Eloquent JavaScript

Review Questions

Key:

This means it’s a question I either struggled with or got wrong. These should be the first to review

This means I have answered but need to confirm the answer

This means I have confirmed the answer or looked it up

Chapter 1: Values, Types, and Operators

1. Integers are considered precise while fractional numbers (floating-point numbers) are not. Why is that? This is because numbers are represented in computers with a finite number of bits. This is fine when the number has an exact end (as integers, which have no trailing numbers). Floating-point numbers (ex. 1/3) many times don’t have an exact end. As a result, the computer has to approximate the answer as closely as possible
2. What is the remainder operator? What is it sometimes called? The remainder operator is %. It will divide the number before the symbol by the number after the symbol and return the remainder (if any) of the operation. It is sometimes called modulo
3. In a string, what character signifies that you’re escaping the character (ex. Creaing a new line) in JavaScript? \n
4. What can you use to concatenate two strings (combine them together into one string)? + as in ‘Hello’ + ‘World’ will return ‘HelloWorld’
5. What characters mark the beginning and end of a template literal? ` `
6. What is a special feature of template literals using $? Provide an example. You can input variables and even operations into a string directly by using ${}. As an example  
    let name = 'Arturo';

console.log(`Hello ${name}! How are you today?`);  
//output→ “Hello Arturo! How are you today?”

1. What is a unary operator? An operator that only takes one operand/argument. Ex ! is the “not” operator and only takes one operand.   
    Ex. !false //output: true
2. What does typeof do? How does it work? typeOf is a unary operator that takes one operand and will return a string indicating the operand’s data type. Ex.  
    typeOf(1); //returns “number”
3. When comparing strings in JavaScript (for example, “Aardvark” < “Zoroaster”, how does JavaScript compare the strings to determine if they are equal? JavaScript goes over the characters from left to right, comparing the Unicode codes one by one. uppercase letters are always “less” than lowercase letters.
4. What is the one value in JavaScript that is not equal to itself? NaN
5. What are the three logical operators (with symbols)? && (and) || (or) ! (not)
6. When does || result in true? If either side of the operator is true
7. When does && result in true? True only if both sides of the operator are true
8. When does ! result in true? When it’s operand is false – not operator flips the boolean value to it’s opposite
9. What is a ternary operator? What is the syntax of the ternary operator in JavaScript?  
   A ternary operator is an operator that takes 3 arguments. In JavaScript, the only ternary operator is the conditional operator. It takes 3 arguments a, b, c in the form a ? b : c  
   The expression will evaluate to b if a is true or otherwise evaluates to c. It is, essentially a short hand way to write an if...else... statement
10. What is type coercion? What is an example of when this might occur? Values have different types (ex. string, number, etc) depending on whether it represents text, a number, a boolean value etc. Type coercion is the change between one type into a different type.   
    This can happen either implicitly or explicitly. Explicitly means that we change the type directly by inputting code to make the change. Ex.   
     Number(“1”); //will return the value 1 as a number instead of a string  
    Implicitly means that JavaScript will quietly convert the value to the type it needs when an operator is applied to the “wrong” type of value. It does this by following a set of rules as to which value to change and depending on the circumstances. Ex  
     console.log(1 + “1”);

will return the string “11”, by coercing the number 1 into a string value “1” and then concatenating that string to the other “1” string  
  
Implicitly means that the type held by a

1. What do arithmetic operations on NaN produce? How is this different from an arithmetic operation on null?
2. When something that doesn’t map to a number in an obvious way (such as “five”) is converted to a number, what is the value?
3. How can you perform comparison without type conversion? Show the symbols used for precisely equal to and precisely not equal to
4. Logical Operators work in a peculiar way. They convert values and decide what to do in a unique fashion. Explain how this works for && and ||
5. The rules for converting strings and numbers to Boolean values state that 0, NaN and “” count as …?

Chapter 2: Program Structure

1. Define expression
2. Define statement
3. Define binding (aka variable)
4. Bindings don’t contain values, they…
5. Define keyword
6. What is the syntax to declare multiple values with a single let statement
7. What is the difference between var, let/const?
8. What is the difference between declaring variables with let vs const
9. What are the rules for giving bindings aka variables names? (4 rules)
10. Define the environment
11. When a program starts, is the environment empty? If not, what does it contain?
12. Define function
13. How do you call a function?
14. Some functions will accept values passed to them in the () after a function. What are these values called?
15. Explain how console.log() is not a simple function. What is it?
16. What is the difference between side effects and return values?
17. Control Flow: Generally, programs will run from ….
18. Control flow: What can you use to modify the control flow in a program? (We’re looking for 2 different general ways to do this)
19. What is the Number() function and what does it work?
20. What does Number() return if you pass it a string that cannot be transformed into a valid number?
21. “if” keyword – what does it do?
22. Syntax to use the “if” keyword
23. What function can you use to determine if a value is NaN?
24. What is Number.isNan() ? How does it work?
25. What are {} braces used for in JavaScript (not their use with objects)?
26. What is a block?
27. “else” keyword – what is it used for
28. Syntax to use “else” keyword
29. You can chain multiple if...else if… statements together. What is important to keep in mind about the number of blocks in the chain that will run
30. Define loop
31. How does the while loop work?
32. What is the syntax for a while loop?
33. What do you need to make sure happens in the body of a while loop so that you don’t create an infinite loop?
34. What is the syntax for a “do while” loop?
35. What is the syntax for a “for” loop?
36. What does the “break” keyword do and how is it used?
37. What does the “continue” keyword do and how is it used?
38. How can you use the modulo (aka reminder) operator to test if a number is divisible by another number?
39. What is the switch statement?
40. What is the syntax of the switch statement?
41. What type of case do we use for variable names (per convention to make it easier to read)?
42. What does Number() have the first letter capitalized?
43. Do the exercised in Chapter 2: Program Structure

