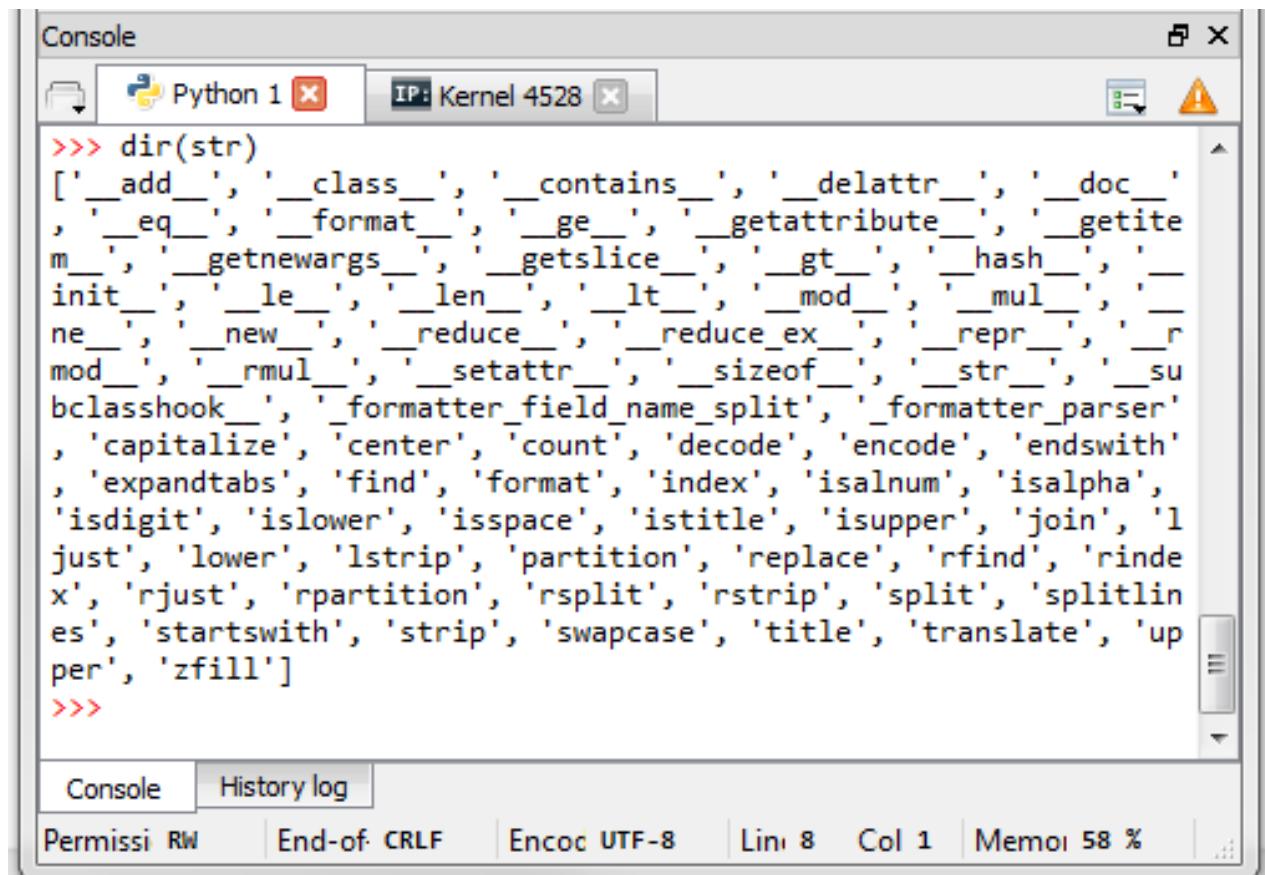


Strings: Methods

str Type Methods

>>> *dir(str)*



```
Console
Python 1 IP: Kernel 4528
>>> dir(str)
['__add__', '__class__', '__contains__', '__delattr__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
 '__getnewargs__', '__getslice__', '__gt__', '__hash__', '__init__',
 '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__',
 '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__',
 '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__',
 '__formatter_field_name_split__', '__formatter_parser__',
 'capitalize', 'center', 'count', 'decode', 'encode', 'endswith',
 'expandtabs', 'find', 'format', 'index', 'isalnum', 'isalpha',
 'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust',
 'lower', 'lstrip', 'partition', 'replace', 'rfind', 'rindex',
 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines',
 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper',
 'zfill']
>>>
Console History log
Permissions RW End-of CRLF Encoding UTF-8 Line 8 Col 1 Memory 58 %
```

>>> *str('Mozzarella').istitle()*

True

>>> *str("Mozzarella").find("ar")*

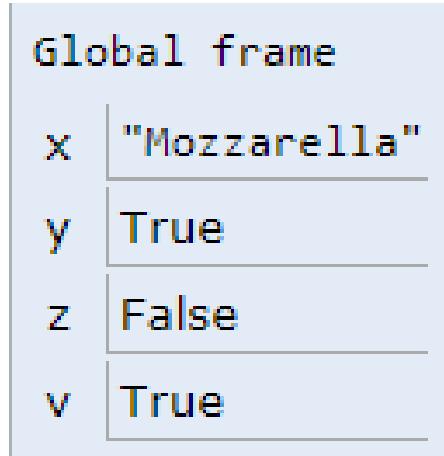
4

Methods: *startswith()*, *endswith()*

0	1	2	3	4	5	6	7	8	9
M	O	Z	Z	a	r	e	I	I	a
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
