

# Python String

“ABC”

# Python String

A Python string is a sequence of characters that can be written in three ways.

single quotes

‘Python’

Double quotes

“Python”

Triple quotes

““Python””



Python  
String

Its coding time...

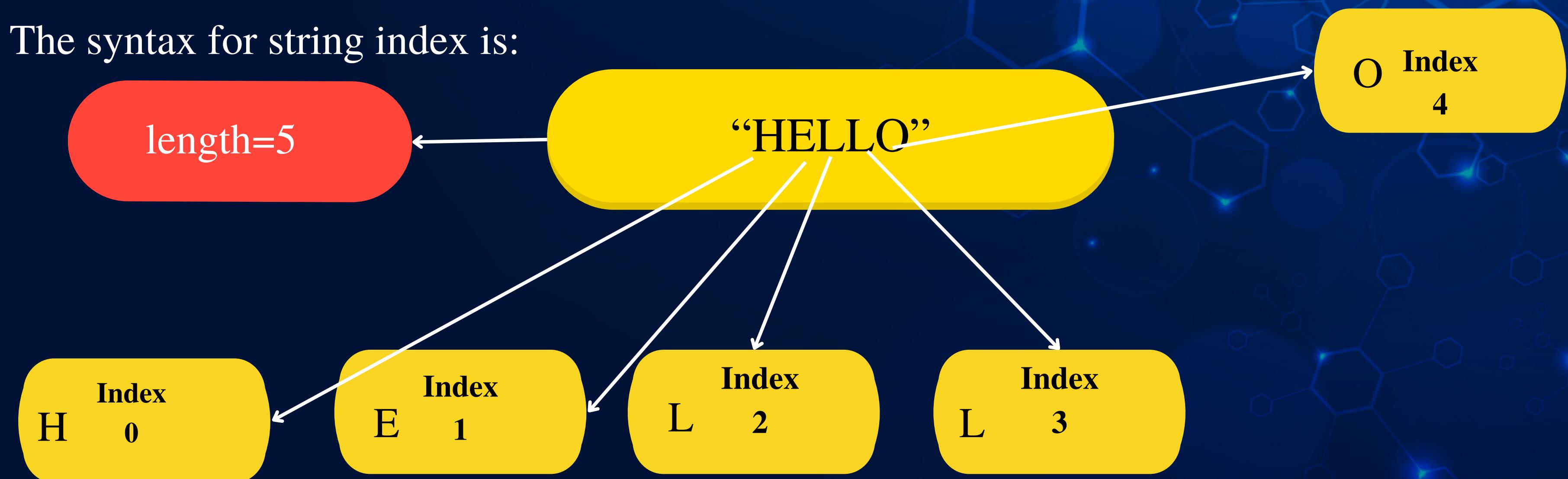


# String indexing

In Python, indexing refers to accessing individual elements or subsets of elements within a data structure like a list, tuple, string, or any other iterable object. Indexing allows you to retrieve specific items based on their position or index within the sequence.

The indexing in Python starts from 0, meaning the first element of a sequence has an index of 0, the second element has an index of 1, and so on.

The syntax for string index is:



Python  
String

Its coding time...



# String slicing

*String slicing in Python refers to extracting a substring (a portion of a string) from a given string. It allows you to specify a range of indices and obtain the substring that falls within that range.*

The syntax for string slicing is:

String [Start:Stop:Step]

stop: The index where the slicing ends. It indicates the position of the character just after the last character to include in the sliced substring. If omitted, slicing goes until the end of the string.

step (optional): The step size used to increment the index while slicing. It determines the stride or how many characters to skip between successive characters in the sliced substring. The default value is 1.

start: The index from which the slicing starts. It indicates the position of the first character to include in the sliced substring. If omitted, slicing starts from the beginning of the string (index 0).

Python  
String

Its coding time...



# Negative Indexing

*Negative indexing in strings is a method of accessing individual characters of a string by counting from the end of the string.*

The syntax for string slicing is:

String [Start:Stop:Step]

Example

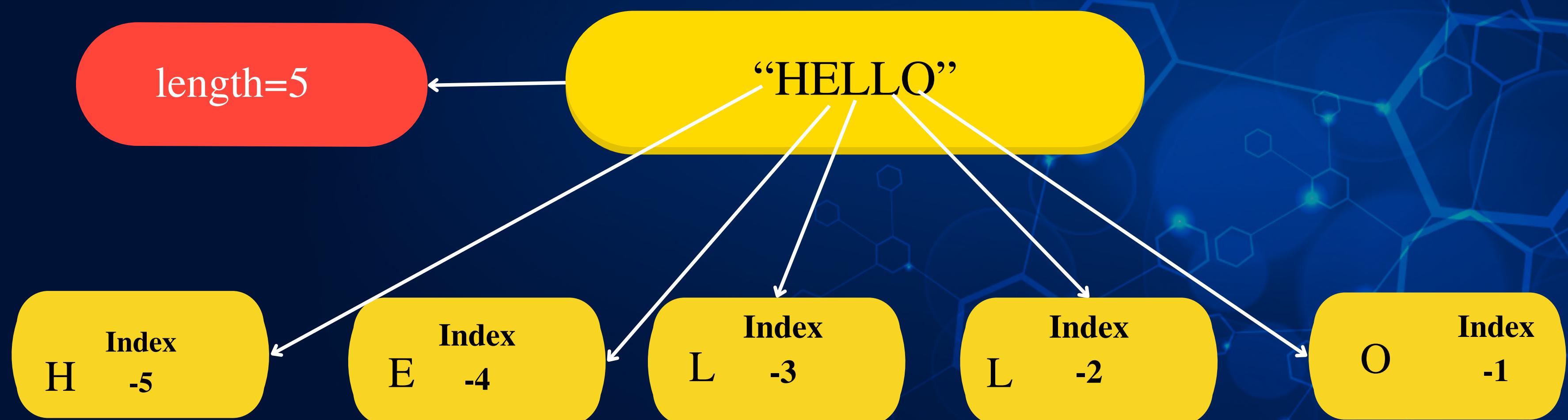
String [-4:-1:1]

Start of negative index

where to stop

Step to jump

# Example negative indexing



Python  
String

Its coding time...



# String Built-in Methods



# String Inbuilt Method...

`string.capitalize()`

Converts the first character to upper case

`String.upper()`

Converts a string into upper case

`String.index()`

returns the position of specified value where it was found

`string.swapcase()`

Swaps cases, lower case becomes upper case and vice versa

`string.lower()`

Converts a string into lower case

`string.replace(f,r)`

specified value is replaced with a specified value

`string.swapcase()`

Swaps cases, lower case becomes upper case and vice versa

# Join() Inbuilt Method...

The join() method in Python is used to concatenate elements of an iterable (such as a list, tuple, or set) into a single string. It concatenates each element of the iterable with a specified separator string and returns the concatenated string.

Here's the syntax of the join() method:

**string.join(iterable)**

where string is the separator string and iterable is the iterable containing the elements to be joined.

```
# Define a list of strings
my_list = ["Hello", "world", "this", "is", "Python"]

# Join the list elements into a single string with a space as delimiter
result = " ".join(my_list)

# Output the result
print(result)
```

# THANK YOU