Assignment 1 – Python

Please add any code files to this document and any code you wrote for optional problems).

Exercise – Variable Names

The Python interpreter has strict rules for variable names. Which of the following are legal Python names? If the name is not legal, state the reason.

1. and – not legal because it is a python reserved word
2. and - not legal same as before because it is a python reserved word
3. var - legal

1. var1 - legal

1. 1var – not legal because a var name must begin with a letter

1. my-name – not legal because it uses a math operator (subtraction)

1. your name – not legal because of the space in between your and name

1. COLOR - legal

Exercise – Types

It is important that we know the type of the values stored in a variable so that we can use the correct operators (as we have already seen!). Python automatically infers the type from the value you assign to the variable. Write down the type of the values stored in each of the variables below. Pay special attention to punctuation: values are not always the type they seem!

1. a = False – Boolean (bool)
2. b = 3.7 - float

1. c = ’Alex’ – string (str)

1. d=7 – integer (int)
2. e = ’True’ – string (str)

1. f = 17 – integer (int)

1. g = ’17’ – string (str)

1. h = True – Boolean (bool)

1. i = ’3.14159’ – string (str)

To To verify your answers, you can use the interactive Python shell, but first try to do the exercise without help.

>>> x = 100 >>> type(x) <type ’int’>

>>>

Exercise – Boolean operators

Boolean operators can seem tricky at first, and it takes practice to evaluate them correctly. Write the value (True or False) produced by each expression below, using the assigned values of the variables a, b, and c. Try to do this without using your interpreter, but you should check yourself when you think you’ve got it. Hint: Work from the inside out, starting with the inner-most expressions, like in arithmetic.

a = False

b = True

c = False

1. b and c – True and False = False
2. b or c – True or False = True

1. not a and b – not False and True = True and True = True

1. (a and b) or not c – (False and True) or True = False or True = True
2. not b and not (a or c) – not True and not (False or False) = False and not False = False and True = False

Exercise – String Operations

String operators might be a little less intuitive than those on numbers. This exercise will give you a chance to practice those. Given the following variables:

look = ’Look at me!’

now = ’ NOW’

What are the values of the following expressions? Try to guess on your own before using your interpreter (but feel free to use your interpreter once you get stuck).

1. look[:4] – ‘Look’
2. look[-1] – ‘!’
3. look\*2 – ‘Look at me!Look at me!’
4. look[:-1] + now + look[-1] – ‘Look at me NOW!’
5. now[1] – ‘N’
6. now[4] – string index out of range
7. look\*2 + look[:-1] + now + look[-1] - 'Look at me!Look at me!Look at me NOW!'

Exercise – List Operations

For the following, write the line(s) of code that will emit the given Output. For each problem there may be more than one correct answer; just give one.

1. >>> a\_list = [3, 5, 6, 12]

>>> min(a\_list)

3

2. >>> a\_list = [3, 5, 6, 12]

>>> max(a\_list)

12

3. >>> a\_list = [3, 5, 6, 12]

>>> a\_list.remove(3)

[5, 6, 12]

4. >>> a\_list = [3, 5, 6, 12]

>>> print(\*a\_list, sep= "\n")

3

5

6

12

5. >>> a\_list = [3, 5, 6, 12]

>>> a\_list.reverse()

[12, 6, 5, 3]

6. >>> a\_list = [3, 5, 6, 12]

>>> [x \* 3 for x in [3, 5, 6, 12]]

[9, 15, 18, 36]

7. >>> a\_list = [3, 5, 6, 12]

>>> a = bool(a\_list[0])

>>> b = bool(a\_list[1])

>>> c = bool(a\_list[2])

>>> d = bool(a\_list[3])

>>> print(not a, not b, c, d)

[False, False, True, True]

Exercise – Mutability

We’ve learned about many Python data structures (strings, lists, tuples, dictionaries). For both “mutable” and “immutable”, please give a short (5 words or fewer) definition, and then list what data structure(s) have that characteristic.