>>> x = "Mozzarella"  
>>> y = x.startswith("Moz")  
>>> z = x.startswith("hello")  
>>> v = x.endswith("ella")
```

Frames	Objects								
<pre>Global frame</pre> <table style="width: 100%; border-collapse: collapse;"><tr><td style="width: 10%;">x</td><td style="width: 90%;">"Mozzarella"</td></tr><tr><td>y</td><td>True</td></tr><tr><td>z</td><td>False</td></tr><tr><td>v</td><td>True</td></tr></table>	x	"Mozzarella"	y	True	z	False	v	True	
x	"Mozzarella"								
y	True								
z	False								
v	True								

A screenshot of a Python debugger interface. It shows a table titled 'Global frame' with four rows. The first row has 'x' in blue and a text input field containing 'Mozzarella'. The second row has 'y' in blue and a text input field containing 'True'. The third row has 'z' in blue and a text input field containing 'False'. The fourth row has 'v' in blue and a text input field containing 'True'. The background is light grey, and the table has a thin black border.

Methods: *lower()*, *upper()*

0	1	2	3	4	5	6	7	8	9
M	O	Z	Z	a	r	e	I	I	a
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
>>> x = "Mozzarella"
```

```
>>> y = x.upper()
```

```
>>> z = x.lower()
```

Frames

Objects

Global frame	
x	"Mozzarella"
y	"MOZZARELLA"
z	"mozzarella"