Chapter 3: Functions

1. Name the 3 techniques to create a function
2. Syntax to create a function with a function definition
3. Syntax to create a function with a function declaration
4. Syntax to create a function with an arrow function
5. When defining a function or using an arrow function, how can you ensure that the function can’t be overwritten? Can?
6. A simple way to say how an arrow function works: An arrow function expressed something like “this input (the parameters) \_\_\_\_\_\_ this result (the body)”
7. How does a return statement work in a function? (2 things happen)
8. When creating a function, what are the variables between () called?
9. Parameters are like normal bindings/variables but the values are given by …
10. What is a scope?
11. What is the difference between global and local scope?
12. What determines whether a variable is a global or local variable?
13. What happens to local bindings on a function every time the function is called?
14. What is the benefit of the new creation of local bindings when the function containing local bindings is called?
15. Bindings created with let and const are local to the \_\_\_\_\_\_ that they are declared in
16. As a result of where bindings created with let and const are local to, code before or after a loop where the variable is created CAN/CANNOT “see” the variable. (CHOOSE ONE)
17. Why do we use let/const to declare a variable instead of var?
18. Each scope can look \_\_\_ into the scope around it
19. What is lexical scoping?
20. How are function declarations - ex.   
     function FuncName(x) {}  
    different from other code in terms of flow of control?
21. Explain the call stack - use of control flow through a program with a function. (i.e. What happens when a function is called in the normal course of a program running)
22. What happens if you pass more arguments to a function than the number of parameters in the function?
23. What are default parameters? What is the syntax for creating a default parameter in a function?
24. Define closure
25. What is A closure?
26. Function values contain both the code in their body and the… When called, the function body sees…, not where it is called
27. What does it mean for a function to be recursive?
28. Is recursion usually the most efficient implementation? If not, what is the reason for using recursion?
29. Being overly concerned with efficiency can be paralyzing and distracting. What is the best methodology?
30. What are 2 more or less natural ways for functions to be introduced?
31. How can you find the length of a string (ex.- a string called string)?
32. What is a pure function? Do you have to worry about using “non-pure” functions?
33. A key aspect to understanding functions is understanding scopes
    1. each block creates a new…
    2. Parameters/bindings declared in a given scope with let/const are … ie not visible …
       1. Bindings created with var behave differently than this. Explain how

