

STRINGS

**EXAMPLE 29:** STRINGS AND NUMBERS -BUT NOT  
OBJECTS - ARE 'PASSED-BY-VALUE' TO FUNCTIONS

RECAP

**EXAMPLE 29:** STRINGS AND NUMBERS -BUT NOT  
OBJECTS - ARE 'PASSED-BY-VALUE' TO FUNCTIONS

IN OTHER WORDS, A COPY IS MADE  
OF STRING OR NUMBER VARIABLES

AND THE FUNCTION WORKS WITH  
THE COPY - NOT THE ORIGINAL.

RECAP

## EXAMPLE 29: STRINGS AND NUMBERS -BUT NOT OBJECTS - ARE 'PASSED-BY-VALUE' TO FUNCTIONS

IN OTHER WORDS, A COPY IS MADE  
OF STRING OR NUMBER VARIABLES

AND THE FUNCTION WORKS WITH  
THE COPY - NOT THE ORIGINAL.

EVEN IF THE FUNCTION MODIFIES THE  
VARIABLE, THE ORIGINAL IS UNCHANGED.

RECAP

**EXAMPLE 30: OBJECTS AND ARRAYS ARE  
'PASSED-BY-REFERENCE' TO FUNCTIONS**

**RECAP**

# EXAMPLE 30: OBJECTS AND ARRAYS ARE 'PASSED-BY-REFERENCE' TO FUNCTIONS

OBJECTS AND ARRAYS - UNLIKE NUMBERS OR  
STRINGS - ARE PASSED IN "AS-IS" TO FUNCTIONS

SO THE FUNCTION WORKS WITH  
THE ORIGINAL - NOT A COPY

RECAP



# EXAMPLE 30: OBJECTS AND ARRAYS ARE 'PASSED-BY-REFERENCE' TO FUNCTIONS

OBJECTS AND ARRAYS - UNLIKE NUMBERS OR  
STRINGS - ARE PASSED IN "AS-IS" TO FUNCTIONS

SO THE FUNCTION WORKS WITH  
THE ORIGINAL - NOT A COPY

IF THE FUNCTION MODIFIES THE VARIABLE,  
THE ORIGINAL IS INDEED CHANGED!

RECAP

# STRINGS

IN JAVASCRIPT, STRINGS ARE OBJECTS,  
BUT THEY BEHAVE LIKE PRIMITIVE TYPES.



**IN JAVASCRIPT, STRINGS ARE OBJECTS,  
BUT THEY BEHAVE LIKE PRIMITIVE TYPES.**

**STRINGS ARE OBJECTS - THEY HAVE A HOST OF  
HELPFUL PROPERTIES THAT WE USE ALL THE TIME**

IN JAVASCRIPT, STRINGS ARE OBJECTS,  
**BUT THEY BEHAVE LIKE PRIMITIVE TYPES.**

**FOR INSTANCE, STRINGS "SEEM" TO BE PASSED BY  
VALUE - CHANGING A STRING INSIDE A FUNCTION  
DOES NOT CHANGE ITS VALUE**

FOR INSTANCE, STRINGS “SEEM” TO BE PASSED BY  
VALUE - CHANGING A STRING INSIDE A FUNCTION  
DOES NOT CHANGE ITS VALUE

IN JAVASCRIPT, STRINGS ARE OBJECTS,  
BUT THEY BEHAVE LIKE PRIMITIVE TYPES.

THIS IS BECAUSE STRINGS ARE “IMMUTABLE  
OBJECTS” (LIKE IN JAVA)

FOR INSTANCE, STRINGS “SEEM”  
TO BE PASSED BY VALUE

THIS IS BECAUSE STRINGS ARE “IMMUTABLE  
OBJECTS” (LIKE IN JAVA)

YOU CAN NEVER CHANGE A STRING, ONLY  
CREATE A NEW ONE.

YOU CAN NEVER CHANGE A STRING, ONLY  
CREATE A NEW ONE.

ANY OPERATION YOU PERFORM THAT  
SEEMS TO CHANGE A STRING - CREATES  
A NEW STRING!



# FOR INSTANCE, STRINGS “SEEM” TO BE PASSED BY VALUE

THIS IS BECAUSE STRINGS ARE  
“IMMUTABLE OBJECTS” (LIKE IN JAVA)

YOU CAN NEVER CHANGE A STRING, ONLY  
CREATE A NEW ONE.

SO EVEN THOUGH STRINGS ARE PASSED BY  
REFERENCE, THEY CAN NEVER BE MODIFIED

(AND REASSIGNMENTS OF OBJECTS DON'T REFLECT ANYWAY!)

## EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT



# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";
var lastName = 'Srinivasan';
// concatenation
var fullName = firstName + " " + lastName;
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +
fullName);
// find substring position
var positionOfLastName = fullName.indexOf(lastName);
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +
positionOfLastName);

var nickName = "Humpty Dumpty";
var nameParts = fullName.split(" ");
console.log("Fullname split into words (delimited by space) " + nameParts);
fullName = nickName + " " + nameParts[1];
console.log("Reconstructed fullname = " + fullName);
```

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";
var lastName = 'Srinivasan';
// concatenation
var fullName = firstName + " " + lastName;
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +
fullName);
// find substring position
var positionOfLastName = fullName.indexOf(lastName);
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +
positionOfLastName);

var nickName = "Humpty Dumpty";
var nameParts = fullName.split(" ");
console.log("Fullname split into words (delimited by space) " + nameParts);
fullName = nickName + " " + nameParts[1];
console.log("Reconstructed fullname = " + fullName);
```

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vittthal";  
var lastName = 'Srinivasan';
```

**IN JAVASCRIPT, SINGLE AND DOUBLE QUOTES CAN BOTH BE USED WITH STRINGS**

```
// concatenation  
var fullName = firstName + " " + lastName;  
console.log("Combine firstName " + firstName + " and LastName " + lastName + "\n" +  
fullName);  
// find substring position  
var positionOfLastName = fullName.indexOf(lastName);  
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +  
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";  
var nameParts = fullName.split(" ");  
console.log("Fullname split into words (delimited by space) " + nameParts);  
fullName = nickName + " " + nameParts[1];  
console.log("Reconstructed fullname = " + fullName);
```

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";  
var lastName = 'Srinivasan';
```

```
// concatenation
```

```
var fullName = firstName + " " + lastName;
```

```
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +  
fullName);
```

```
// find substring position
```

```
var positionOfLastName = fullName.indexOf(lastName);
```

```
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +  
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";
```

```
var nameParts = fullName.split(" ");
```

```
console.log("Fullname split into words (delimited by space) " + nameParts);
```

```
fullName = nickName + " " + nameParts[1];
```

```
console.log("Reconstructed fullname = " + fullName);
```

**SIMPLY USE THE + OPERATOR TO  
CONCATENATE STRINGS**

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName + " " + lastName;  
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +  
fullName);
```

```
// find substring position
```

```
var positionOfLastName = fullName.indexOf(lastName);
```

```
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +  
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";  
var nameParts = fullName.split(" ");  
console.log("Fullname split into words (delimited by space) " + nameParts);  
fullName = nickName + " " + nameParts[1];  
console.log("Reconstructed fullname = " + fullName);
```

## FIND THE POSITION OF A STRING INSIDE ANOTHER STRING



# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName + " " + lastName;  
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +  
fullName);
```

**// find substring position**

```
var positionOfLastName = fullName.indexOf(lastName);
```

```
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +  
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";  
var nameParts = fullName.split(" ");  
console.log("Fullname split into words (delimited by space) " + nameParts);  
fullName = nickName + " " + nameParts[1];  
console.log("Reconstructed fullname = " + fullName);
```

## FIND THE POSITION OF A STRING INSIDE ANOTHER STRING

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName + " " + lastName;  
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +  
fullName);
```

**// find substring position**

```
var positionOfLastName = fullName.indexOf(lastName);
```

```
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +  
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";  
var nameParts = fullName.split(" ");  
console.log("Fullname split into words (delimited by space) " + nameParts);  
fullName = nickName + " " + nameParts[1];  
console.log("Reconstructed fullname = " + fullName);
```

## FIND THE POSITION OF A STRING INSIDE ANOTHER STRING



# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName + " " + lastName;  
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +  
fullName);
```

**// find substring position**

```
var positionOfLastName = fullName.indexOf(lastName);
```

```
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +  
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";  
var nameParts = fullName.split(" ");  
console.log("Fullname split into words (delimited by space) " + nameParts);  
fullName = nickName + " " + nameParts[1];  
console.log("Reconstructed fullname = " + fullName);
```

## FIND THE POSITION OF A STRING INSIDE ANOTHER STRING

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName + " " + lastName;  
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +  
fullName);
```

**// find substring position**

```
var positionOfLastName = fullName.indexOf(lastName);
```

```
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +  
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";  
var nameParts = fullName.split(" ");  
console.log("Fullname split into words (delimited by space) " + nameParts);  
fullName = nickName + " " + nameParts[1];  
console.log("Reconstructed fullname = " + fullName);
```

## FIND THE POSITION OF A STRING INSIDE ANOTHER STRING

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";
var lastName = 'Srinivasan';
// concatenation
var fullName = firstName + " " + lastName;
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +
fullName);
// find substring position
var positionOfLastName = fullName.indexOf(lastName);
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";
```

```
var nameParts = fullName.split(" ");
```

```
console.log("Fullname split into words (delimited by space) " + nameParts);
```

```
fullName = nickName + " " + nameParts[1];
```

```
console.log("Reconstructed fullName = " + fullName);
```

**SPLIT A STRING INTO AN ARRAY OF SUBSTRINGS  
(AND SPECIFY ANY DELIMITER YOU NEED!)**

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";
var lastName = 'Srinivasan';
// concatenation
var fullName = firstName + " " + lastName;
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +
fullName);
// find substring position
var positionOfLastName = fullName.indexOf(lastName);
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";
```