Methods: *find()*, *count()*

0	1	2	3	4	5	6	7	8	9
M	O	Z	Z	a	r	e	l	l	a
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
>>> x = "Mozzarella"  
>>> y = x.find("ella")  
>>> z = x .find("hello")  
>>> v = x count("z")  
>>> w = x.count("zoo")
```

Frames

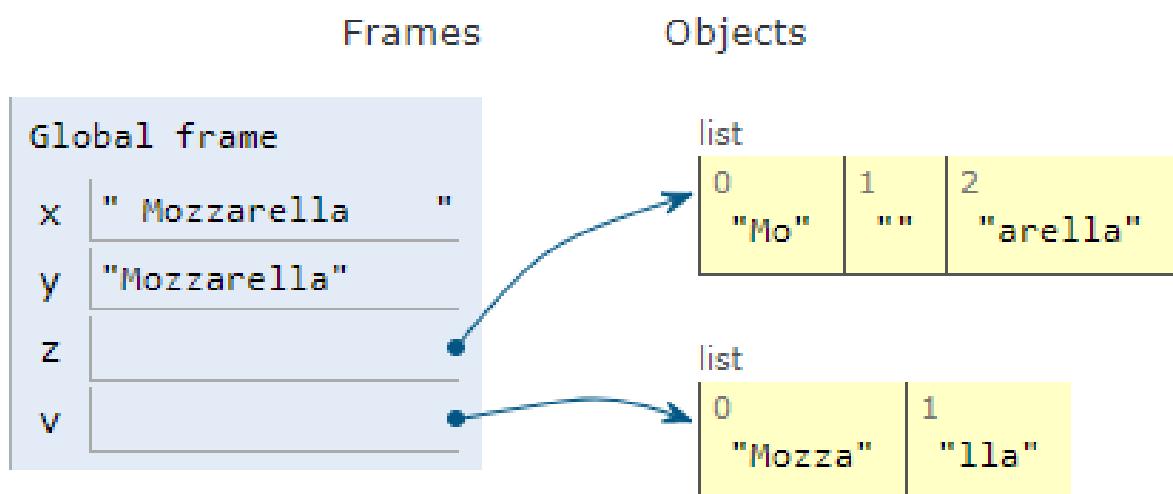
Objects

Global frame	
x	"Mozzarella"
y	6
z	-1
v	2
w	0

Methods: *strip()*, *split()*

0	1	2	3	4	5	6	7	8	9
M	o	z	z	a	r	e	l	l	a
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

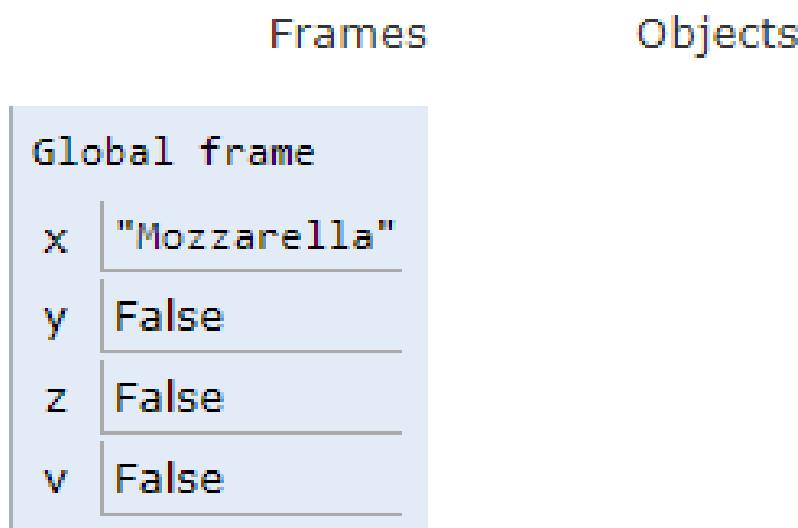
```
>>> x = " Mozzarella "
>>> y = x.strip()
>>> z = y.split("z")
>>> v = y.split("re")
```



Methods: *islower()*, *isupper()*, *isdigit()*

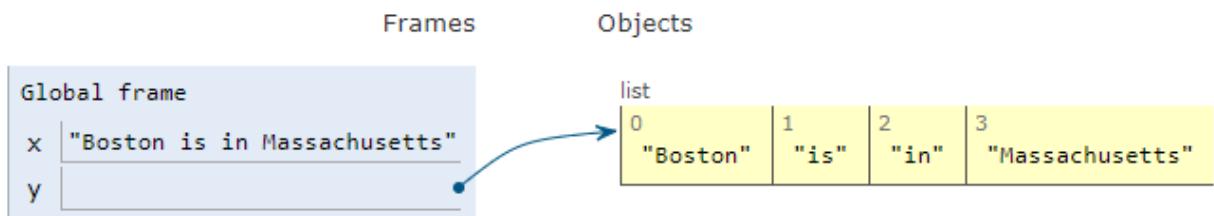
0	1	2	3	4	5	6	7	8	9
M	O	Z	Z	a	r	e	I	I	a
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
>>> x = "Mozzarella"  
>>> y = x.isupper()  
>>> z = x.islower()  
>>> v = x.isdigit()
```

