Web Design & Development (JavaScript-2)

Class-15 (28/1/21)

Jibitesh Mishra

JavaScript data types

- String data types are written within single quotes or double quotes
- Numeric data types are just numbers
- javaScript types are dynamic
- Same variable can be used to hold different data types

JavaScript array

- Javascript arrays are written in square bracket
- Array items are separated by commas
- Array indexes are zero based, which means first item is [0]
- For example :
  - var cars = ["Saab","Volvo","BMW"];

JavaScript objects

- JavaScript objects are written with curly braces {}
- Object properties are written as:
  - Name-value pairs separated by commas

For example:

```
var person = firstName : "John", lastName  : "Doe", age     : 50, eyeColor  : "blue"
```

JavaScript data types

- Undefined can be one datatype.
- Object declared as undefined has datatype as undefined
- Object declared as null has datatype object, its value is null but type is still an object
- However, object declared as undefined though has value null, but the type is undefined

Primitive datatypes

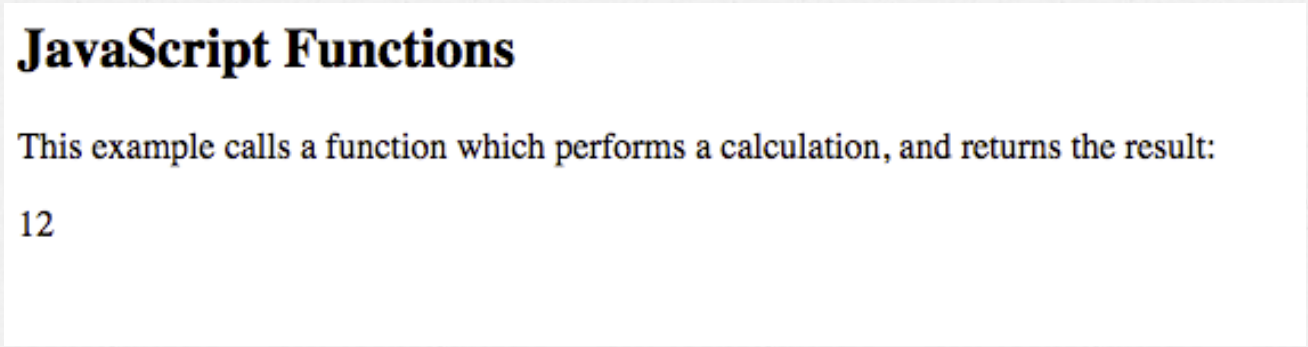
- The 4 primitive data types are :
  - String
  - Number
  - Boolean
  - Undefined

  Other two datatypes are function and object

Function example

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Functions</h2>
<p>This example calls a function which performs a calculation, and returns the result:</p>
<p id="demo"></p>
<script>
function myFunction(p1, p2) {
  return p1 * p2;
}
document.getElementById("demo").innerHTML = myFunction(4, 3);
</script>
</body>
</html>
```

output



JavaScript Functions

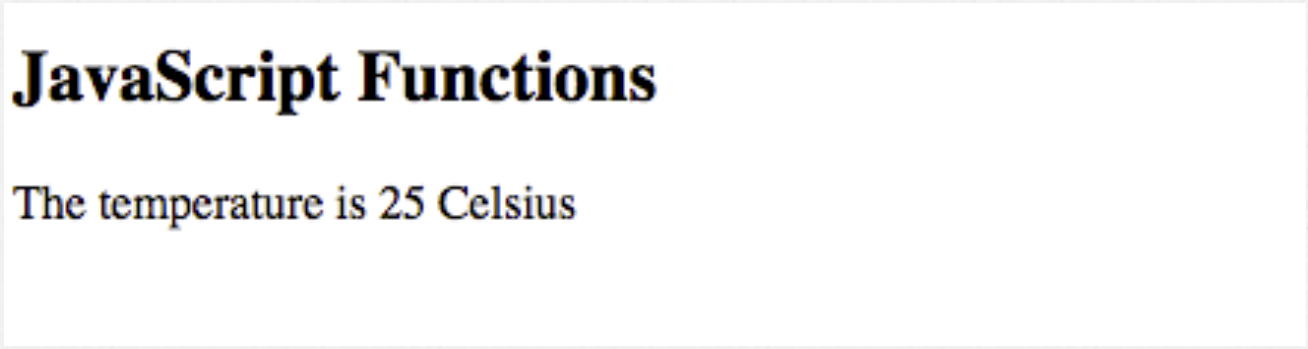
This example calls a function which performs a calculation, and returns the result:

12

Reusable function

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Functions</h2>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML =
"The temperature is " + toCelsius(77) + " Celsius”;
function toCelsius(fahrenheit) {
  return (5/9) * (fahrenheit-32);
}
</script>
</body>
</html>
```