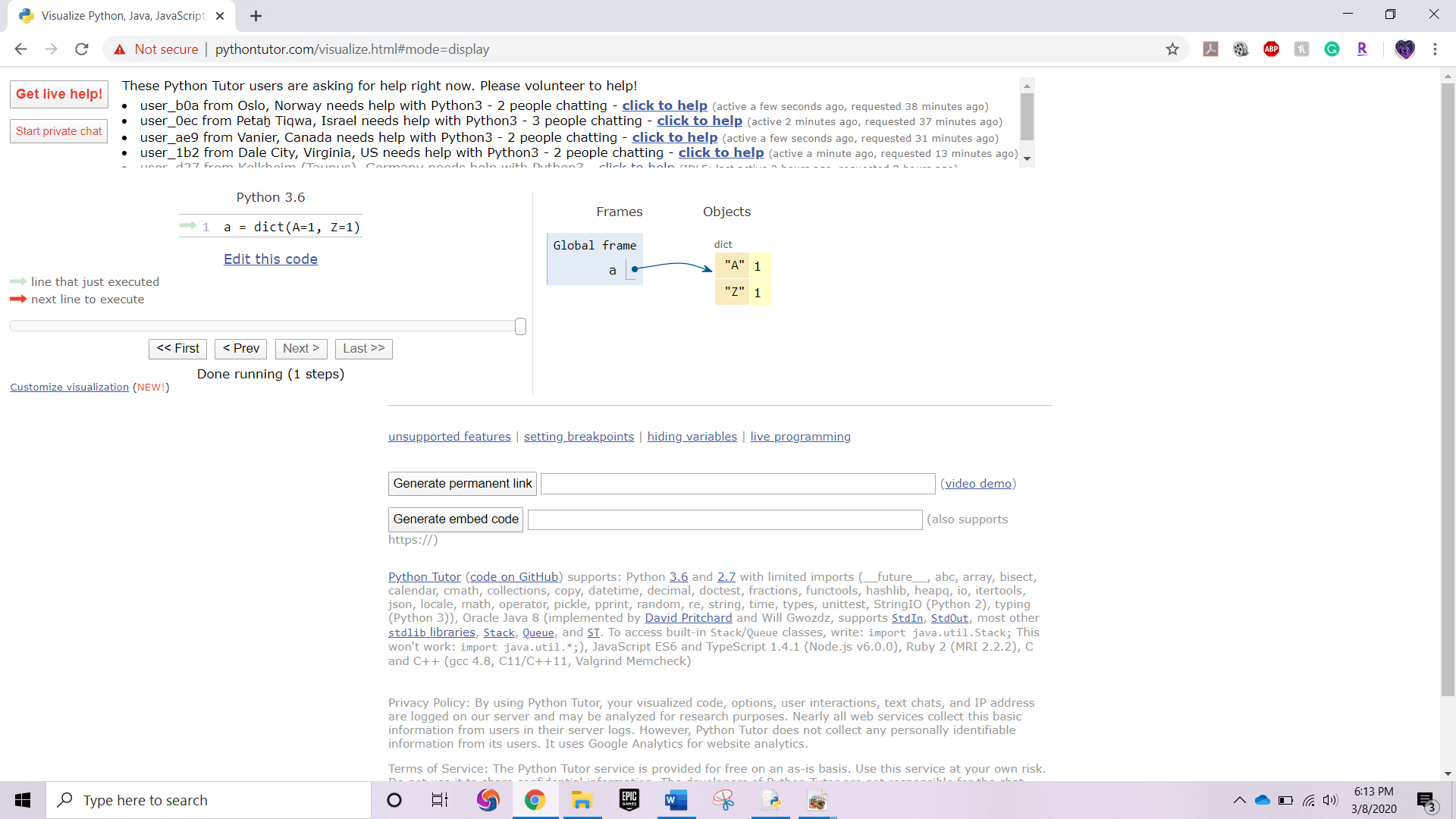
Mutable: Can be changed after created. List, dict, set

Immutable: Can’t be changed after created. Int, float, bool, string, Unicode, tuple

