Intro to JavaScript

Pre / Post Test

Name:_____ Date:_____

1. What would you call this pattern?

```
(function () {
  Car = function (make) {
    this.make=make;
    this.go = function () {
      // Do stuff to make it 'go'
    }
  };
})();
```

   a. Anonymous function pattern
   b. Revealing object pattern
   c. Immediately invoked function expression pattern
   d. Inner function creation pattern
   e. Internal function creation pattern
   f. None of the above
   g. I don't know

2. JavaScript debuggers …
   a. Have emulation mode so you can test in the five major browsers simultaneously
   b. Usually ship with the JavaScript compilers
   c. Are built into most browsers because they run client-side
   d. All of the above
   e. I don't know

3. You want to find out if a is exactly the same as b. Which of the following is correct?
   a. a.equals(b)
   b. a = b
   c. a == b
   d. a === b
   e. A or C
   f. I don't know

4. JavaScript variables …
   a. have local scope unless you use the var keyword in a function
   b. have global scope unless you use the var keyword in a function
   c. always have local scope (local to the block)
   d. always have global scope
   e. I don't know

5. It is okay to leave the semicolon off the end of JavaScript statements.
   a. True
   b. False
   c. I don't know

6. Which is the proper syntax to import a JavaScript module into another if it were exported like this: "export const foo = [1, 2, 3];"?
   a. import "foo" from "./foo";
   b. var foo = require("./foo");
   c. import foo from "./foo";
   d. import { foo } from "./foo";
   e. Any of the above
   f. I don't know

7. What is wrong with the following code?

```
function buttonClick() {
    function registerUser(user, first, last) {
        var reply = callSomeAjaxFunction();
        if (reply == 0)
            return true;
        else
            return false;
    }
    var reply = registerUser(username);
    alert("Register user function returned " + reply);
}
```

   a. There are no curly braces after the if statement
   b. The variable reply can become corrupted
   c. You can't have a function inside a function
   d. The call to registerUser has a bad number of parameters
   e. There's nothing wrong with the code. It will work fine
   f. I don't know

8. What goes in the blank below to see something in the console?

```
const x = __________;
if (x) {
  console.log("It is true");
);
```

   a. "false"
   b. {}
   c. -1
   d. { validate: "true" }
   e. "true"
   f. All the above
   g. I don't know

9. In modern browsers, which is the preferred style of wiring event handlers?
   a. element.onevent = function () {...};
   b. element.addEventListener('event', function () {...});
   c. element.attachEvent('onevent', function () {...});
   d. <element onevent="f()" />
   e. I don't know

10. Which statement about prototyping is true?
   a. Prototypes can't be modified before the object is instantiated
   b. Prototypes can't be modified after the object is instantiated
   c. Prototypes allow objects of the same type to share behavior
   d. Prototypes allow classes to share behavior
   e. Prototypes can't be added to instantiated objects
   f. Prototypes can't be added to declared objects
   g. I don't know

11. Assuming the url returned good JSON data with an array of objects, what would be in the variable foo if we ran "var foo = fetch('/api/persons')"?
    a. An array of persons
    b. An array of objects
    c. An array of strings
    d. A string
    e. None of the above
    f. I don't know

12. Given this code, how would you get p1's id?
```
Person.prototype.getId = function() {
  return this.id;
};
var p1 = new Person(42);
```
    a. Person.getId();
    b. Person.prototype.getId();
    c. p1.getId();
    d. p1.prototype.getId();
    e. C or D
    f. All of the above
    g. I don't know

13. Which of these is NOT a valid event?
    a. click
    b. dblclick
    c. blur
    d. rightclick
    e. reset
    f. I don't know

14. JavaScript supports traditional inheritance
    a. True
    b. False
    c. I don't know

15. How would you include an array inside another in JavaScript?
    a. const array2 = [1, 2, 3, ...array1, 9, 10,];
    b. var array2 = [array1[0 .. 2], array1, array1[3 .. 4]];
    c. let array2 = array1.splice[0,2] + array1 + array1.splice[3,4];
    d. var array2 = [array1[0 ... 2], array1, array1[3 ... 4]];
    e. None of the above
    f. I don't know

16. JavaScript is a strongly typed language.
    a. True
    b. False
    c. I don’t know

17. Which of these Array.prototype methods would be good to create a new array based on an existing array?
    a. Array.prototype.find()
    b. Array.prototype.some()
    c. Array.prototype.every()
    d. Array.prototype.new()
    e. Array.prototype.map()
    f. I don't know

18. Your JavaScript code cannot exist in …
    a. a <script> tag in the <head> section of your page
    b. a <script> tag in the <body> section of your page
    c. a file on our server completely separate from your page
    d. a file on someone else's server completely separate from your page
    e. encrypted format so that the browser can't see what will execute.
    f. E nor D
    g. I don't know

19. webpack can bundle, minify, and process ...
    a. css
    b. html
    c. javascript
    d. A and C
    e. B and C
    f. All of the above
    g. I don't know

20. npm is good for what two things? (Choose the best answer)
    a. Testing and linting JavaScript
    b. MInifying and bundling JavaScript
    c. Testing JavaScript and organizing libraries
    d. Automating the build process and organizing libraries
    e. Automating the build process and minifying JavaScript
    f. I don't know