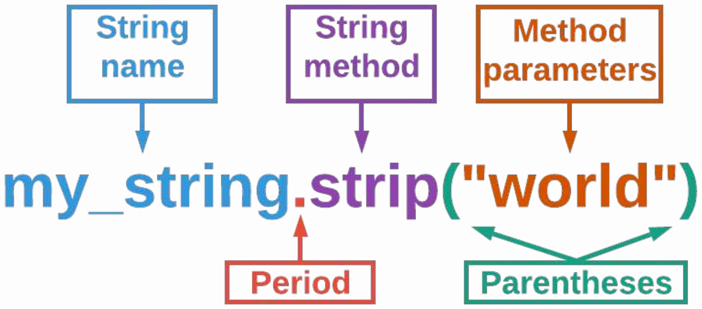
Learning Objectives - String Methods

- **Define a string method**
- **Describe the syntax of a method**
- **Identify some commonly used string methods**

Strip

What is a String Method?

Strings have special commands called methods (more on methods in a later lesson). Methods have a special syntax. First, start with a string (often a variable that represents a string). Add a period after the string. Finally, add the name of the method with any parameters. Parameters are values that the method will use.



String Method with Parameters

Translation: Remove the string `"world"` from the string `my_string`.

The Strip Method

The `strip` method removes characters from the beginning or end of a string. `strip` returns a modified copy of the original string.

```
string1 = "Hello world"
string2 = "world"
print(string1.strip(string2))
```

challenge

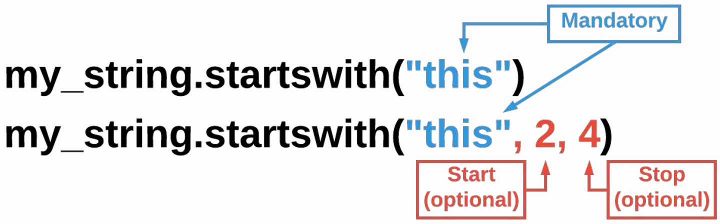
What happens if you:

- Change `string2` to `"Hello"`?
- Change `string2` to `"ld"`?
- Change `string2` to `"ell"`?

Starts With

The Starts With Method

The `startswith` method returns either `True` or `False` if a string starts with another string. For example, `my_string.startswith("this")` will return `True` if `my_string` starts with `"this"`. If not, it will return false. The `startswith` method is a bit different because some of the parameters are optional. The first parameter, a string, is mandatory. `startswith` will start the comparison with the first character in the string. However, you can change where the comparison starts and ends with the optional parameters.



```
my_string.startswith("this")
my_string.startswith("this", 2, 4)
```

Optional Parameters

```
my_string = "this is a string"
my_bool = my_string.startswith("this")
print(my_bool)
```

challenge

What happens if you:

- Change `my_bool` to `my_string.startswith("This")`?
- Change `my_bool` to `my_string.startswith("is", 2)`?
- Change `my_bool` to `my_string.startswith("is", 2, 3)`?
- Change `my_bool` to `my_string.startswith("is", 2, 4)`?

Replace

The Replace Method

The `replace` method returns a copy of the original string in which part of the original string (also called a substring) has been replaced with another set of characters.

```
my_string = "dog mouse fish dog bear"
new_string = my_string.replace("dog", "cat")
print(new_string)
```

challenge

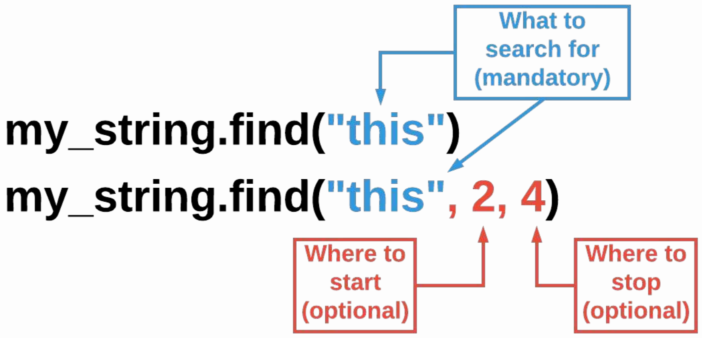
What happens if you:

- Change `my_string` to `"dogmousefishdogbear"`?
- Change `new_string` to `"my_string.replace("Dog", "cat")`?
- Change `new_string` to `"my_string.replace("dog", "cat", 1)`?