output



Javascript object

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Objects</h2>
<p>An object method is a function definition, stored as a property value.</p>
<p id="demo"></p>
<script>
// Create an object:
var person = {
  firstName: "John",
  lastName : "Doe",
  id     : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
// Display data from the object:
document.getElementById("demo").innerHTML = person.fullName();
</script>
</body>
</html>
```

output

JavaScript Objects

An object method is a function definition, stored as a property value.

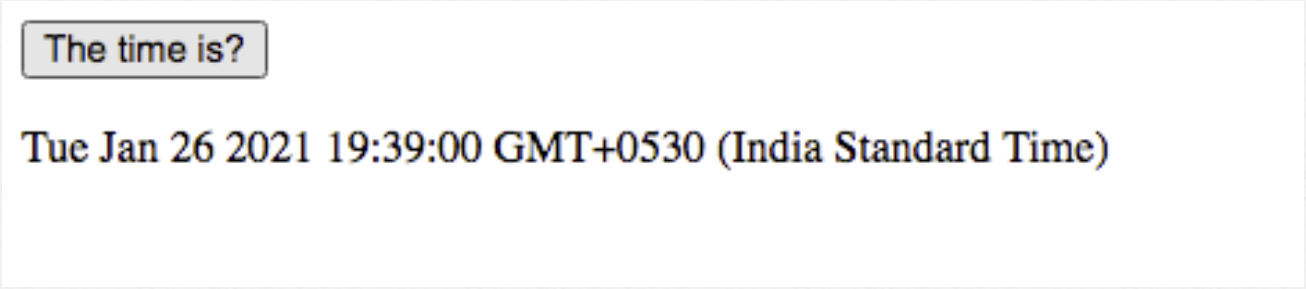
John Doe

events

```
<!DOCTYPE html>
<html>
<body>
<button
onclick="document.getElementById('demo').in
nerHTML=Date()">The time is?</button>
<p id="demo"></p>
</body>
</html>
```

output

On clicking


The time is?
Tue Jan 26 2021 19:39:00 GMT+0530 (India Standard Time)

Common HTML elements

- Onclick : user clicks an HTML event
- Onchange: an HTML element has been changed
- Onmouseover: the user moves the mouse over an HTML element
- Onmouseout: the user moves the mouse away from an HTML element
- Onload: the browser has finished loading a page
- Onkeydown: the user pushes the keyboard key

Length property

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript String Properties</h2>
<p>The length property returns the length of a string:</p>
<p id="demo"></p>
<script>
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
var sln = txt.length;
document.getElementById("demo").innerHTML = sln;
</script>
</body>
</html>
```

output

JavaScript String Properties

The length property returns the length of a string:

26

Escape sequence

- \b : backspace
- \f : form feed
- \n : new line
- \r : carriage return
- \t : horizontal tabulator
- \v : vertical tabulator

Inserting single quote

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Strings</h2>
<p>The escape sequence \' inserts a single quote in a string.</
p>
<p id="demo"></p>
<script>
var x = 'It\'s alright.';
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

output

JavaScript Strings

The escape sequence \' inserts a single quote in a string.

It's alright.