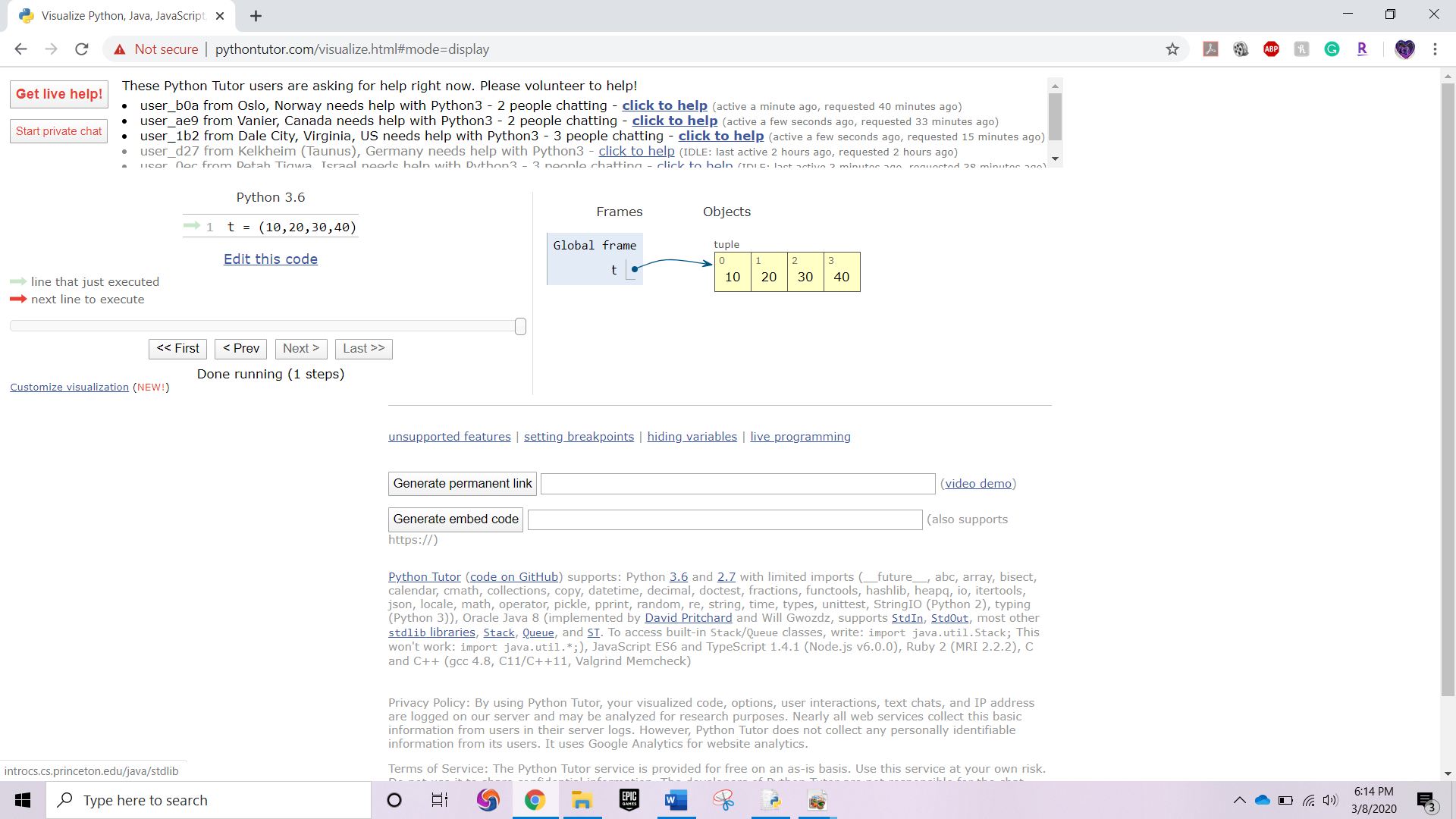
Exercise – Python Tutor

Go to the website pythontutor.com and generate the namespace and objects for the following data: (Include a screenshot of each in your submission)