Find

The Find Method

The `find` method searches for a word or character in a string. If the word or character is found, the index is returned. If not, `-1` is returned. You can tell `find` where to start the search and where to end the search. By default, `find` will search the entire string.


What to search for (mandatory)
my_string.find("this")
my_string.find("this", 2, 4)
Where to start (optional)
Where to stop (optional)

Find Method

```
string1 = "The brown dog jumps over the lazy fox."
string2 = "brown"
print(string1.find(string2))
```

challenge

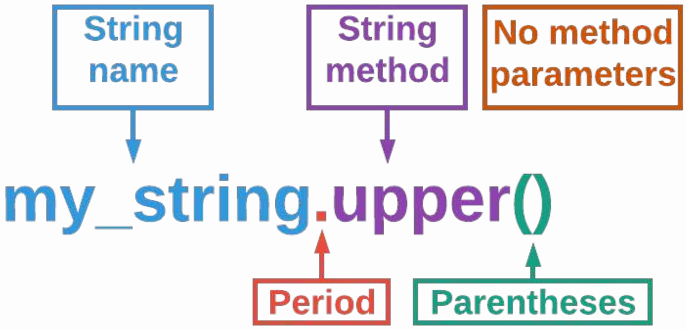
What happens if you:

- Change `string2` to `"zebra"`?
- Change `string2` back to `"brown"` and change the `print` statement to `print(string1.find(string2, 10))`?
- Change the `print` statement to `print(string1.find(string2, 0, 3))`?

Upper

The Upper Method

There are some string methods that do not require parameters. You still must use the parentheses even if there are no parameters. The `upper` method is an example of this. The `upper` method returns a copy of the original string with all uppercase characters.



String Method with No Parameters

Translation: Convert all the characters of `my_string` to uppercase.

```
my_string = "the big brown dog"
print(my_string.upper())
```

challenge

What happens if you:

- Change `my_string` to `"ThE bIg BrOwN dOg"`?
- Change `my_string` to `"THE BIG BROWN DOG"`?
- Change `my_string` to `"123!@#"`?

Lower

The Lower Method

The `lower` method creates a copy of a string, and returns the copy with all lowercase characters. `lower` does not take any parameters.

```
my_string = "THE BIG BROWN DOG"
print(my_string.lower())
```

challenge

What happens if you:

- Change `my_string` to `"tHe BiG bRoWn DoG"`?
- Change `my_string` to `"the big brown dog"`?
- Change `my_string` to `"214%#%"`?

Capitalize

The Capitalize Method

The `capitalize` method returns a copy of a string with only the first character capitalized.

```
my_string = "the big brown dog"
print(my_string.capitalize())
```

challenge

What happens if you:

- Change `my_string` to `"tHe BiG bRoWn DoG"`?
- Change `my_string` to `"THE BIG BROWN DOG"`?
- Change `my_string` to `"123^&*"`?

Title

The Title Method

The `title` method creates a copy of a string, and returns a string with the first letter of each word capitalized. All other characters of the word will be lowercase.

```
my_string = "the big brown dog"
print(my_string.title())
```

challenge

What happens if you:

- Change `my_string` to `"tHe BiG bRoWn DoG"`?
- Change `my_string` to `"thebigbrowndog"`?
- Change `my_string` to `"a1 1a a a a"`?

▼ **Other String Methods**

There are many more string methods. Here are a few examples:

Method	Example	Description
Center	`center(width, fill)`	Center a string in a given width, fill any whitespace with a given character
Count	`count(str, start, end)`	Count how many times a string appears
Ends With	`endswith(str, start, end)`	Return `True` if a string ends with a specific string
Index	`index(str, start, end)`	Return index of `str` in a string, will raise an exception if not found
Is Alphanumeric	`isalnum()`	Returns `True` if string is alphanumeric
Is Alphabetic	`isalpha()`	Returns `True` if string is alphabetic
Is Digit	`isdigit()`	Returns `True` if string is just digits
Is Lower	`islower()`	Returns `True` if the string is lowercase
Is Space	`isspace()`	Returns `True` if the strings is nothing but spaces

Is Title	`istitle()`	Returns `True` if the string is title case
Is Upper	`isupper()`	Returns `True` if string is all uppercase

Formative Assessment 1

Formative Assessment 2