```
var nameParts = fullName.split(" ");
```

```
console.log("Fullname split into words (delimited by space) " + nameParts);
```

```
fullName = nickName + " " + nameParts[1];
```

```
console.log("Reconstructed fullName = " + fullName);
```

**SPLIT A STRING INTO AN ARRAY OF SUBSTRINGS  
(AND SPECIFY ANY DELIMITER YOU NEED!)**

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";
var lastName = 'Srinivasan';
// concatenation
var fullName = firstName + " " + lastName;
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +
fullName);
// find substring position
var positionOfLastName = fullName.indexOf(lastName);
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +
positionOfLastName);
```

```
var nickName = "Humpty Dumpty";
```

```
var nameParts = fullName.split(" ");
```

```
console.log("Fullname split into words (delimited by space) " + nameParts);
```

```
fullName = nickName + " " + nameParts[1]
```

```
console.log("Reconstructed fullName = " + fullName);
```

**SPLIT A STRING INTO AN ARRAY OF SUBSTRINGS  
(AND SPECIFY ANY DELIMITER YOU NEED!)**



# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName = "Vitthal";
var lastName = 'Srinivasan';
// concatenation
var fullName = firstName + " " + lastName;
console.log("Combine firstName " + firstName + " and lastName " + lastName + " into fullName = " +
fullName);
// find substring position
var positionOfLastName = fullName.indexOf(lastName);
console.log("Inside " + fullName + " the string " + lastName + " occurs at position " +
positionOfLastName);

var nickName = "Humpty Dumpty";
var nameParts = fullName.split(" ");
console.log("Fullname split into words (delimited by space) " + nameParts);
fullName = nickName + " " + nameParts[1];
console.log("Reconstructed fullname = " + fullName);
```

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName =  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName +  
console.log(
```

Combine firstName Vitthal and lastName Srinivasan into fullName = Vitthal Srinivasan

```
// find  
var position = fullName.indexOf('Srinivasan');  
console.log('Inside Vitthal Srinivasan the string Srinivasan occurs at position ' + position);  
var words = fullName.split(' ');  
console.log('Fullname split into words (delimited by space) ' + words);  
var reconstructedFullName = words[0] + ' Humpty Dumpty ' + words[1];  
console.log('Reconstructed fullname = ' + reconstructedFullName);
```

```
var nickName =  
var nameParts = fullName.split(' ');  
console.log('Name parts: ' + nameParts);  
fullName = nickName + ' ' + nameParts[1];  
console.log('New full name: ' + fullName);
```



# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName =  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName +  
console.log(
```

fullName Combine firstName Vitthal and lastName Srinivasan into fullName = Vitthal Srinivasan

```
// find  
var position = fullName.indexOf('Srinivasan');  
console.log(position);
```

Inside Vitthal Srinivasan the string Srinivasan occurs at position 8

```
var words = fullName.split(' ');  
console.log(words);
```

Fullname split into words (delimited by space) Vitthal,Srinivasan

position Reconstructed fullname = Humpty Dumpty Srinivasan

```
var nickName = 'Humpty';  
var nameParts = fullName.split(' ');  
console.log(nameParts[0] + nickName);  
fullName = nickName + nameParts[1];  
console.log(fullName);
```

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName =  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName +  
console.log(
```

fullName Combine firstName Vitthal and lastName Srinivasan into fullName = Vitthal Srinivasan

```
// find  
var position = fullName.indexOf('Srinivasan');  
console.log('Inside Vitthal Srinivasan the string Srinivasan occurs at position ' + position);
```

```
var words = fullName.split(' ');  
console.log('Fullname split into words (delimited by space) ' + words.join(','));
```

position Reconstructed fullname = Humpty Dumpty Srinivasan

```
var nickName = 'Humpty';  
var nameParts = fullName.split(' ');  
console.log('Reconstructed fullname = ' + nameParts[0] + ' ' + nameParts[1] + ' ' + nickName);  
fullName = nickName + ' ' + nameParts[0] + ' ' + nameParts[1];  
console.log('Reconstructed fullname = ' + fullName);
```

# EXAMPLE 37: SIMPLE STRING OPERATIONS IN JAVASCRIPT

```
var firstName =  
var lastName = 'Srinivasan';  
// concatenation  
var fullName = firstName +  
console.log(
```

fullName	Combine firstName Vitthal and lastName Srinivasan into fullName = Vitthal Srinivasan
// fin	Inside Vitthal Srinivasan the string Srinivasan occurs at position 8
var po	Fullname split into words (delimited by space) Vitthal,Srinivasan
console	
position	Reconstructed fullname = Humpty Dumpty Srinivasan

```
var nickName =  
var nameParts = fullName.split(  
console.log(  
fullName = nickName +  
console.log(
```