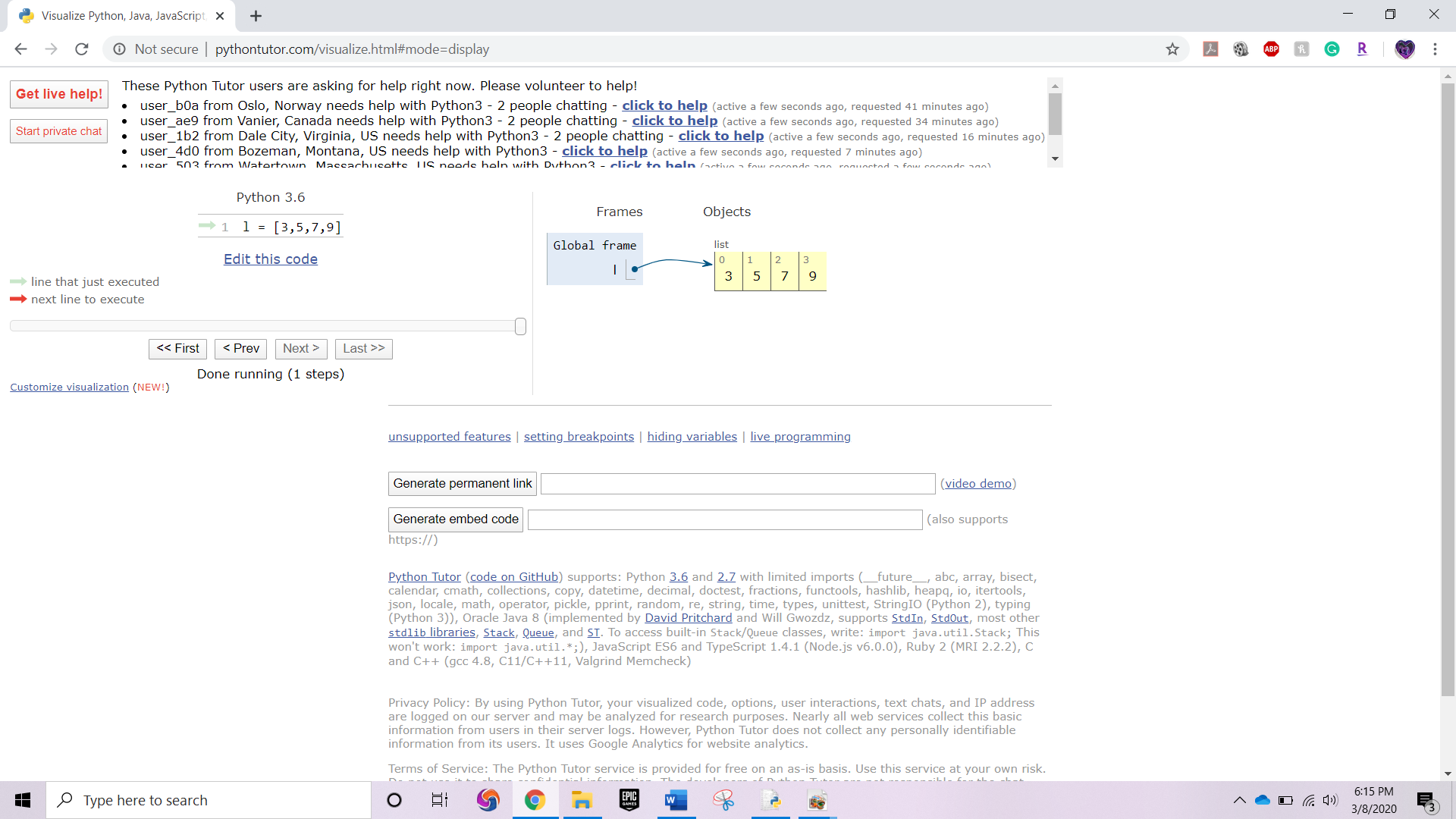
1. a = dict(A=1, Z=1)