Chapter 4: Data Structures: Objects and Arrays

1. Describe an array
2. Values in an array are stored in such a way that the position of each element can be computer from its…
3. How is an array written?
4. What is the first index of an array?
5. How can you get an element from an array? Please provide an examples
6. What is a property in JavaScript?
7. Do most values JavaScript values have properties? Or is it only objects?
8. What are the two notations to access a property on a value?
9. What is the difference between accessing a property with dot notation or square bracket notation?
10. If the property name is stored in a variable, use .. so that the variable is evaluated to the string property name
11. Objects use dot notation when the property name looks like… Provide an example
12. Arrays use numbers as property names, so we use… to get the properties. Provide an example
13. Define method
14. What is the toUpperCase() method? Where is it located and what is the syntax to make it work?
15. What is the toLowerCase() method? Where is it located and what is the syntax to make it work?
16. What is the push() method? Where is it located and what is the syntax to make it work?
17. What is the pop() method? Where is it located and what is the syntax to make it work?
18. Define a stack
19. Define object
20. How can you create an object? Provide the syntax
21. What is the “in” operator? What does it do and how does it work?
22. What is the Object.keys() function? What does it do and what is the syntax to make it work?
23. What is the Object.assign() function? What does it do and what is the syntax to make it work?
24. Arrays are just a special type of…
25. Do objects necessarily need to be named? Provide an example when they wouldn’t need to be named
26. What does it mean to say that strings, booleans and numbers are immutable values? Can you provide an example?
27. With objects, there is a difference between having two references to the same objects and having two different objects that contain the same properties. Consider the following code:  
    let object1 = {value:10};  
    let object2 = object1;  
    let object3 = {value:10};  
    1. Which objects have the same reference?
    2. Which contain the same properties but a different reference?
    3. Which could you change to make a change in another object?
    4. For which does objectA === objectB return true? Why?
28. What does it mean to say that two objects have the same identity?
29. When you compare objects with JavaScript’s == operator, it compares the objects by…
30. consider the following code   
    function addEntry(events,squirrel){  
     journal.push({events,squirrel});  
    }  
    Instead of declaring the properties like events:events, it just lists a property name. Why will this work?
31. What is the Math.sqrt() function? What does it do and what is the syntax to make it work?
32. What is the includes() method? Where is it located, What does it do and what is the syntax to make it work?
33. What would you use a for...of loop? Where does it work? What is the syntax?
34. What is the unshift() method? Where is it located, What does it do and what is the syntax to make it work?
35. What is the shift() method? Where is it located, What does it do and what is the syntax to make it work?
36. What is the indexOf()array method? What does it do and what is the syntax to make it work?
37. What is the lastIndexOf() method? Where is it located, What does it do and what is the syntax to make it work?
38. What is the slice() array method? What does it do and what is the syntax to make it work?
39. What is the concat() method? Where is it located, What does it do and what is the syntax to make it work?
40. What is the indexOf() string method? What does it do and what is the syntax to make it work?
41. What is the slice() string method? What does it do and what is the syntax to make it work?
42. What is the trim() method? Where is it located, What does it do and what is the syntax to make it work?
43. What is the padStart() method? Where is it located, What does it do and what is the syntax to make it work?
44. What is the split() method? Where is it located, What does it do and what is the syntax to make it work?
45. What is the join() method? Where is it located, What does it do and what is the syntax to make it work?
46. What is the repeat() string method? What does it do and what is the syntax to make it work?
47. How can you access individual characters on a string?
48. What is the spread operator?
49. How can you set up a function so that it can accept any number of arguments in its parameters?
50. What is a rest parameter? How is it related to the spread operator?
51. How can you call a function with an array of arguments such that each of the elements in the array are “spread out” as separate arguments?
52. How can you “spread” the elements of one array into another? Please provide an example.
53. What is namespacing (according to coding train)? And why would we use it?
54. What is the global namespace? What is the problem with defining global variables?
55. What is the reasoning behind having the Math object to hold math related functions like max and min?
56. What is the Math.max() method? What does it do and what is the syntax to make it work?
57. What is the Math.min() method? What does it do and what is the syntax to make it work?
58. What is Math.PI? What does it do and what is the syntax to make it work?
59. What is the Math.random() method? What does it do and what is the syntax to make it work?
60. What is the Math.floor() method? What does it do and what is the syntax to make it work?
61. How do you get a random whole value number between 0 and another number (ex.10)
62. What is the function you can use to round a number up?
63. What is the function you can use to round a number to the nearest whole?
64. What is the function you can use to get the absolute value of a number?
65. What is destructing? Can you provide an example?
66. “Because properties only grasp their value, rather than contain it, objects and arrays are stroed in the computer’s memory as a sequence of bits holding their \_\_\_\_\_\_”
67. How do we transmit data in a file/save the data for later when objects and arrays only store addresses?
68. What does JSON stand for?
69. What are two differences between JSON and the way we make JavaScript arrays and objects?
70. What function can we use to convert data to the JSON format?
71. What function can we use to convert data from the JSON format?
72. Complete the exercises at the end of Chapter 4: Data Structures: Objects and Arrays

Chapter 5: Higher-order Functions

1. Define higher order functions
2. Why are higher order functions useful?
3. What is abstraction? Why is it useful?
4. Can functions return other functions? Give an example
5. Consider the following function  
   function noisy(f) {  
    return (...args) => {  
    console.log("calling with", args);  
    let result = f(...args);  
    console.log("called with", args, ", returned", result);  
    return result;  
    };  
   }  
   noisy(Math.min)(3, 2, 1); //returns 1  
   What is the path of execution here?
6. What is the forEach() method? Where is it located, What does it do and what is the syntax to make it work?
7. How is forEach() different from (for...of)?
8. What is the filter() method? Where is it located, What does it do and what is the syntax to make it work?
9. Does filter() modify the array it filters?
10. What is the map() method? Where is it located, What does it do and what is the syntax to make it work?
11. What is the reduce() method? Where is it located, What does it do and what is the syntax to make it work?
12. Filter, map and reduce are more computationally expensive than simply creating a loop. Why might you use the methods over the loop? And vice versa?
13. What is the some() method? Where is it located, What does it do and what is the syntax to make it work?
14. What is a code unit (in reference to how strings are coded)?
15. What is UTF-16?
16. What is a problem caused with how UTF-16 describes certain characters (ex emoji)?
17. How can you tell if a code unit you’re looking at describes a whole character or is part of a larger character?
18. What is the codePointAt() method? Where is it located, What does it do and what is the syntax to make it work? What is a key benefit it has over charCodeAt()?
19. What is a benefit of using a for/of loops when trying to loop over a string with regards to UTF-16?
20. What is the findIndex() method? Where is it located, What does it do and what is the syntax to make it work?
21. How does findIndex() differ from indexOf()?
22. Complete the exercises at the end of Chapter 5: Higher-order Functions