Audit Access Expires 5 de ago de 2020
You lose all access to this course, including your progress, on 5 de ago de 2020.

Exercise 3

Finger Exercises due Aug 5, 2020 20:30 -03 *Completo*

Exercise 3

8/8 points (graded)
ESTIMATED TIME TO COMPLETE: 6 minutes

Consider the function `normalize` that takes as input a list of positive numbers `numbers` and returns a list of numbers that are a fraction of the maximum element in the list. Try to answer the questions without running the code. Check your answers, then run the code for the ones you get wrong. You'll learn the most this way, by figuring things out, instead of just running the code and reading off the answers.

```
def normalize(numbers):
    max_number = max(numbers)
    for i in range(len(numbers)):
        numbers[i] /= float(max_number)
    return numbers
```

The code below tries to call `normalize` with one particular input. Answer the next 5 questions based on the following code.

```
try:
      normalize([0, 0, 0])
except ZeroDivisionError:
      print('Invalid maximum element')
```

1. Does the `try` block throw (also known as raise) an exception?

- (●) Yes
- ( ) No

✔

2. What is the name of the exception the code is trying to catch?

ZeroDivisionError ✔

Answer: ZeroDivisionError

3. What is the output?

Invalid maximum element ✔

Answer: Invalid maximum element

4. Since we are dividing by the maximum element in a list of positive numbers, we know that `normalize` will return a value between 0 and 1. What type of condition is this?
   - ( ) pre condition
   - (●) post condition

   ✔

5. We also know the result is not meaningful when the maximum element is 0, so we want to ensure that the numbers in the list do not violate this. What type of condition is this?
   - (●) pre condition
   - ( ) post condition

   ✔

Now assume the definition of the function `normalize` is rewritten as follows

```
def normalize(numbers):
    max_number = max(numbers)
    assert(max_number != 0), "Cannot divide by 0"
    for i in range(len(numbers)):
        numbers[i]  /= float(max_number)
        assert(0.0 <= numbers[i] <= 1.0), "output not between 0 and 1"
    return numbers
```

Answer the next 3 questions based on this code.

1. Which condition does the line `assert(max_number != 0)` correspond to?

- (●) pre condition
- ( ) post condition

✔

2. Which condition does the line `assert(0.0 <= numbers[i] <= 1.0)` correspond to?

- ( ) pre condition
- (●) post condition

✔

3. What does the function call `normalize([0, 0, 0])` print out?

Cannot divide by 0 ✔

Answer: AssertionError: Cannot divide by 0

Enviar

ℹ Answers are displayed within the problem

Exercise 3

Topic: Lecture 8 / Exercise 3

Ocultar discussão

Show all posts por atividade recente

There are no posts in this topic yet.

✖