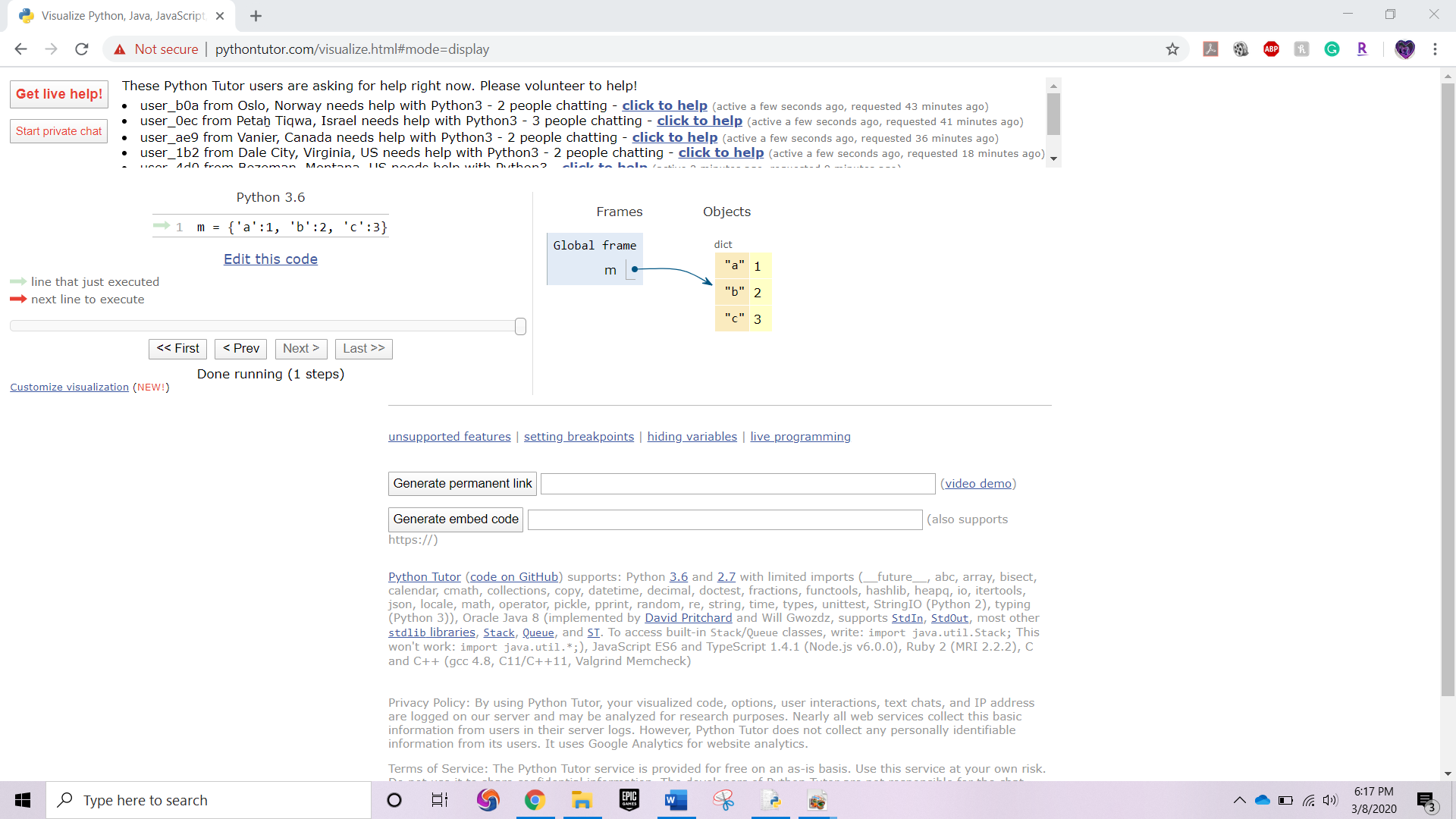
2. t = (10,20,30,40)



