Strings

This lesson highlights the key features of the string data type.

We'll cover the following

At the start of the course, we learned how to print "**Hello World**" on the terminal.

A group of characters such as this is an example of the string data type.

> A string is a collection of characters closed within single or double quotation marks.

A string can also contain a single character or be entirely empty.

```
print("Harry Potter!")  # Double quotation marks

got = 'Game of Thrones...'  # Single quotation marks
print(got)
print("$")  # Single character

empty = ""
print(empty)  # Just prints an empty line
```

From the examples above we can see that a blank space inside the string quotation marks is also considered to be a character.

The Length of a String

The length of a string can be found using the `len` statement. This length indicates

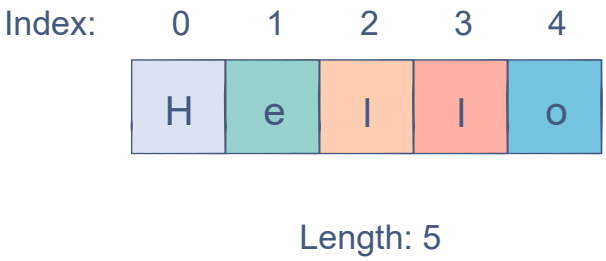
the number of characters in the string:

```
random_string = "I am Batman"  # 11 characters
print(len(random_string))
```

Indexing

In a string, every character is given a numerical **index** based on its position.

A string in Python is indexed from `0` to `n-1` where `n` is its length. This means that the index of the first character in a string is `0`.



Accessing Characters

Each character in a string can be accessed using its index. The index must be closed within square brackets, `[]`, and appended to the string.

```
batman = "Bruce Wayne"

first = batman[0]  # Accessing the first character
print(first)

space = batman[5]  # Accessing the empty space in the string
print(space)

last = batman[len(batman) - 1]
print(last)
# The following will produce an error since the index is out of bounds
# err = batman[len(batman)]
```

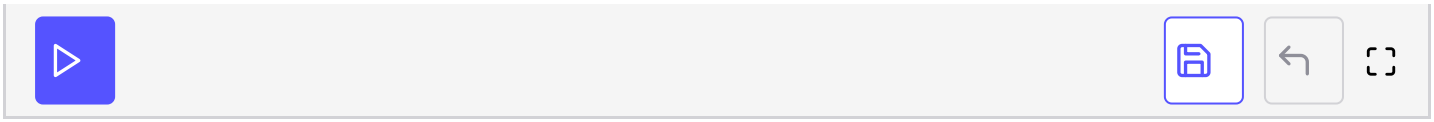
If we try to execute the code on **line 12**, we would get an error because the maximum index is `len(batman) - 1`. A higher value is not within the bounds of the string. Since `len(batman)` is larger than `len(batman) - 1`, it will produce an error.

Reverse Indexing

We can also change our indexing convention by using negative indices.

Negative indices start from the opposite end of the string. Hence, the `-1` index corresponds to the last character:

```
batman = "Bruce Wayne"
print(batman[-1])  # Corresponds to batman[10]
print(batman[-5])  # Corresponds to batman[6]
```



In the next lesson, we'll understand the concept of **string slicing**.