Complex Types

COEN 161

Built in JavaScript Objects

- var x1 = new Object(); // A new Object object
- var x2 = new String(); // A new String object
- var x3 = new Number(); // A new Number object
- var x4 = new Boolean(); // A new Boolean object
- var x5 = new Array(); // A new Array object
- var x7 = new Function(); // A new Function object

String Length Property

- Length

```
var txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

var sln = txt.length; // 26
```

String Methods

- To search for a string within a string use…
- indexOf - returns the **first** occurrence of a string

```
var str = "Please locate where 'locate' occurs!";

var pos = str.indexOf("locate"); // 7
```

- lastIndexOf - returns the **last** occurrence of string

```
var str = "Please locate where 'locate' occurs!";

var pos = str.lastIndexOf("locate"); // 21
```

- Both return **-1** when the string is not found

String Methods

- To extract parts of strings
- slice - returns part of string the *start index* to the *end index*

```
var str = "Apple, Banana, Kiwi";

var res = str.slice(7, 13);  // Banana
```

- If either parameter is negative, it counts the position from the end of the string

```
var str = "Apple, Banana, Kiwi";

var res = str.slice(-12, -6); // Banana
```

String Methods

- slice - if you omit the second parameter, it slices out the rest of the string

```
var str = "Apple, Banana, Kiwi";

var res = str.slice(7); // Banana, Kiwi
```

- If the parameter is negative, it counts the position from the end, and slices to the end

```
var str = "Apple, Banana, Kiwi";

var res = str.slice(-12); // Banana, Kiwi
```

String Methods

- substring - similar to slice, except it doesn't accept negative parameters

```
var str = "Apple, Banana, Kiwi";

var res = str.substring(7, 13); // Banana
```

- If you give it only one parameter, it substrings the rest of the string

```
var str = "Apple, Banana, Kiwi";

var res = str.substring(7); // Banana, Kiwi
```

- If param1 is > param2, substring will swap the two arguments
  - slice doesn't swap them and just returns an empty string

String Methods

- substr - like slice, except the second argument specifies the length of the substring

```
var str = "Apple, Banana, Kiwi";

var res = str.substr(7, 6); // Banana
```

- If the first param is negative, it counts from the end of the string
- The second param cannot be negative because it defines length
- If the second param is missing, it returns the rest of the string

String Methods

- replace - replace part of a string with another string

  str = "Please visit Microsoft!";

  var n = str.replace("Microsoft", "W3Schools"); // "Please visit W3Schools!"

- Note: this doesn’t change the original string, it returns a new string with the new value

String Methods

- toUpperCase - changes the whole string to uppercase

```
var text1 = "Hello World!";

var text2 = text1.toUpperCase();  // "HELLO WORLD!"
```

- toLowerCase - changes the whole string to lowercase

```
var text1 = "Hello World!";

var text2 = text1.toLowerCase();  // "hello world!"
```

String Methods

- concat - concatenates two or more strings together

```
var text1 = "Hello";

var text2 = "World";

var text3 = text1.concat(" ", text2);
```

- Does the same thing as the + operator on strings

```
var text = "Hello" + " " + "World!";

var text = "Hello".concat(" ", "World!");
```

String Methods

- charAt - returns the character at a given position

```
var str = "HELLO WORLD";

str.charAt(0);      // "H"
```

- charCodeAt - returns the unicode value for the character at the given position

```
var str = "HELLO WORLD";

str.charCodeAt(0);  // returns 72
```

String Methods

- You can access strings as arrays, but it is considered unsafe

```
var str = "HELLO WORLD";

str[0];                    // returns H
```

- It doesn’t work in all browsers
- It makes strings look like arrays, and they’re not
- Assignment doesn’t work (strings are immutable)

```
str[0] = "H";
```

String to Array

- split - takes a *delimiter* and returns an array of strings separated by that delimeter

```
var txt = "a,b,c,d,e";    // String

txt.split(",");           // Split on commas

                          // ["a", "b", "c", "d", "e"]
```

String to Array

- Passing an empty string to split returns all the individual characters in an array

  var txt = "Hello"; // String

  txt.split(""); // Split in characters ["H", "e", "l", "l", "o"]

- If you don’t pass anything to split, returns the whole string in index 0

  var txt = "Hello"; // String

  txt.split(); // Split in characters ["Hello"]

toString

- Most objects have a toString method that returns a string version of that object

  ```
  var fruits = ["Banana", "Orange", "Apple", "Mango"];

  // "Banana,Orange,Apple,Mango"

  var fruit = { name:"Banana" };

  // "[object Object]"
  ```

Array Methods

- join - joins all the elements into one string
- Takes an optional delimiter that separates each element in the string

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.join(" * "); // "Banana * Orange * Apple * Mango"
```

Array Methods

- pop - returns the last element from the array

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.pop();  // "Mango"
```

- push - adds an element to the end of the array

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.push("Kiwi");

// ["Banana", "Orange", "Apple", "Mango", "Kiwi"]
```

Array Methods

- shift - removes the first element in the array

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.shift();                 // Returns "Banana"
```

- unshift - adds an element to the beginning of the array

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.unshift("Lemon");
// ["Lemon", "Banana", "Orange", "Apple", "Mango"]
```

Accessing Array Elements

- In array, all elements have an index, starting from 0

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits[0] = "Kiwi";

// Changes the first element of fruits to "Kiwi"

// ["Kiwi", "Orange", "Apple", "Mango"]
```

Accessing Array Elements

- Using length, we can append elements to the array easily

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits[fruits.length] = "Kiwi";

// ["Banana", "Orange", "Apple", "Mango", "Kiwi"]
```

Array Methods

- splice - used to add multiple elements to an array

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.splice(2, 0, "Lemon", "Kiwi");

//["Banana", "Orange", "Lemon", "Kiwi", "Apple", "Mango"]
```

- The first parameter defines where to splice
- The second parameter defines how many elements to remove
- The rest of the parameters are the elements to be added

Array Methods

- splice - can also be used to delete elements from the array

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.splice(0, 1); // Removes the first element of fruits

// ["Orange", "Apple", "Mango"]
```

Array Methods

- concat - used to concatenate multiple arrays together

```
var myGirls = ["Cecilie", "Lone"];

var myBoys = ["Emil", "Tobias", "Linus"];

var myChildren = myGirls.concat(myBoys);

// ["Cecilie", "Lone", "Emil", "Tobias", "Linus"]
```

Array Methods

- slice - takes out a part of the array

  ```
  var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];

  var citrus = fruits.slice(1); // "Orange"
  ```

- You can give it two parameters to take out larger slices

  ```
  var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];

  var citrus = fruits.slice(1, 3);  // ["Orange", "Lemon"]
  ```

Array Methods

- sort - sorts an array, in order (ascending, alphabetically, etc.)

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];

fruits.sort();  // ["Apple", "Banana", "Mango", "Orange"]
```

- reverse - reverse the order of the array

```
fruits.reverse(); // ["Orange", "Mango", "Banana", "Apple"]
```

Array Methods

- sort - can also take a function as an argument

```
//ascending

var points = [40, 100, 1, 5, 25, 10];

points.sort(function(a, b){return a - b});

// descending

var points = [40, 100, 1, 5, 25, 10];

points.sort(function(a, b){return b - a});
```

Reversing a string

Example

Resources

https://www.w3schools.com/js/js_type_conversion.asp

https://www.w3schools.com/js/js_string_methods.asp

https://www.w3schools.com/jsref/jsref_obj_string.asp

https://www.w3schools.com/js/js_array_sort.asp