Python String Slicing

To access a range of characters in a string, you need to slice a string. One way to do this is to use the simple slicing operator :

With this operator you can specify where to start the slicing, where to end and specify the step.

Slicing a String

If S is a string, the expression S [start : stop : step] returns the portion of the string from index start to index stop, at a step size step.

Syntax

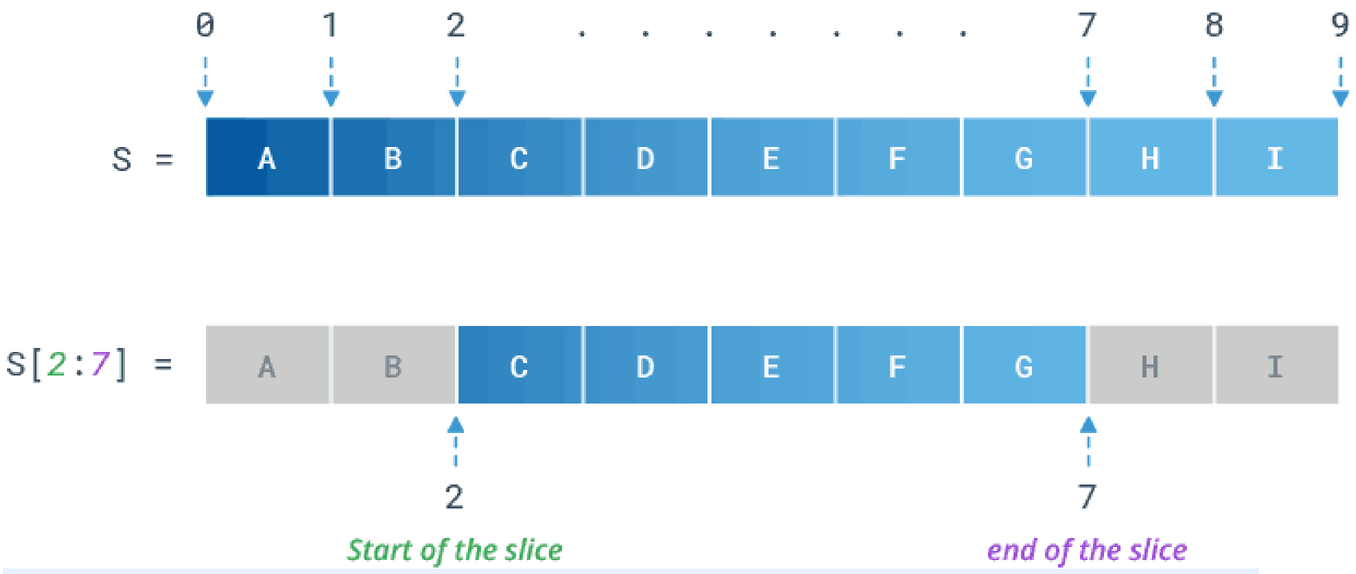
```
S[start:stop:step]
```

Start position   *End position*   *The increment*

Basic Example

Here is a basic example of string slicing.

```
S = ' A B C D E F G H I '
print(S[2:7])          #   C D E F G
```

