Strings, Part 1

can code communities

String

Strings are how we store text in python.

Strings are actually a sequence of characters.

Strings are immutable - this means that if we want to change a string we have to completely remake the string. (More on this next class)

String Operators 1

+ For strings this is called **concatenation** and will put the second string on the end of the first.

* For strings this is a type of multiplication: it takes the string and repeats it the following number of times.

For these operations, data types are important:

- Concatenation can only happen between 2 or more strings.
- String multiplication can only occur between a string and a number.

String Operators 2

in - Returns True if a string appears inside another string (as a **substring**), and False otherwise.

- `"e" in "hello"` is True
- `"a" in "hello"` is False

len() - Returns the length of a string, aka the number of characters in a string.

- `len("hello")` is 5

String Methods 1

`capitalize()` Converts the first character to upper case

`casefold()` Converts string into lower case

`center()` Returns a centered string

`count()` Returns the number of times a specified value occurs in a string

`expandtabs()` Sets the tab size of the string

`find()` Searches the string for a specified value and returns the position of where it was found

`index()` Searches the string for a specified value and returns the position of where it was found

String Methods 2

`isalnum()` Returns True if all characters in the string are alphanumeric

`isalpha()` Returns True if all characters in the string are in the alphabet

`isdecimal()` Returns True if all characters in the string are decimals

`isdigit()` Returns True if all characters in the string are digits

`islower()` Returns True if all characters in the string are lower case

`isupper()` Returns True if all characters in the string are upper case

`isnumeric()` Returns True if all characters in the string are numeric

`isspace()` Returns True if all characters in the string are whitespaces

`istitle()` Returns True if the string follows the rules of a title

String Methods 3

`join()` Joins the elements of an iterable to the end of the string

`lower()` Converts a string into lower case

`partition()` Returns a tuple where the string is partitioned into three parts

`replace()` Returns a string where a specified value is replaced with a specified value

`split()` Splits the string at the specified separator, and returns a list

`splitlines()` Splits the string at line breaks and returns a list

`startswith()` Returns true if the string starts with the specified value

`strip()` Returns a trimmed version of the string

You can find a list of all the string methods on w3schools:

https://www.w3schools.com/python/python_ref_string.asp

Reference vs Value Equality: **is**

== Returns True if two objects have the same value

is Returns True if two objects have the same reference - in other words they are the same object

Usually we want to use == instead of is, for most simple code.

Discussion: What is the output of this code?

```
x = "hello"
str2 = "HELLO".lower()
print(str2 == x)
print(str2 is x)
```

User Input

You can take input from the user using the input() function.

User input is always stored as a **string**.

If you want to change the input to a different data type, you have to cast it to that type.

Input Sanitization

When you're taking input from the user, it's a good idea to sanitize it to make sure no errors occur.

One example of this is the strip() function, which removes white space on both sides of the input.

This could also include casting the user input to a different data type if that's the goal of your program, or converting it to lower case.

Many of the string methods can be used to sanitize user input.

It's usually a good idea to assume the user is trying to break your program, and write code to prepare for the worst case scenario.

Example

Write some code that will take a string from the user and print if it is a number or not.

Examples:

```
apple
False

4
True
```

Exercise

Take a word from the user. Then take a number from the user. Then print whether the word is longer than the number.

Examples:

```
apple
6
False

python
4
True
```

Hint: use `len()` to find the length of the string, and don't forget to cast to `int()`

Exercise

Write some code that takes a string from the user, and prints how many vowels are in the string. (Vowels are a, e, i, o, u)

- How will you count both uppercase and lowercase vowels?
- What string method can you use to count the number of vowels?

Example

User input: `Computer`

```
3
```

Printing Strings

If you want to print multiple strings, you can separate them with concatenation (+) or commas (,)

- If you use commas, you can print things that aren't strings, and everything is separated by a space
- If you use concatenation, everything has to be a string (if you're trying to print other data types you must cast them), and there are no spaces

Exercise - Printing

You have a variable called `hours` which equals `24`, the number of hours in a day.

Print `There are 24 hours in a day.` Make sure to use your variable.

- First, print using commas. Remember that using commas automatically adds spaces!
- Now, print using string concatenation. Remember to cast `hours` to a string and manually add the spaces!

Exercise - Challenge

Write some code that will print a box around a string.

Examples:

User input: `hello`

```
*******
*hello*
*******
```

User input: `programming is fun`

```
**********************
*programming is fun*
**********************
```

Hint: You will have to use the len() function, string concatenation (+), and string multiplication (*)

Resources

https://docs.python.org/3/library/string.html

https://www.w3schools.com/python/python_strings.asp

https://realpython.com/python-strings/