

# String Functions

There also exist algebraic operators (+, -, \*, /) to perform string concatenation and deletion.

## Example

```
SELECT "Hello"+"World" FROM Stream
=> "HelloWorld"
SELECT "HelloWorld"-"World" FROM Stream
=> "Hello"
SELECT "HelloWorld"*3 FROM Stream
=> "HelloWorldHelloWorldHelloWorld"
SELECT "HelloWorldHelloWorldHelloWorld"/"World" FROM Stream
=> 3
```

- `Concat(String s1, String s2)`
- `SubString(String s, Number begin, Number end)`
- `SubString(String s, Number begin)`
- `Length(String s)`
- `Upper(String s)`
- `Lower(String s)`
- **Parsing of Strings**
  - `split(String s, String delimiter, Number index)`

## Concat(String s1, String s2)

Returns a new string that is a concatenating the arguments.

## SubString(String s, Number begin, Number end)

Returns a new string that is a substring of the value with given begin and end index

## Example

```
SELECT substring("Hello World",1,3) FROM Stream
=> "el"
```

## SubString(String s, Number begin)

Returns a new string that is a substring of the value starting at the begin index until the end of the string

## Example

```
SELECT substring("Hello World",1) FROM Stream
=> "ello World"
```

## Length(String s)

Returns the length of a string.

## Example

```
SELECT length("Hello") FROM Stream
=> 5
```

## Upper(String s)

Returns the string converted to uppercase.

## Example

```
SELECT upper("Hello World") FROM Stream
=> "HELLO WORLD"
```

## Lower(String s)

Returns the string converted to lowercase.

### Example

```
SELECT lower("Hello World") FROM Stream  
=> "hello world"
```

## Parsing of Strings

The most functions provide one return value. Sometimes it is needed to split an incoming attribute to multiple values

### split(String s, String delimiter, Number index)

This function returns number (third parameter) strings from the input string (first parameter) with the delimiter (second parameter).

```
SELECT split(input,",",3) FROM stream
```

Remark: There is no way to parse different types with this function. Each incoming string must contains exactly the given numbers of inputs.