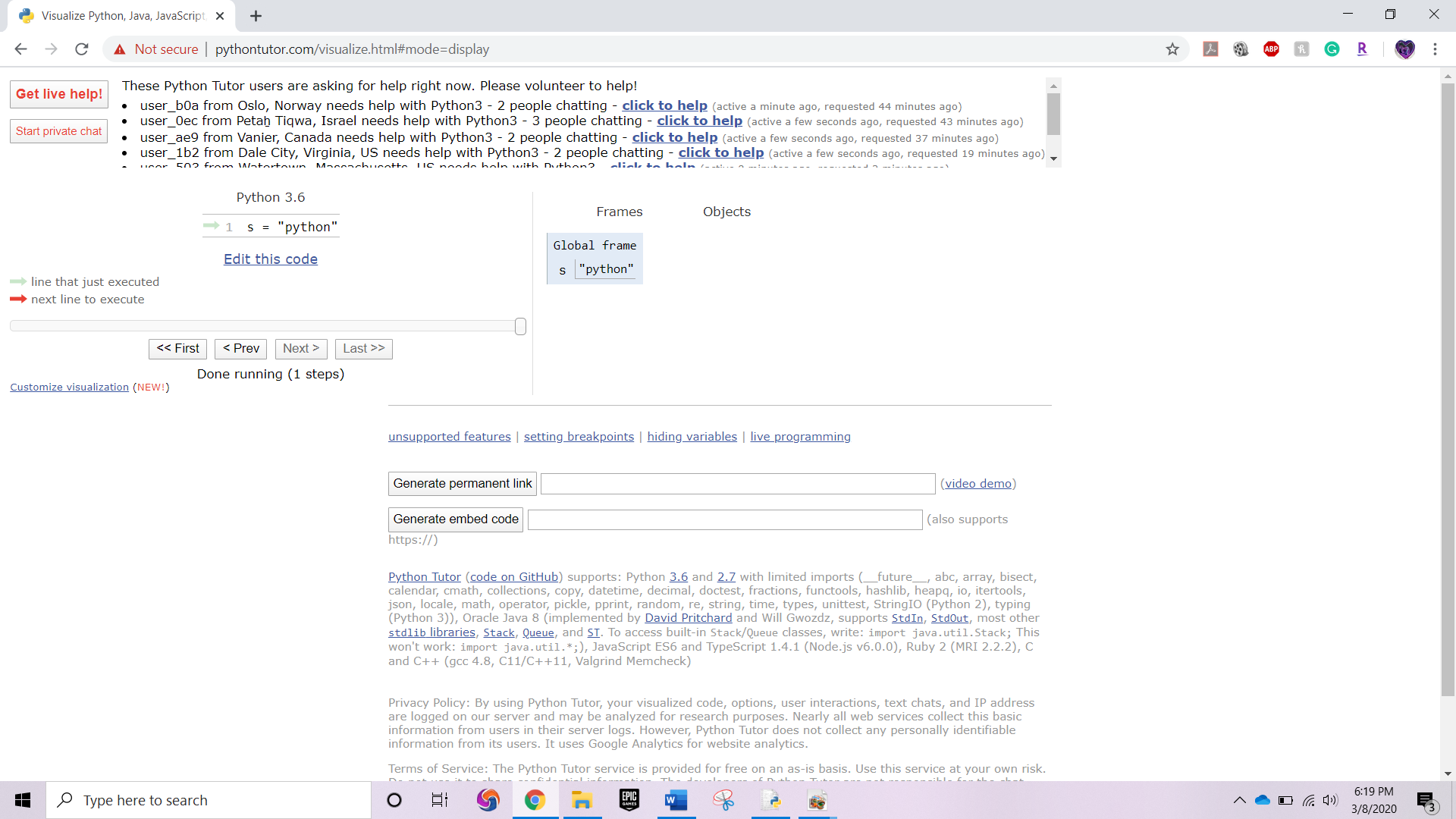
3. l = [3,5,7,9]



4. m = {‘a’:1, ‘b’:2, ‘c’:3}



5. s = “python”

