# Final Study Guide

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

1. Explain how the special variable `this` differs from other variables (e.g. local or global variables).
2. Javascript, HTML and CSS are all responsible for different parts of a webpage. Describe the role of each.
3. 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")`

4. What happens when we call a function with more arguments than its definition suggests?

   i. An error is thrown
   ii. A warning is thrown
   iii. The extra arguments can be accessed via a specific syntax
   iv. The extra arguments are ignored

5. What do we mean when we say that functions are “first-class values”? Illustrate with examples.
6. What is the result of the expression `("2" + 3)`?

   i. `"23"`
   ii. `23`
   iii. `"5"`
   iv. `5`
   v. `NaN`
   vi. An error

7. Similar question for `("2" * 3)` and `("2" * "3")`.
8. Which of the following values are treated as “falsy” (i.e. considered as false for the purposes of a conditional)?

   i. `0`

ii. NaN
iii. −1
iv. false
v. "false"
vi. ""
vii. {}
viii. null
ix. undefined
x. []

9. 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.

10. What is the difference between the expressions (a in obj) and (obj.hasOwnProperty("a"))?

11. 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?

12. Which of the following can be possible scopes for local variables?

    i. Function bodies
    ii. Any sets of curly braces
    iii. the bodies of for loops

13. Describe the "Immediate Function Invocation" pattern and what its purpose is.

14. Demonstrate how we can write a function oneTime(f) with the following behavior. It expects as argument a function f that would be called with no arguments. It then returns a function g that when called (you can assume with no arguments) will call f and return the same value that f would return, but so that subsequent calls to g just return that same value without calling f again. Further, the function f should not be called until g is called for the first time.

15. How can we arrange for a function f to be called at some time in the future, say in 1 second? Show the precise syntax. Provide reasons why f might not actually be called in exactly 1 second.

16. Outside of the Function prototype methods like bind, call and apply, there are three other ways of invoking a function:

    i. Function invocation
    ii. Method invocation
    iii. Constructor invocation

Describe what the syntax for each of these is, and what the value of the this object is in each case.

17. Describe the Observer pattern. What is its goal, and how is this goal achieved in Javascript?

18. Describe the Visitor pattern, what problem it solves, how it achieves it (in Javascript), and in what situations it might be appropriate.

19. Describe the Iterator pattern, what problem it solves, and how we might implement it in Javascript.

20. Many Graphical User Interface applications promote the Model-View-Controller pattern. Describe what these three components do, what their responsibilities are, and how they interact.

21. Two fundamental methods of code "reuse" are *inheritance* and *composition*. Describe how each works and what some tradeoffs are.

22. What is the *same-origin policy*? Why is it in place? Why does it not affect script tags?