Breaking long code lines

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Strings</h2>
<p>
You can break a code line within a text string with a backslash.
</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "Hello \
Dolly!";
</script>
</body>
</html>
```

output

JavaScript Strings

You can break a code line within a text string with a backslash.

Hello Dolly!

Another way

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Strings</h2>
<p>The safest way to break a code line in a string is using string addition.</p>
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = "Hello " +
"Dolly!";
</script>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var x = "John";        // x is a string
var y = new String("John");  // y is an object
document.getElementById("demo").innerHTML =
typeof x + "<br>" + typeof y;
</script>
</body>
</html>
```

output

```
string
object
```

Example of boolean

```
<!DOCTYPE html>
<html>
<body>
<h2>Never Create Strings as objects.</h2>
<p>Strings and objects cannot be safely compared.</p>
<p id="demo"></p>
<script>
var x = "John";        // x is a string
var y = new String("John");  // y is an object
document.getElementById("demo").innerHTML = (x===y);
</script>
</body>
</html>
```

output

Never Create Strings as objects.

Strings and objects cannot be safely compared.

false

Even javascript obejcts can't be compaired

```
<!DOCTYPE html>
<html>
<body>
<h2>Never Create Strings as objects.</h2>
<p>JavaScript objects cannot be compared.</p>
<p id="demo"></p>
<script>
var x = new String("John");  // x is an object
var y = new String("John");  // y is an object
document.getElementById("demo").innerHTML = (x==y);
</script>
</body>
</html>
```

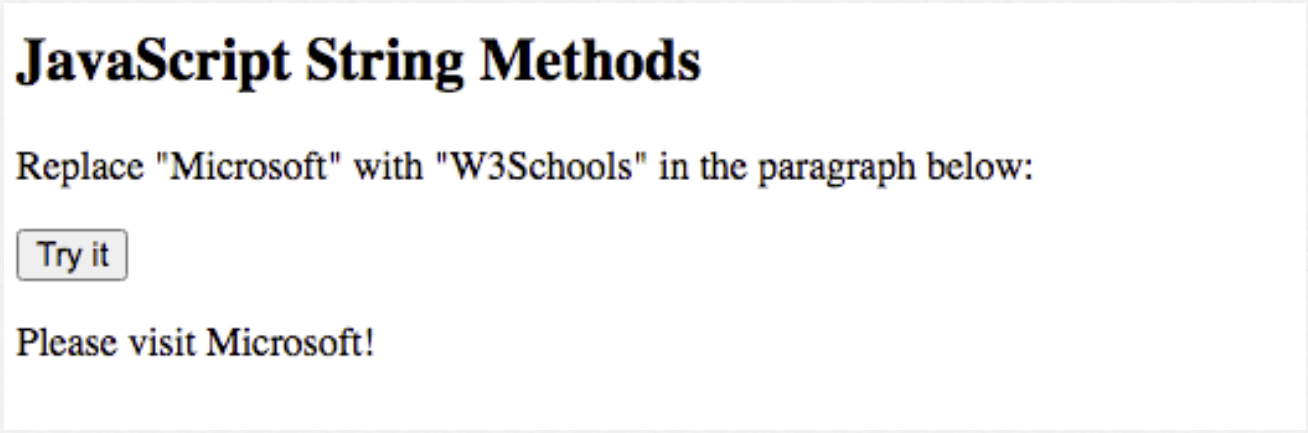
String methods

- Slice (start, end)
- Substring (start, end) is similar to slice, but without negative index
- Substr(start, length)
- If one omits the second parameter, substr() will slice out rest of the string
- If the first parameter is negative, the position counts from the end of the string

