Translate the Following Into Python Algebraic or Boolean Expressions and Then Evaluate Them:

In [ ]:
```
# The Difference Between Annie's Age (25) and Ellie's (21)
```

In [1]:
```
25 - 21
```

Out[1]: 4

In [ ]:
```
# The Total of  14.99, 14.99, 27.95, and 19.83
```

In [2]:
```
14.99 + 14.99 + 27.95 + 19.83
```

Out[2]: 77.75999999999999

In [ ]:
```
# The Area of a Rectangle of Length 20 and Width 15
```

In [3]:
```
20 * 15
```

Out[3]: 300

In [ ]:
```
#  2 to the 10th Power
```

In [4]:
```
2**10
```

Out[4]: 1024

In [ ]:
```
# The Minimum of 3, 1, 8, -2, 5, -3, and 0
```

In [5]:
```
min(3, 1, 8, -2, 5, -3, 0)
```

Out[5]: -3

In [ ]:
```
# 3 Equals 4-2
```

In [6]:
```
3 == 4 - 2
```

Out[6]: False

In [ ]:
```
#  The Value of 17 // 5 Is 3
```

In [7]:
```
17 // 5 == 3
```

Out[7]: True

In [ ]:
```
#  The Value of 17 % 5 Is 3
```

In [8]:
```
17 % 5 == 3
```

Out[8]: False

In [ ]:
```
# 284 is Even
```

In [9]:
```
284 % 2 == 0
```

Out[9]: True

In [ ]:
```
# 284 Is Even and 284 Is Divisible by 3
```

In [10]:
```
284 % 2 == 0 and 284 % 3 == 0
```

Out[10]: False

In [ ]:
```
# 284 Is Even or 284 Is Divisible by 3
```

In [11]:
```
284 % 2 == 0 or 284 % 3 == 0
```

Out[11]: True

Write Python Expressions Involving Strings s1, s2, and s3 That Correspond to:

In [12]:
```
s1 = 'good'
s2 = 'bad'
s3 = 'silly'
```

In [ ]:
```
# "ll" Appears In s3
```

In [13]:
```
'll' in s3
```

Out[13]: True

In [ ]:
```
# The Blank Space Does Not Appear In s1
```

In [14]:
```
' ' not in s1
```

Out[14]: True

In [ ]:
```
# The Concatenation of s1, s2, and s3
```

In [15]:
```
s1 + s2 + s3
```

Out[15]: 'goodbadsilly'

In [ ]:
```
# The Blank Space Appears In the Concatenation of s1, s2, and s3
```

In [16]:
```
' ' in s1 + s2 + s3
```

Out[16]: False

In [ ]:
```
# The Concatenation of 10 Copies of s3
```

In [17]:
```
10 * s3
```

Out[17]: 'sillysillysillysillysillysillysillysillysillysillysilly'

In [ ]:
```
# The Total Number of Characters In the Concatenation of s1, s2, and s3
```

In [18]:
```
len(s1 + s2 + s3)
```

Out[18]: 12

String s Is Defined to be "abcdefgh". Write Expressions Using s and the Indexing Operator [] That Return the Following Strings:

In [19]:
```
s = 'abcdefgh'
```

In [20]:
```
# a
s[0]
```

Out[20]: 'a'

In [21]:
```
# c
s[2]
```

Out[21]: 'c'

In [22]:
```
# h
s[-1]
```

Out[22]: 'h'

In [23]:
```
# h
s[len(s) - 1]
```

Out[23]: 'h'

List lst Is a List of Prices For a Pair of Boots at Different Online Retailers

In [24]:
```
lst = [159.99, 160.00, 205.95, 128.83, 175.49]
```

In [ ]:
```
# a. You Found Another Retailer Selling the Boots for $160.00
# Add This Price to List lst
```

In [25]:
```
lst.append(160.00)
```

In [26]:
```
lst
```

Out[26]: [159.99, 160.0, 205.95, 128.83, 175.49, 160.0]

In [ ]:
```
# b. Compute the Number of Retailers Selling the Boots for $160.00
```

In [27]:
```
lst.count(160.00)
```

Out[27]: 2

In [ ]:
```
# c. Find the Minimum Price In lst
```

In [28]:
```
min(lst)
```

Out[28]: 128.83

In [ ]:
```
# Using c), Find the Index of the Minimum Price In List lst
```

In [29]:
```
lst.index(min(lst))
```

Out[29]: 3

In [ ]:
```
# e. Using c), Remove the Minimum Price From List lst
```

In [30]:
```
lst.remove(min(lst))
```

In [31]:
```
lst
```

Out[31]: [159.99, 160.0, 205.95, 175.49, 160.0]

In [ ]:
```
# f. Sort List lst In Increasing Order
```

In [32]:
```
lst.sort()
```

In [33]:
```
lst
```

Out[33]: [159.99, 160.0, 160.0, 175.49, 205.95]

Write a Python Expression That Assigns to Variable c

In [ ]:
```
# a. The Length of the Hypotenuse In a Right Triangle Whose Other Two Sides Have Length
```

In [34]:
```
import math
c = math.sqrt(3**2 + 4**2)
c
```

Out[34]: 5.0

In [ ]:
```
# b. The Value of the Boolean Expression That Evaluates Whether the Length of the Above
```

In [35]:
```
c = math.sqrt(3**2 + 4**2) == 5
c
```

Out[35]: True

In [ ]:
```
# c. The Area of a Disk of Radius 10
```

In [36]:
```
c = math.pi * 10 ** 2
c
```

Out[36]: 314.1592653589793

In [ ]:
```
# d. The Value of the Boolean Expression That Checks Whether
# a Point With Coordinates (5, 5) Is Inside a Circle With Center (0, 0) and Radius 7
```

In [37]:
```
c = math.sqrt(5**2 + 5**2) < 7
c
```

Out[37]: False