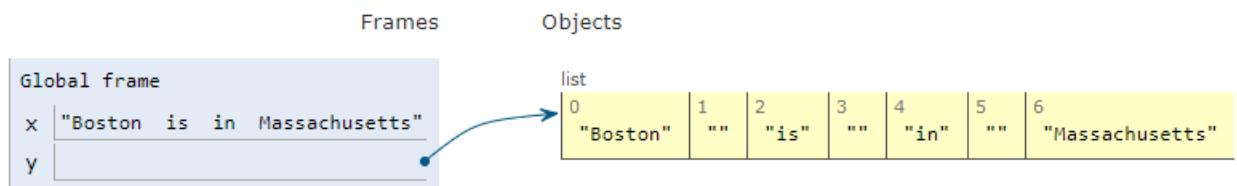


Methods: *split()*

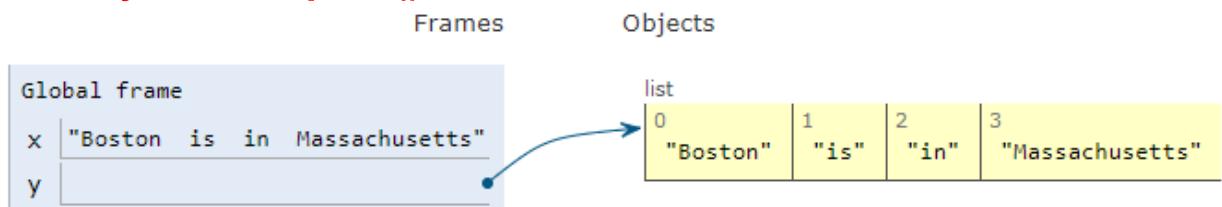
```
>>> x = "Boston is in Massachusetts"  
>>> y = x.split(' ')
```



```
>>> x = "Boston is in Massachusetts"  
>>> y = x.split(' ')
```



```
>>> x = "Boston is in Massachusetts"  
>>> y = x.split()
```



Chaining Methods

0	1	2	3	4	5	6	7	8	9
M	o	z	z	a	r	e	l	l	a
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

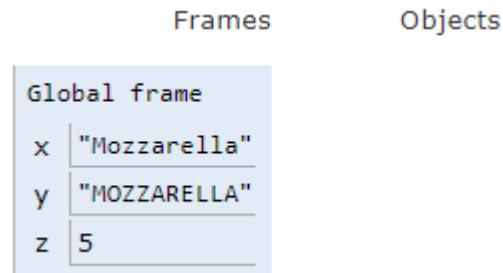
```
>>> x = "Mozzarella"
```

```
>>> y = x.isupper()
```

```
>>> z = y.find('RE')
```

>>> Z

5



- can chain methods together
 - evaluation from left to right

```
>>> z = x.isupper().find('RE')
```

Review Problems

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and red, and a help/gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is visible. The main area contains a question: "Write an expression that evaluates to True if the str associated with s starts with 'p'." A vertical scrollbar is on the right side of the main content area.

Exercise 51757 —

WORK AREA **SOLUTIONS**

Content Support

Write an expression that evaluates to True if the str associated with s starts with "p".

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and red, and a help/gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following question:

Write an expression that returns False if the str associated with s begins with "ecto".

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and orange, and a help/gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following question:

Write an expression that returns True if the str associated with s ends with "ism".

