What are base data types in JavaScript?
Learn more about JavaScript data types here:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures
Note: There are 3 correct answers to this question.

- Null
- Boolean
- Float
- Pointer
- Object

Which of the following statements regarding objects in JavaScript are true?
Learn more about JavaScript objects here: http://bonsaiden.github.io/JavaScript-Garden/#object
Note: There are 3 correct answers to this question.

- Objects cannot contain other objects, only functions.
- Everything except for base data types is an object in JavaScript.
- Objects are unordered collections of name-value pairs.
- Inheritance of objects is based on the prototype in JavaScript.
- Objects can access the file system of the client.

What is a closure in JavaScript?
Learn more about closures in JavaScript here: http://bonsaiden.github.io/JavaScript-Garden/#function.closures

- A construct to stop the user from executing dangerous logic twice
- A callback function that waits for a specific event to happen
- Two functions executed at the same time that wait for each other
- A nested function that inherits the scope of its parent function

Learn more about "this" in JavaScript
here:https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Operators/this
Note: There are 2 correct answers to this question.

- In a function scope, "this" points to the last changed DOM element.
- In a constructor scope, "this" points to the first argument of the function.
- In the global scope, "this" points to the window object.
- In an object scope, "this" points to the current object instance.

What is the result when you enter 0.1 + 0.2 in the developer console of your browser?
Learn more about the JavaScript Number Type here: http://javascript.info/tutorial/number-math#imprecise-calculations

- 0.30000000000000004
- 0
- 1.2
- 0.3

What is the result when you enter "5" * "2" in the developer console of your browser?
Learn more about the Type Coercion (Implicit type conversion)
here:http://bonsaiden.github.io/JavaScript-Garden/#types.equality

- 10

- 52
- 7
- fiftytwo

What is the result when you enter 1 == true in the developer console of your browser?
Learn more about Equality and Comparisons in JavaScript here:https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Comparison_Operators

- 1
- false
- true
- undefined

Which of the following statements regarding functions in JavaScript are true?
Learn more about JavaScript functions here: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions
Note: There are 3 correct answers to this question.

- In JavaScript, only a function scope exists and no block scope.
- Functions are objects with an executable part.
- Functions can be passed as arguments to other functions.
- Functions always have a name and arguments.
- Functions are always defined globally in JavaScript.

Why should you make use of asynchronous processing in your JavaScript code?
Learn more about synchronous and asynchronous requests here:https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Synchronous_and_Asynchronous_Requests
Note: There are 2 correct answers to this question.

- To speed up server requests as they do not need authorization checks on the server
- To optimize resource loading and parallelize tasks
- To avoid server crashes due to multiple requests
- To avoid your UI becoming unresponsive due to long-running tasks

What is method chaining (cascading)?
Learn more about method chaining here:http://www.w3schools.com/jquery/jquery_chaining.asp

- The chaining together of multiple function calls within a single JavaScript statement
- The definition of an inner function call inside an outer function
- The definition of two functions in one statement
- The definition of multiple functions inside an object

What is the result of the following function call?

```
function outer(param) {
  var one = "Scopes";
  inner();


  function inner() {
  var one = "Closures";
```

```
    var two = "are";
    three = param;
        console.log(one + " " + two + " " + three);
    }
}
outer("awesome!");
```

Learn more about Closures in JavaScript here: http://bonsaiden.github.io/JavaScript-Garden/#function.closures

- awesome! are Scopes!
- Scopes are awesome!
- Closures are awesome!
- awesome! are Closures!

What is a closure in JavaScript?
Learn more about Closures in JavaScript here: http://bonsaiden.github.io/JavaScript-Garden/#function.closures

- A special kind of object that combines two things: a function, and the environment in which that function was created
- A set of properties in an array that is closed, meaning that no additional properties can be added to the array
- An anonymous function that is never called. It causes memory leaks and should be avoided at all times.
- An object that has been assigned to two variables. When you update the object by accessing variable1 it will also update the object assigned to variable2.

What is "this" when the following method was called?

```
var myObj = {
  property : "that",
  whatsThis : function(that) {
    return this;
  }
}
myObj.whatsThis("?");
```

Learn more about "this" in JavaScript here:https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Operators/this

- myObj
- window
- that
- ?

What is logged to the console as "this" when the following method was called?

```
var myObj = {
  property : "that",
  whatsThis : function(that) {
    setTimeout(function () {
      console.log(this);
    }, 0);
  }
}
myObj.whatsThis("?");
```

Learn more about "this" in JavaScript here:https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Operators/this

- ○ window
- ○ undefined
- ○ that
- ○ this

Which of the following statements evaluate to true?
Learn more about truthy and falsy values here:https://developer.mozilla.org/de/docs/Glossary/Truthy
Note: There are 3 correct answers to this question.

- ○ 1 === true
- ○ 1 == true
- ○ 0 == ""
- ○ "false" === true
- ○ "1" == true