EngrMAE-10 Fall 2018, Matlab R2018a +
10/16/2018, Nolan Bonnie (LARC Tutor)

Addressing the Confusion between ' ' and " "

MAE10 students,

Recently in a few of my LARC sessions, as well as a couple times in office hours, there has been some confusion revolving around declaring a string with single quotes ' ' or with double quotes " ". This confusion comes as result of the newest 2018 Matlab update, which made the single quotes and double quotes not equivalent anymore. I would like to clear up some of this confusion.

Let's start with the definitions:

SINGLE QUOTES: Single quotes are used to declare a **character array**, which is exactly as it sounds. A character array is an array filled with letters instead of numbers.

For example: >> x = 'test'

**Matlab stores this variable as   x = ['t', 'e', 's', 't']
and as a result can be indexed:**

>> x(1)

ans =

        't'

>> x(2)

ans =

        'e'

As a result, x has dimensions 1x4

>>size(x)

ans =

        1   4

<u>DOUBLE QUOTES:</u> Double quotes are used to declare a **string**. Think of a string as a box that holds all of your text together.

For example: >> y = "test"

**Matlab stores this variable as   y = ["test"]
and as a result can't be indexed by letter:**

>> y(1)

ans =

        "test"

>> y(2)   % Yields an error

Index exceeds array bounds.

As a result, y has dimensions 1x1

>>size(y)

ans =

        1   1

<u>SINGLE vs DOUBLE:</u> Initially I was very tempted to say use double quotes for all your work. I don't believe we'll come across a situation in this class where you'll need to index a character array - so it seems pointless to ever use the single quotes. **However** it appears that not all of Matlab's built in functions have been converted to accept double quotes. Which brings us to our first example of a contradiction.

input(' ')
It's very important for us to use the input function, especially on homework. Let's look at getting a string for an input. There are only 2 options to use the input function:

                input('prompt')
        or   input('prompt','s')

The first example, input('prompt') **<u>REQUIRES</u>** a character array prompt if you want input to prompt the user with a text, and will **ONLY** read in a numeric value from the user (unless the user explicitly typed in "test" for their input). The second way to use input is with input('prompt',**'s'**) which indicates that input will save user inputed text as a **character array**.

For example: >>x = input('Enter a value for x: ','s')

        >>x

        ans =

            'test'

BOOLEAN LOGIC OF STRINGS AND CHARACTER ARRAYS

Because of these differences, we must be careful when logically comparing a string and a character array. We'll split this into 4 cases to be thorough.

>>"test" == "test"

ans =

    logical

    1

>>'test' == "test"

ans =

    logical

    1

>>"test" == 'test'

ans =

    logical

    1

>>'test' == 'test'

ans =

    1×4 logical array

    1 1 1 1

Observe that the only case which produced an undesired result was when we compared two character arrays together. Because of this, when we compare character arrays from the input function to conditions — lets say in an if statement — it's best practice not to compare to another character vector.

I hope this clears up some of the confusion that has been going around. If there are any additional questions about this I'm always at Thursday's office hours to help.

-Nolan Bonnie