# text organization on screeen

print("String organization with newlines and tabs:")

print("'line 1 \nand new line 2\n\t<- tabbed, done'\n")

# here's a string we'll use

string = "Here's A string of text"

print("\na predefined string to work with 's = text':")

print(string)

# show length of characters in string sequence

print("\nhow many characters in string ' len(string)':\n",len(string))

# indexing

print("\nzero index of string '(string[0])':\n",string[0])

print("\none index of string '(string[1])':\n",string[1])

# slicing shows only parts of strings for specific uses

print("\nin slicing, only 'string[:]' prints everything:\n", string[:])

print("\nslicing from string index 1 onward 'string[1:]':")

print(string[1:])

print("\nup to the fifth index 'string[:5]':\n",string[:5])

print("\nslicing backwards, the last character '(string[-1])':\n", string[-1])

print("\neverything but the last character 'string[:-1]:\n",string[:-1])

print("\nslicing steps or patterns,this means every other step'string(::2)':\n",string[::2])

print("\nusing string steps, the string can be printed backwards (string[::-1]):\n",string[::-1])

# concatenate string to add more text

print("\ntemporarily add text to string (string + 'more text'):\n",string + ' and more text')

print("\npermanently change the sequence of string value (string = string + new text):")

string = string + " and more text"

print(string)

# changing the string text for the next set of examples

print("\nim changing the string value to smaller stuff:")

string = "Text"

print(string)

# repeat the string

print("\nrepeat strings by multiplying (string\*10):\n", string\*10)

# make string uppercase or lowercase

print("\nstring methods by function. here's 'string.Upper()': \n", string.upper())

print("\nand 'string.lower():\n", string.lower())

# changing string again for new examples

print("\nI'm changing 'string' again:")

string = "Hello World"

print(string)

# splitting strings can remove string sequence items or other things

print("\nspliting strings at blank space (string.split()):\n",string.split())

print("\nsplit the string wherever the element is (string.split('l')):\n",string.split('l'))

# string formatting can put variables directly into print statements or specify how numbers should act

# '%s' states that a string variable is to be inserted, '%(string)' defines the string to be used.

print("\nstring formatting can put variables into print statement: 'print('var is: %s' %(string))':")

print("variable is: %s" %(string))

print("\nmultiple variables in strings ('strings %s %s' %('hi','tj')):\n","strings: %s %s:" %('hi','tj'))

print("""\nformatting float number to get an accurate number.

use '%n1.n2f', where n1 is how large the whole number could be

(not to important, leave at 1), n2 is how many digits after decimal point.

this makes 500 shortened to 5 using ('%0.1f' %(3.500)):""")

print("number was 3.500, is now: %0.1f" %(3.500))

# insert an int variable to string

print("\n'%s' can also convert to string: ''%s' %(1)':")

print("convert int to string: %s" %(1))

# quicker cleaner way to define multiple variables using item.format()

print("\ncleaner way to define in-line variables: ('x: {x} y: {y}'.format(x='x',y='y'))")

print('item1: {x} item2: {y}'.format(x='one',y='two'))