Name: _____ **Section:** 1 2 3 4

1 = Mutchler, 1st-2nd periods. **2 =** Mutchler, 3rd-4th periods. **3 =** Anderson, 7th-8th periods. **4 =** Anderson, 9th-10th periods.

Use this quiz to help make sure you understand the videos/reading. ***Answer all questions.*** Make additional notes as desired. ***Not sure of an answer?*** Ask your instructor to explain in class and revise as needed then. **Please print two-sided.**

Textbook Reading: ***Section 2.4 – Strings*** (pages 48-54)
Supplemental Reading: ***String methods from Python.og, Strings from "How to Think Like a Computer Scientist: Learning with Python 3" See index page for links and descriptions.***

1. In the ASCII subset of Unicode, what is the decimal value of the code for the question mark character, '?'
   Answer here _____

2. The **ord** function converts a _____ to a _____.
   The **chr** function converts a _____ to a _____.

3. A backslash in a string begins an _____ sequence.

4. **'\n'** is the way we represent the _____ character in Python code.

5. Unicode attempts to encode _____.

6. The remainder of this quiz consists of Python code. **Beside each print statement, write what it prints.** **First, you should try to figure out** what each statement does (in many cases this will require info from the Supplemental Reading documents (linked form the main page for Session 17) or from Google searches.

   A major part of this problem is practice finding information that is not in your textbook. Programmers spend a lot of time doing this kind of thing. If you are not sure of your answer, actually enter the code in Python. **If you are still not sure,** make a note (on this quiz is fine) so you can ask "why" in the Session 15 class.

   Note: It will be tempting to "just get the answers" by simply typing the code in Python instead of first trying to figure it out yourself. **This quiz will be mostly worthless to you** if that is what you do.

```
import math

s1 = 'the mighty Nile river'

print(s1.lower())

print(s1.capitalize())
```

```
print(s1)

print(s1.title())

print(s1.replace('i', ''))

print(s1.count('i'))

print(s1.count('n'))

print(s1.lower().count('n'))

print(s1.endswith('ver'))

print(s1.find('mi'))

print(s1.find('hg'))

print(s1[4:10])  # this is called a slice

print(s1[-3:-1])

print(s1[1:12:3])

print(s1[12:1:-3])



print(s1[:3])

print(s1[11:])

s2 = s1[:]

print(s2 == s1)

word_list = s1.split(' ')
```

```
print(word_list)

print('!!'.join(word_list))

s2 = 'abcd'

print(list(s2))

print('_'.join(list(s2)))

for c in s2:

    print(c, end=':')

print()

print(s2 < 'cde')

print(s2 < "Olin")  # think ASCII

print(s2 < "Olin".lower())

print('a' in 'apple')

print('pl' in 'aple')

print('pa' in 'apple')

print('' in'apple')

print('apple' in 'apple')

print([1, 2] == [1, 2])  # same contents?

print([1, 2] is [1, 2])  # same object?

print('abc' == 'abc')
```

```
print('abc' is 'abc')

print('*' + '   abc\n ' + '*')

print('*' + '   abc\n '.strip() + '*')

print('*' + s2.center(8) + '*')

print('*' + s2.ljust(8) + '*')

print('*' + s2.rjust(8) + '*')

s3 = '1\n23\n456\n7890'

print(s3)

print(s3.splitlines())

print(s3.splitlines()[2])  # are there quotes?

print ("The square root of {0} is {1}".format(7, math.sqrt(7)))

print ("The square root of {0} is {1:.3f}".format(7, math.sqrt(7)))

print ("The square root of {0} is {1:7.3f}".format(7, math.sqrt(7)))

print ("The square root of {} is {:7.3f}".format(7, math.sqrt(7)))

print ("The square root of {1} is {0:.3f}".format(math.sqrt(7), 7))
```