CS 101, LAB 2: ASSIGNMENTS AND STRINGS

Name:_____Group:_____Roll Number:_____

The purpose of this lab is to get you comfortable with using assignment statements, strings, and their associated methods.

This lab is very similar to the previous one, in that you will be typing commands into the Python interactive shell and recording the results.

Variables and Assignment Statements

As we saw in class, assignment statements are different from expressions. A statement like

a = 3 < 5

is a command to do something. In particular, this command

(1) evaluates the expression on the right-hand side of the = (in this case, 3 < 5), and

(2) stores its value in the variable on the left-hand side of the =, in this case, a.

Because it is not an expression, Python will not actually output a result when you type it in; it will just perform the command silently. It is important that you understand the difference. In the table on the next page, the first column contains either an expression or a command. If it is an expression, write the value. If it is a command, you should just write "None" (we have done the first one for you). Because some of the entries are commands, it is important that you enter the expressions or commands in exactly the order they are given.

Statement or Expression	Expected Value	Calculated Value	Reason for Calculated Value
i = 6	None		
i	6		
j	NameError: j not defined		j not yet defined
j = 1	None		
j+4	5		Gives value of 1+4 but note that j has not updated to 5
j = j + i	None		
j	7		1+6
i	6		

w = 'Hello'	None		
i + w	TypeError		‘int’ and ‘str’ are unsupported operand types for ‘+’

String Expressions

Throughout this section, pay close attention to spaces and to the different types of quotation marks being used. We use both ' (single quote) and " (double quote).

Expression	Expected Value	Calculated Value	Reason for Calculated Value
'Truth ' + 'is ' + ‘always’ + 'best'	‘Truth is alwaysbest’		
"Truth " + "is " + "best"	‘Truth is best’		
"Truth " + ('is ' + "best")	‘Truth is best’		
'A double quote: "'	'A double quote: "'		
"A single quote: '"	"A single quote: '"		
'A single quote: "	Syntax Error		EOL while scanning string literal

Expression	Expected Value	Calculated Value	Reason for Calculated Value
" + 'lol'	‘lol’		
" + '4 / 2'	‘4/2’		
" + 4 / 2	TypeError		Cannot implicitly convert float type to str
" + str(4 / 2)	‘2.0’		Evaluated float 2.0 is converted to string

Functions

Built-in functions are those that do not require you to import a module to use them. You can find a list of them in the Python documentation:

http://docs.python.org/library/functions.html

Note that the casting and typing operations are listed as functions. While this is true in Python, this is not always the case in other programming languages.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
min(25, 6)	6		Gives minimum of the two arguments
max(27, 4)	27		Gives maximum of the two arguments
min(25, max(28, 4))	25		Min on 25 and 28 is 25
abs(26)	26		Gives absolute value
abs(-26)	26		Gives absolute value
round(23.6)	24		Rounds off to nearest integer
round(-23.6)	-24		Rounds off to nearest integer
round(23.64, 0)	24.0		Rounds off to nearest float value accurate upto 0 decimal place
round(23.64, 1)	23.6		Rounds off to nearest float value accurate upto 1 decimal place
round(23.64, 2)	23.64		Rounds off to nearest float value accurate upto 2 decimal places
len('Truth')	5		Gives length of the string(includes white spaces as well if any)

Using a Python Module

One of the more important Python library modules is the math module. It contains essential mathematical functions like sin and cos. It also contains variables for mathematical constants such as pi. To learn more about this module, look at its online documentation:

http://docs.python.org/library/math.html

To use a module, you must import it. Type the following into the Python interactive shell:

```
import math
```

You can now access all of the functions and variables in math. However, to use any of them, you have to put "math." before the function or variable name. For example, to access the variable pi, you must type math.pi. Keep this in mind as you fill out that table below.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
math.sqrt(16)	4.0		
math.sqrt(-16)	math domain error		sqrt function expects a non-negative numeric value
math.floor(6.7)	6		Returns greatest integer lesser than

			or equal to the numeric value passed
math.ceil(6.7)	7		Returns smallest integer greater than or equal to the numeric value passed
math.ceil(-6.7)	-6		Returns greatest integer lesser than or equal to the numeric value passed
math.copysign(2,-6.7)	-2.0		Returns a float with the magnitude of first and sign of second
math.trunc(6.7)	6		Truncates value to integral
math.trunc(-6.7)	-6		
math.pi	3.141592...		Mathematical constant
math.cos(math.pi)	-1.0		Gives cosine of pi

In addition to the above expressions, type the following code into the Python interactive shell:

```
math.pi = 3
math.pi
```

What happens and why?

The value printed in 3 as the value of pi in the math module is overriden to the new value 3. It is possible to modify attribute values externally.

String Methods

Now that you understand strings and (calling) functions, you can put the two together. Strings have many handy methods, whose specifications can be found at the following URL:

http://docs.python.org/2/library/stdtypes.html#string-methods

Before starting with the table below, enter the following statement into the Python shell:

```
s = 'Hello World!'
```

Once you have done that, use the string stored in s to fill out the table, just as before.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
s[1]	‘e’		
s[15]	Index out of range		

s[1:5]	‘ello’		substring from index from 1 to 4 returned
s[:5]	‘Hello’		Substring before 5th index returned
s[5:]	‘ World’		Substring from 5th index to last returned
'H' in s	True		Checks for presence of substring in the given string
'x' in s	False		Checks for presence of substring in the given string
s.index('w')	Substring not found		
s.index('x')	Substring not found		
s.index('l', 5)	9		Returns index of ‘l’ found in substring from 5th index to last
s.find('e')	1		Returns index of first occurrence of ‘e’
s.find('x')	-1		If substring not found, find returns -1

Let q1 is a variable assigned as in the below given statement.

```
q1 = 'The phrase, "Don\'t panic!" is frequently uttered by consultants.'
```

You could also write a string like this using double quotes as the delimiters. Rewrite the assignment statement for q1 above using a double-quoted string literal:

```
q1="The phrase, \"Don\'t panic!\" is frequently uttered by consultants."
```

For your last exercise, we want you to write Python code to extract the substring inside the double quotes (which is "Don't panic!"). But we want to do it in a way that is independent of q1. That means, even if you change the contents of q1, your answer should still work, provided that q1 still has a pair of double-quote characters somewhere.

In the box below, write a sequence of one or more assignment statements, ending with an assignment to a variable called inner (the other variables can be named whatever you want). The assignment statements should use string slicing to remove the unwanted parts of q1. When you are done, the contents of inner should be the substring inside the double-quote characters (but not including the double quotes themselves).

```
s=q1.index("")
e=q1.index("",s+1)
inner=q1[s+1:e]

#Another approach
inner=q1.split("",2)[1]
```

To test that your statements are correct, do the following. First, type in

```
q1 = 'The phrase, "Don\'t panic!" is frequently uttered by consultants.'
```

Then type in your statements from the box above. Finally, print the value of inner. You should see Don't panic, without the quotes (printing always removes the quotes from a string value).

Now, try the process again with

```
q1 = 'The question "Can you help me?" is often asked in consulting hours.'
```

Type in the assignment statement above, then type in your statements from the box, and finally print the value of inner. You should see Can you help me?, without the quotes. If you had to modify your answer in the box for the second q1, you have done it incorrectly.