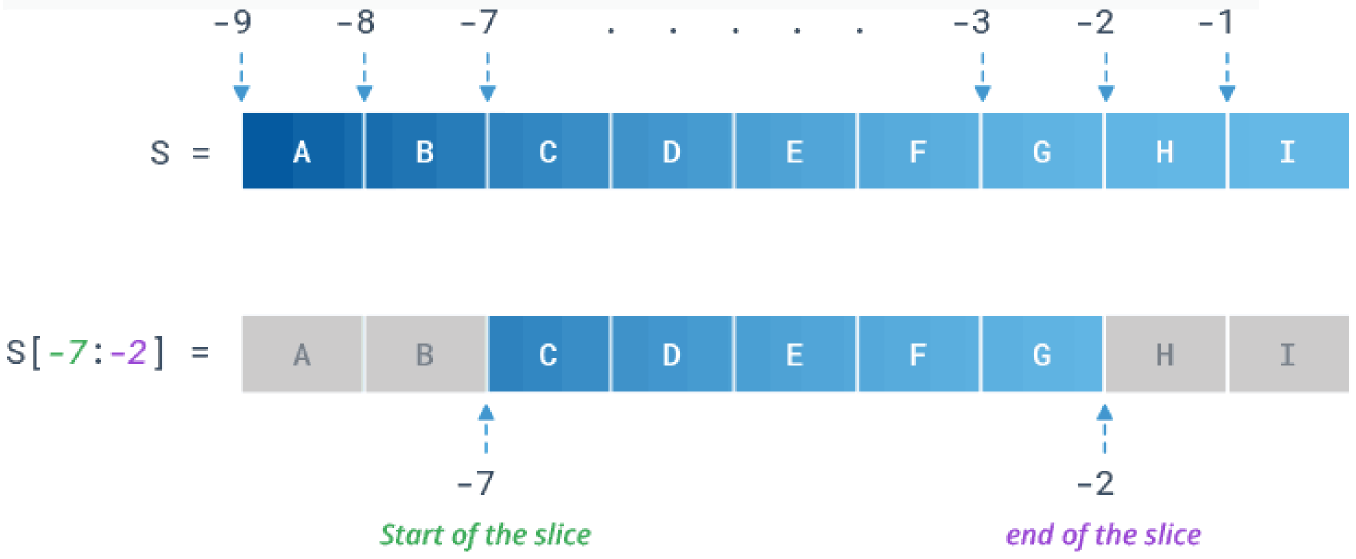


Note that the item at index 7 'H' is not included.

Slice with Negative Indices

You can also specify negative indices while slicing a string.

```
S = ' A B C D E F G H I '
print(S[-7:-2])      #   C D E F G
```



Slice with Positive & Negative Indices

You can specify both positive and negative indices at the same time.

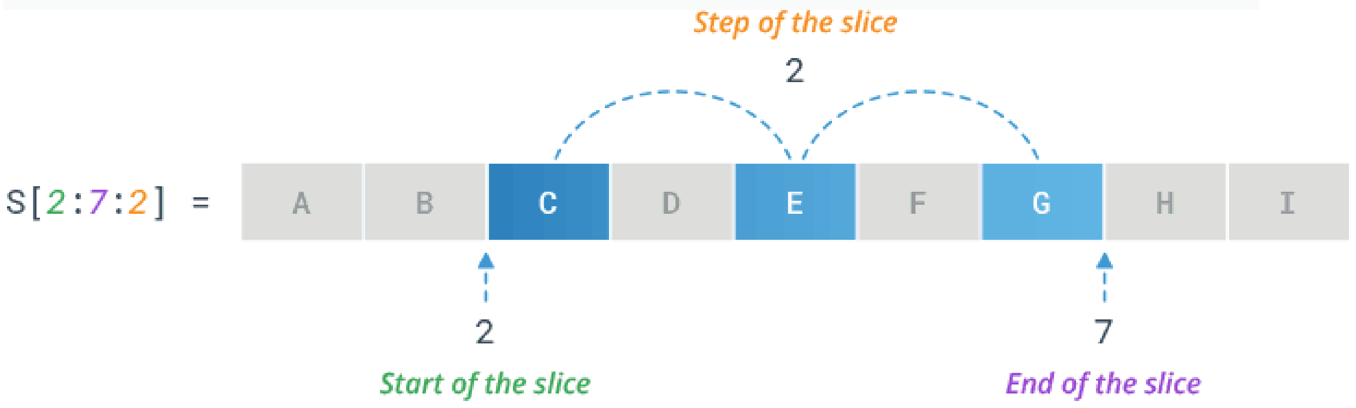
```
S = ' A B C D E F G H I '
print(S[2:-5])        #   C D
```

Specify Step of the Slicing

You can specify the step of the slicing using step parameter. The step parameter is optional and by default 1.

```
# Return every 2nd item between position 2 to 7

S = ' A B C D E F G H I '
print(S[2:7:2])       #   C E G
```



Negative Step Size

You can even specify a negative step size.

```
# Returns every 2nd item between position 6 to 1 in reverse order

S = ' A B C D E F G H I '
print(S[6:1:-2])   #   G E C
```

Slice at Beginning & End

Omitting the start index starts the slice from the index 0. Meaning, S[:stop] is equivalent to S[0:stop]

```
# Slice first three characters from the string

S = ' A B C D E F G H I '
```

```
print(S[:3])    #    A B C
```

Whereas, omitting the `stop` index extends the slice to the end of the string. Meaning, `S[start:]` is equivalent to `S[start:len(S)]`

```
# Slice last three characters from the string

S = ' A B C D E F G H I '
print(S[6:])    #    G H I
```

Reverse a String

You can reverse a string by omitting both `start` and `stop` indices and specifying a `step` as -1.

```
S = ' A B C D E F G H I '
print(S[::-1])    #    I H G F E D C B A
```