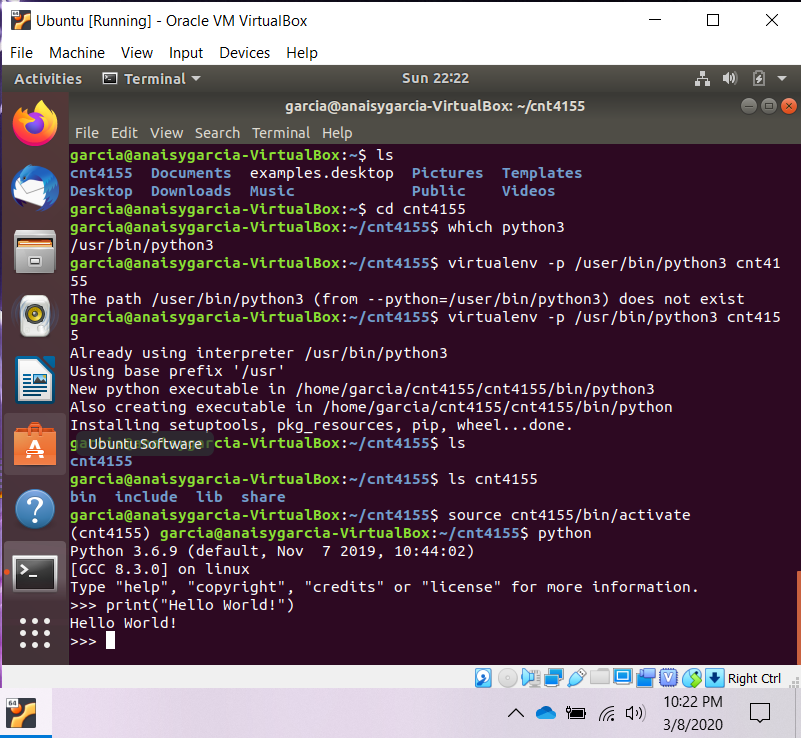


Exercise Hello, world!

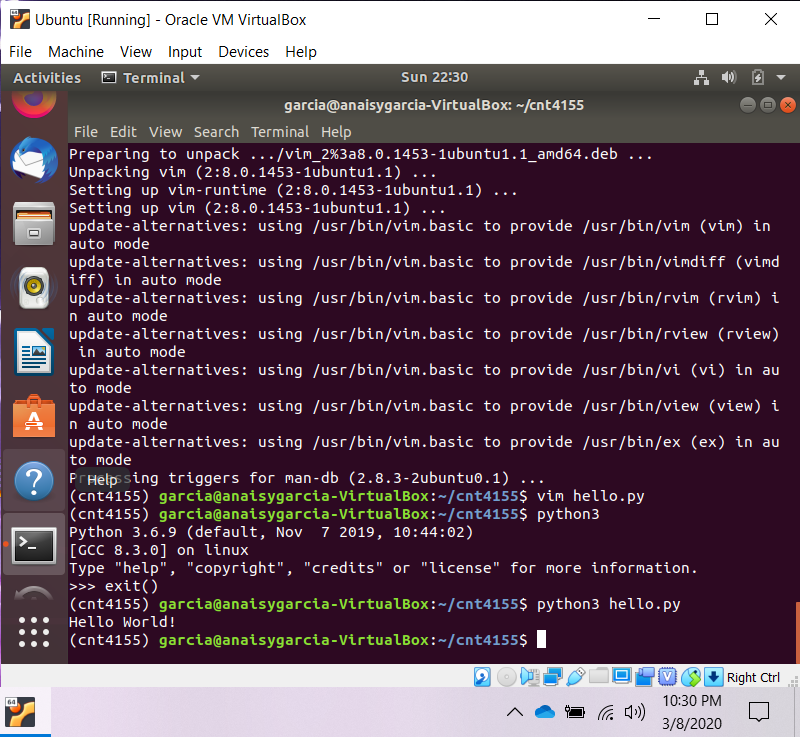
Recall that a program is just a set of instructions for the computer to execute. Let’s start with a basic command:

print x: Prints the value of the expression x, followed by a new line.

1. Open the python interactive shell and type the program that prints “Hello World!”.



2. Create a new program called helloworld.py. You will use this file to write your very first ‘Hello, world!’ program.



Execute both and get comfortable in using python code in both contexts. Include screenshots of both.