlet (called and orifice) at the bottom of the pond, is the diameter of the pipe. is the orifice discharge coefficient given as 0.98 for a rounded orifice, is the acceleration due to gravity = 9.81 m/s2, Qout is the design discharge rate out of the pond and h is the vertical storage capacity of the pond.

Check your program:

|  |  |
| --- | --- |
| Inputs | Output: needed orifice diameter (D) |
| h = 2 m  Qout = 2 m3/s | 0.64 m |
| h = 1 m  Qout = 2 m3/s | 0.77 m |
| h = 2 m  Qout = 0.5 m3/s | 0.32 m |
| h = 1 m  Qout = 0.5 m3/s | 0.38 m |

**Part C:** From HW 2 – write a function that outputs the change in power generation for any power type given as an input to the function between any given input of years between 2020 and 2021.

For example: percent\_change\_power = change\_worldpower(year1, year2, power\_type, csv file)

\*\*Note you can actually pass your CSV file to your function, read the CSV in your function pass the output back to your main program.

Recall: The worldpower dataset is linked on Canvas (same one you used previously)

The change in power from year1 to year2 in %

= ((power\_year1 – power\_year2) / (power\_year1)) \* 100

Check your change\_worldpower function:

Between years 2000 and 2021 the Hydro power generation changed by 60.0 %.

Between years 2010 and 2021 the Coal power generation changed by 20.0 %.

Between years 2010 and 2021 the Gasoline power generation changed by 31.0 %.

Between years 2010 and 2021 the Wind power generation changed by 423.0