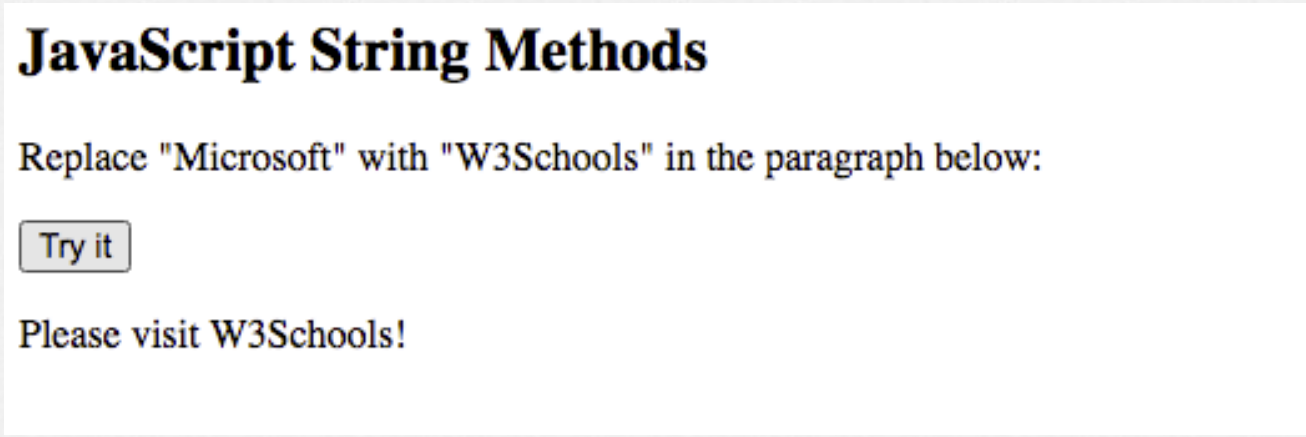
Replace method

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript String Methods</h2>
<p>Replace "Microsoft" with "W3Schools" in the paragraph below:</p>
<button onclick="myFunction()">Try it</button>
<p id="demo">Please visit Microsoft!</p>
<script>
function myFunction() {
  var str = document.getElementById("demo").innerHTML;
  var txt = str.replace("Microsoft","W3Schools");
  document.getElementById("demo").innerHTML = txt;
}
</script>
</body>
</html>
```

output


JavaScript String Methods
Replace "Microsoft" with "W3Schools" in the paragraph below:
Try it
Please visit Microsoft!

On clicking


JavaScript String Methods
Replace "Microsoft" with "W3Schools" in the paragraph below:
Try it
Please visit W3Schools!

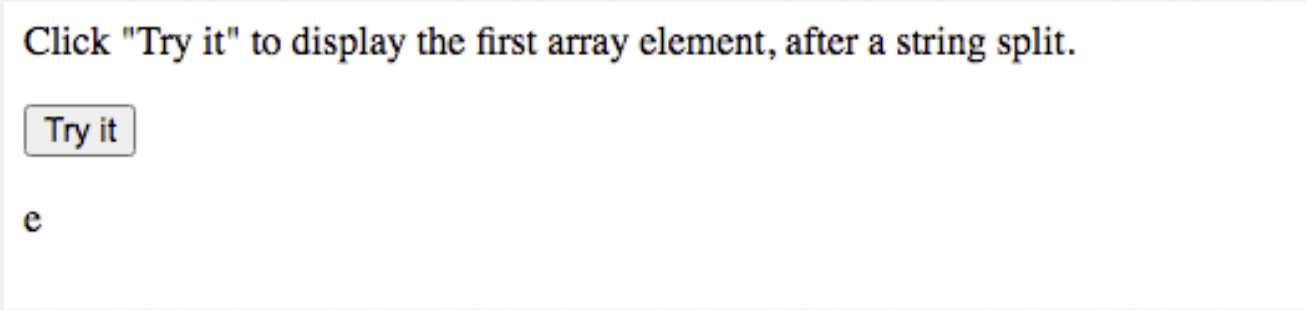
Other string methods

- toUpperCase()
- toLowerCase()
- concat() joins two or more string
- trim() removes white spaces from both sides of a string
- charAt() returns the character at a specified index in a string
- split()

split() example

```
<!DOCTYPE html>
<html>
<body>
<p>Click "Try it" to display the first array element, after a string split.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
  var str = "a,b,c,d,e,f";
  var arr = str.split(",");
  document.getElementById("demo").innerHTML = arr[4];
}
</script>
</body>
</html>
```

output


Click "Try it" to display the first array element, after a string split.
Try it
e

Some number methods

- toString()
- toExponential()
- toFixed()
- toPrecision()

Control structure

- If, else, else if
- Switch
- for – loops through a block of code
- while– loops through a block of code while a specified condition is true
- do while– also loops through a block of code while a specified condition is true

try … catch

- try
- catch