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and red, and a help icon with a question mark and gear. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located near the bottom left of the main area. The main content area contains the following text:

Write an **expression** whose value
is True if all the letters in the str
associated with s are all lower case.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT'. To the right of these are links for 'Workbench', '?', and a gear icon. Below the navigation is the title 'Exercise 51767 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above a large text area. The text area contains the following instruction:

Write an **expression** whose value is True if all the letters in the str associated with s are upper case.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help/gear icon on the right. Below these are two tabs: 'WORK AREA' (selected) and 'SOLUTIONS'. A 'Content Support' button is located above the main content area. The main content area contains the following text:

Write an **expression** whose value is the same as the str associated with s but with **reversed case**. Thus, if the str associated with s is "McGraw", the value of the expression would be "mCgRAW".

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help/gear icon on the right. Below the title, the text 'Exercise 51773 —' is displayed. Underneath, there are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located on the right side of the main content area. The main content area contains the following text:

Given `variables` `first` and `last`, each of which is associated with a str, representing a first and a last name, respectively. Write an `expression` whose value is a str that is a full name of the form "Last, First". So, if `first` were associated with "alan" and `last` with "turing", then your expression would be "Turing,Alan". (Note the capitalization! Note: no spaces!) And if `first` and `last` were "Florean" and "fortescue" respectively, then your expression's value would be "Fortescue,Florean".

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a software interface for a programming exercise. At the top, there are navigation buttons labeled "PREV" and "NEXT" with circular arrows, a title "Workbench" in the center, and a help icon with a question mark and gear symbol on the right. Below the title, the text "Exercise 51890—" is displayed. Underneath, there are two tabs: "WORK AREA" and "SOLUTIONS". A green button labeled "Content Support" is located on the right side of the main content area. The main content area contains the following text:

Assume that sentence is a variable that has been associated with a string consisting of words separated by single space characters with a period at the end. For example: "This is a possible value of sentence."

Write the statements needed so that the variable secondWord is associated with the second word of the value of sentence . So, if the value of sentence were "Broccoli is delicious." your code would associate secondWord with the value "is" .

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and orange, and a help icon with a question mark and gear. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Write a sequence of statements that finds the first comma in the string associated with the variable line, and associates the variable clause the portion of line up to, but not including the comma.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon (?) and settings gear icon on the right. Below the navigation is the title 'Exercise 51270 —'. Underneath the title are two tabs: 'WORK AREA' (highlighted in white) and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main content area contains the following text: 'Given the string line, create a set of all the vowels in line. Associate the set with the variable vowels.
'. The entire interface is contained within a light gray frame.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons labeled "PREV" and "NEXT" with circular arrows, a "Workbench" button, and a help icon with a question mark and gear. Below the navigation is the title "Exercise 51271". Underneath the title are two tabs: "WORK AREA" and "SOLUTIONS". A green button labeled "Content Support" is located to the right of the tabs. The main content area contains the following text:

Given the strings `s1` and `s2` that are of the same length, create a new string consisting of the first character of `s1` followed by the first character of `s2`, followed by the second character of `s1`, followed by the second character of `s2`, and so on (in other words the new string should consist of alternating characters of `s1` and `s2`). For example, if `s1` contained "hello" and `s2` contained "world", then the new string should contain "hweolrlod". Associate the new string with the variable `s3`.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a software interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, and 'Workbench' in the center. To the right of 'Workbench' are two icons: a question mark and a gear. Below the top bar, the title 'Exercise 51272 —' is displayed. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main content area contains the following text:

Given the strings `s1` and `s2` that are of the same length, create a new string consisting of the last character of `s1` followed by the last character of `s2`, followed by the second to last character of `s1`, followed by the second to last character of `s2`, and so on (in other words the new string should consist of alternating characters of the reverse of `s1` and `s2`). For example, if `s1` contained "hello" and `s2` contained "world", then the new string should contain "odlllreohw". Assign the new string to the variable `s3`.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT'. To the right of these is the title 'Workbench' and a help icon (a question mark). Below the title, the text 'Exercise 51273 —' is displayed. Underneath this, there are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main area contains the following text:

