

Final Study Guide

Here is a representative list of questions for the final. You should be able to answer these questions or questions similar to them.

1. Explain the differences between local variables, parameters, the special variable `this`, and global variables.
2. Which of the following are valid ways of accessing the element at the entry with index 1 in an array `a`?
 - i. `a.1`
 - ii. `a."1"`
 - iii. `a1`
 - iv. `a[1]`
 - v. `a["1"]`
 - vi. `a(1)`
 - vii. `a("1")`
3. Which of the following will the expression `Object.create({ a: 2 })` create?
 - i. An object with a property `a`.
 - ii. An object whose prototype has a property `a`.
 - iii. An empty object with empty prototype.
 - iv. Nothing useful, it is not a valid expression.
4. Create an object `obj` that has an enumerable property `b` but for which the expression `obj.hasOwnProperty("b")` returns `false`. What would then be two valid ways to test that `obj` does indeed have a property `b`?
5. Which of the following can be possible scopes for local variables (distinguish between variables declared via `var` and those declared via `let` or `const`)?
 - i. Function bodies
 - ii. Any sets of curly braces
 - iii. the bodies of `for` loops
6. Using ES6 templates, write a function that takes as a parameter someone's name and returns a string that greets them.
7. What do template libraries like Handlebars bring to us that can't easily be done with ES6 templates?
8. What would be the code that would allow us to add to an object `o` a property `a` whose value is 20, it cannot be changed, and it appears when we look at the object's keys via `Object.keys`?

9. Similar to the Person class that appears at the bottom of the object properties notes¹, write a Point class with the following behaviors: javascript `let point = new Point(2, 3); point.`
10. The following code creates an adder class, and uses it:

```
// In module adder.js
export default class Adder {
  constructor() {
    this.numbers = [];
  }
  add(x) {
    this.numbers.push(x);
  }
  computeSum() {
    let sum = 0;
    for (let x in this.numbers) {
      sum += x;
    }
    return sum;
  }
}

// In another module
import Adder from './adder.js';

let adder = new Adder();
adder.add(4);
adder.add(3);
console.log(adder.computeSum()); // Normally prints 7
```

You also have a module of your own, which gets loaded inbetween the Adder module and the module that uses the Adder module. Describe what changes you would make to achieve each of the following:

- The add method always adds the number 42 in the numbers array.
- The computeSum method always returns 42 as its result.
- The push method does NOT add the elements in the array (or any other array for that matter).
- The add method does its job normally, but the numbers are also added to an array defined in *your* module.

11. We discussed three different graphics technologies for the web. What are they? What are the pros and cons of each?
12. Write SVG.js code that would produce the following:
- Two circles of equal radius that each pass through the other's center.
 - A chessboard pattern (8x8 with alternating colored squares).
 - A circle that changes in size over a period of time.
13. Describe what Javascript Promises are, what states a promise can be in, and how to use a promise to handle an asynchronous event. Use promises to create a `delay(millisecons)` function that can be used to trigger actions after a specific time.

¹notes/object_properties.html

14. What is the *same-origin policy*? Why is it in place? Why does it not affect script tags?
15. Describe how CORS works, being clear about all the actors involved.
16. Describe how the keywords `async` and `await` behave, and what they allow us to do.
17. What is the `XMLHttpRequest` object used for? How is it set up to use so that we do not have to block everything while waiting for a reply?