Closures

event delegation

this

prototypal inheritance

https://codeburst.io/clearing-your-front-end-job-interview-javascript-d5ec896adda4
http://thatjsdude.com/interview/index.html
https://frontendmasters.com/books/front-end-handbook/2018/practice/interview-q.html
https://github.com/h5bp

Math Class

Math.max([value1[, value2[, ...]]])

Function.prototype

.apply()	The apply() method calls a function with a given this value, and arguments provided as an array (or an array-like object). `var max = Math.m ax.a pp ly( null, numbers);`

Array.prototype.

arr.**indexOf(**searchElement[, fromIndex]**)**	The indexOf() method returns the first index at which a given element can be found in the array, or -1 if it is not present. ``
.find()	The find() method returns the value of the first element in the array that satisfies the provided testing function. Otherwise undefined is returned. `var found = array1.fi nd( fun cti on( ele ment) {  return element > 10; });`
arr.**map()**	The map() method creates a new array with the results of calling a provided function on every element in the calling array. `const map1 = array1.map(x => x * 2);`

Objects

An object is a collection of properties, and a property is an association between a name (or key) and a value. A property's value can be a function, in which case the property is known as a method. In addition to objects that are predefined in the browser, you can define your own objects.

A JavaScript object has properties associated with it. A property of an object can be explained as a variable that is attached to the object. Object properties are basically the same as ordinary JavaScript variables, except for the attachment to objects. The properties of an object define the characteristics of the object. You access the properties of an object with a simple dot-notation:

```
objectName.propertyName
var myCar = new Object();
myCar.make = 'Ford';
myCar.model = 'Mustang';
myCar.year = 1969;
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Objects (cont)

Unassigned properties of an object are undefined (and not null).	myCar.color; // undefined
Properties of JavaScript objects can also be accessed or set using a bracket notation (for more details see property accessors). Objects are sometimes called associative arrays, since each property is associated with a string value that can be used to access it. So, for example, you could access the properties of the myCar object as follows:	myCar['make'] = 'Ford'; myCar['model'] = 'Mustang'; myCar['year'] = 1969;
Property accessors provide access to an object's properties by using the dot notation or the bracket notatio	person = {'firstname': 'John', 'lastname': 'Doe'} person['firstname'] = 'Mario'; person['lastname'] = 'Rossi';
myOtherMethod(params) { // ...do something else }	
Object.Prototype	Nearly all objects in JavaScript are instances of Object; a typical object inherits properties (including methods) from Object.prototype, although these properties may be shadowed (a.k.a. overridden). However, an Object may be deliberately created for which this is not true (e.g. by Object.create(null)), or it may be altered so that this is no longer true (e.g. with Object.setPrototypeOf).

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Working_with_Objects

Events

These are code structures which listen for things happening in browser, running code in response.

Attach event to an element

```
document.querySelector('html').onclick = function() {};
```

```
var myHTML = document.querySelector('html'); myHTML.onclick = function() {};
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com