Given the strings `s1` and `s2`, not necessarily of the same length, create a new string consisting of alternating characters of `s1` and `s2` (that is, the first character of `s1` followed by the first character of `s2`, followed by the second character of `s1`, followed by the second character of `s2`, and so on. Once the end of either string is reached, no additional characters are added. For example, if `s1` contained "abc" and `s2` contained "uvwxyz", then the new string should contain "aubvcw". Assign the new string to the variable `s3`.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a software interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue. To the right of 'Workbench' are a question mark icon and a gear icon. Below these are two tabs: 'WORK AREA' (highlighted in blue) and 'SOLUTIONS'. A 'Content Support' button is located in the top right corner of the main content area. The main content area contains the following text:

Given the strings `s1` and `s2`, not necessarily of the same length, create a new string consisting of alternating characters of `s1` and `s2` (that is, the first character of `s1` followed by the first character of `s2`, followed by the second character of `s1`, followed by the second character of `s2`, and so on. Once the end of either string is reached, the remainder of the longer string is added to the end of the new string. For example, if `s1` contained "abc" and `s2` contained "uvwxyz", then the new string should contain "aubvcwxyz". Associate the new string with the variable `s3`.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there is a toolbar with four green buttons: 'PREV' (left arrow), 'NEXT' (right arrow), 'Workbench' (blue text), a question mark icon, and a gear icon. Below the toolbar, the text 'Exercise 51279 —' is displayed. Underneath this, there are two tabs: 'WORK AREA' (highlighted in blue) and 'SOLUTIONS'. In the bottom right corner of the main content area, there is a green button labeled 'Content Support'. The main content area contains the following text:

Assign True to the variable `has_dups` if the string `s1` has any duplicate character (that is if any character appears more than once) and False otherwise

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue, and a help icon ('?') and settings icon ('⚙️') in green. Below these is a title 'Exercise 51006 —'. Underneath the title are two tabs: 'WORK AREA' (highlighted in blue) and 'SOLUTIONS' (in grey). A 'Content Support' button is located above a large text area. The main text area contains the following question:

Given a variable `t` that is associated with a tuple whose elements are strings, write some **statements** that use a **while loop** to count the number of tuple elements that are 4-character strings and associate that number with the variable `four_letter_word_count`.

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon (?) and settings icon (gear) on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main area contains the following text:

Given a **variable** `s` that is associated with the empty string, write some **statements** that use a **while loop** to associate `s` with a **string** consisting of exactly 777 asterisks (*) .

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons labeled "PREV" and "NEXT", a title "Workbench", and two icons: a question mark and a gear. Below the title, it says "Exercise 51003". There are two tabs: "WORK AREA" and "SOLUTIONS", with "WORK AREA" currently selected. A green button labeled "Content Support" is visible. The main area contains the following text:

Given a variable **n** that is associated with a positive integer and a variable **s** that is associated with the empty string, write some **statements** that use a **while loop** to associate **s** with a string consisting of exactly **n** asterisks (*). (So if **n** were associated with 6 then **s** would, in the end, be associated with "*****".)

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT'. To the right of these are links for 'Workbench', '?', and a gear icon. Below the navigation is the title 'Exercise 51004 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main content area contains the following text:

Given a variable `n` that is associated with a positive integer and a variable `s` that is associated with the empty string, write some statements that use a while loop to associate `s` with a string consisting of exactly `2n` asterisks (*). (So if `n` were associated with 4 then `s` would, in the end, be associated with "*****").

Programming Exercise

Worksheet

this page is intentionally left blank

Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below the navigation is the title 'Exercise 51005 —'. Underneath the title are two tabs: 'WORK AREA' (highlighted in blue) and 'SOLUTIONS'. A green button labeled 'Content Support' is located above a large text area. The text area contains the following problem statement:

Given a variable `s` that is associated with non-empty string, write some **statements** that use a **while loop** to associate a variable `vowel_count` with the number of lower-case vowels ("a", "e", "i", "o", "u") in the string .

Programming Exercise

Worksheet

this page is intentionally left blank