Closed book, closed notes, log out of computer! Cheat sheet on reverse. Please write neatly!

1. Answer T (true) or F (false) for the following statements [2 points]

   ___________ The literals "abc" + '' and 'abc' + "" create identical strings in Python
   ___________ The following expression len('one'+2') evaluates to 4

2. Evaluate the following expressions, writing the resulting string into the boxes, one character per box, and shading in unused boxes to indicate the end of the string (the empty string would have all boxes shaded). The value of astring is "a cs class". Note any capital letters. [5 points]

   For example: astring

   a | c | s | c | l | a | s | s |

   astring[3]

   astring[2:4] * 3

   astring[5:]

   astring[1::2].upper()

   astring[1:1]

3. When run via the green arrow in Thonny what will the following code print? [3 points]

   s = "AB"
   t = "150"
   s.lower()
   for i in range(len(s)):
       print(s[i])
       print(t[i])
**Numeric Operators**

+, -, /, *: Addition, subtraction, division, multiplication

//: Floor division: Round division result down to nearest whole number
%

: Modulus: Evaluate to remainder of division

- **Range**
  
  range(stop): Equivalent to range(0, stop, 1)

  range(start, stop[, step]): Create sequence of integers from inclusive start to exclusive stop by step

- **Slicing**

  seq[start[:stop][:step]]: Slice sequence seq from inclusive start to exclusive stop by step

- **Strings**

  - The following functions are built-in
    
    len(string): Returns the number of characters in the string

    int(string), float(string): Converts numeric string to int or float

    str(object): Converts object, e.g. int or float to a string

    sorted(string): Returns the characters of the string as a list in sorted order

  - **String object methods**

    upper(), lower(), capitalize(): Returns a new upper or lower-cased, or 1st letter upper-cased string

    find(some_string): Returns the first index that some_string occurs at in the string or -1 if not found

    find(some_string, index): Same as above, but starts searching at index

    replace(old, new): Return a copy of the string with all occurrences of old substituted with new

    startswith(prefix): Returns True if the string starts with prefix, False otherwise

    endswith(suffix): Returns True if the string ends with suffix, False otherwise

    strip(): Returns a copy of the string with only the leading and trailing whitespace removed

    split(): Return a list of the words in the string using whitespace as the delimiter

  - **String operators**

    string1 + string2: Returns a new string that is the concatenation of string1 and string2

    string * int: Returns a new string that is string repeated int times

    substr in string: Returns True if substr is a substring of string, False otherwise

- **Modules**

  - **random module**

    randint(a, b): Return a random integer $N$ such that $a \leq N \leq b$

    uniform(a, b): Return a random floating point number $N$ such that $a \leq N \leq b$

  - **math module**

    sqrt(num): Return the square root of num