

STRING OPERATIONS

An Introduction to Computer Science



Let's learn about String Operations.

Addition

```
"Hello " + "World"
```



```
"Hello World"
```



Like numbers, you can add two strings together.
This puts them side by side in a single new string.
This is sometimes called "Concatenation".

Comparing Strings

```
"Dog" == "Dog"
```

```
"Dog" != "Cat"
```

```
"Aardvark" < "Zoo"
```



You can test if two strings are equal, not equal, or even compare them using less than and greater than.

This measures which ones come first in the alphabet.

Membership in Strings

```
"house" in "boathouse"
```

```
"e" in "elephant"
```

```
"y" not in "axes"
```

```
"T" not in "Carrot"
```

Capitalization
Matters!



There's another test you can check with Strings: using the "in" operator. This simply checks whether the first string appears in the second. You can also use the "not in" operator to test the opposite.

Subscripting

```
"Hello world"[0]
```



```
"H"
```



Subscripting is one of the more powerful and more complex features of strings. When we **subscript** a string, we extract one or more characters from the string.

Subscript Syntax

Value

`"Harry Potter"[0]`

Opening
Bracket

Closing
Bracket

Variable

`book_title = "Harry Potter"`

`book_title[0]`

Index



We can subscript a string value or variable by using square brackets.

Notice the key components: On the left is the name of the variable or the string literal value.

Next we have an opening square bracket.

Then, we have a number, which is called the index.

Finally, we end with a closing square bracket.

Starting at 0

"Harry Potter"

Index	Character
0	H
1	a
2	r
3	r
4	y
5	
6	P
7	o
8	t
9	t
10	e
11	r



Here's a weird thing: computers start counting at 0, not 1.

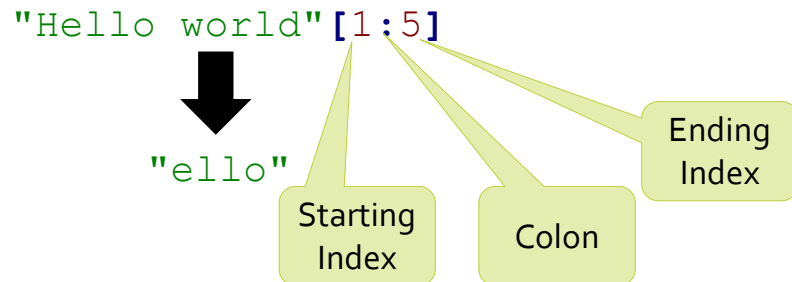
So if you want the first character from the string, you need to write 0 instead of 1.

There are mathematical reasons why computers work this way, but ultimately it ends up being more convenient, once you get used to it.

But you can see in this string that, starting from 0, every character (including the space) is assigned an index.

I can get out the second "r" by using index 3, and the capital P by using index 6.

Subscripting Multiple Characters



It wouldn't be too useful to only grab out one character at a time.
So you can actually grab out more than one by using the subscript range syntax.
Inside the square brackets, you put a pair of numbers separated by a colon.
The first number is the starting index, and the second number is the closing index.

Negative Subscript Indices

"Hamster"[-1]	→	"r"
"Hamster"[1:-1]	→	"amste"
"Hamster"[-3:]	→	"ter"
"Hamster"[:3]	→	"ham"



If you use negative numbers as subscript indexes, you can work from the back of the list.

If you use -1, then you get the last character.

You can combine positive and negative numbers in your indexes.

To go from the start or the end, simply leave the number missing.