

Midterm 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. Describe the syntax for arrow functions, and how the `this` variable behaves for such functions.
10. List at least 4 methods each, and describe the behavior of those methods, for each of the following object classes (you should also know the behavior of all standard methods for these classes, if given their name):
- Array
 - Map
 - Set
 - String
11. Using Javascript's `class` syntax, be able to define a small class that exhibits a certain required behavior (e.g. a counter, person or task class).
12. Which of the following can be possible scopes for local variables declared via `let`? What about variables declared via `var`?
- i. Function bodies
 - ii. Any sets of curly braces
 - iii. the bodies of for loops
13. 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.
14. 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.
15. 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.

16. Describe the Observer pattern. What is its goal, and how is this goal achieved in Javascript?
17. Be able to provide the CSS selector that would represent something like (this is just one example) “all paragraph tags that are within a section tag with class ‘topic’, when we hover over them.”