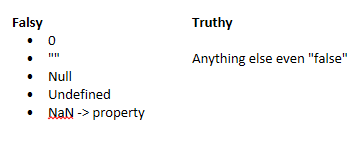
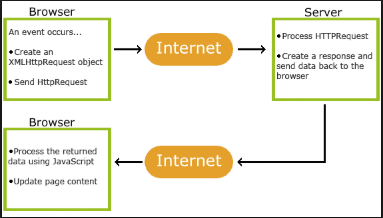
1. What are the truthy and falsy values?



1. Explain the difference between let, var, and const.
   1. **var** variables can be re-declared and updated
   2. **let** is block scoped, can be updated but not re-declared
   3. **const** is block scoped, can not be updated or re-declared
2. What's a callback function? Where have you seen them used?
   1. **callback function** is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action.
   2. Ex: the insertion of the data into our tables
      1. Also, on the traversing DOM function
3. Explain the AJAX workflow.



1. An event occurs in a web page (the page is loaded, a button is clicked)

2. An XMLHttpRequest object is created by JavaScript

3. The XMLHttpRequest object sends a request to a web server

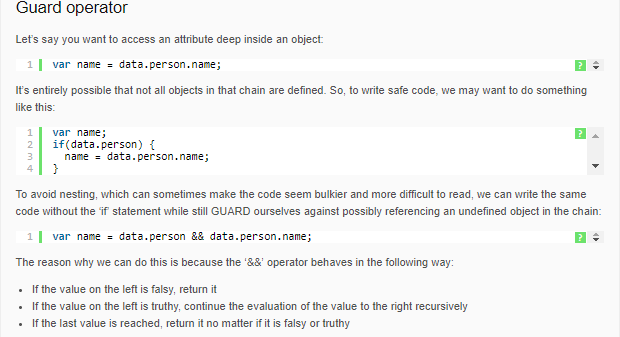
4. The server processes the request

5. The server sends a response back to the web page

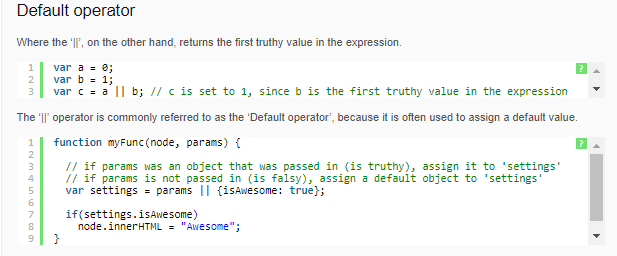
6. The response is read by JavaScript

7. Proper action (like page update) is performed by JavaScript

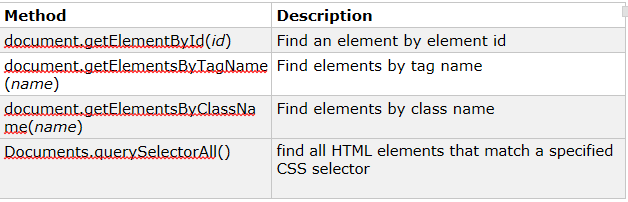
1. What is AJAX? Why do we use it?
   1. AJAX = **A**synchronous **J**avaScript **A**nd **X**ML
      1. AJAX just uses a combination of:
         1. A browser built-in XMLHttpRequest object (to request data from a web server)
         2. JavaScript and HTML DOM (to display or use the data)
   2. To create fast and dynamic webpages
2. Explain type coercion.
   1. The process of converting value from one type to another (such as string to number, object to boolean, and so on).
3. What are the three ways I can handle events with JavaScript?
   1. \*\*\*\***NOTE** not positive on what exactly is being asked but I believe it to be this\*\*\*
   2. HTML event attributes can execute JavaScript code directly
   3. HTML event attributes can call JavaScript functions
   4. You can assign your own event handler functions to HTML elements
4. Name some of the types of events we can work with.
   1. Click
   2. Mouseover
   3. mouseout
   4. Blue
   5. Change
   6. Keydown
   7. Keyup
   8. Load
   9. Submit
5. Explain the guard operator.



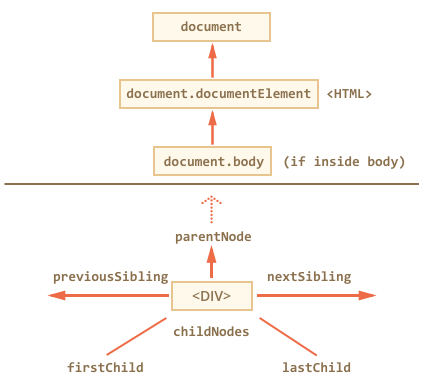
1. Explain the default operator.



1. What are the different scopes in JavaScript?
   1. **Global** - accessible anywhere
   2. **Local**- within the scope of a function
   3. **Block**- applies to any form of control with {}
2. What are the primitive types in JavaScript?
   1. Boolean
   2. String
   3. Number
   4. Null
   5. Undefined
   6. Symbol
3. In what ways can I select elements from the DOM?



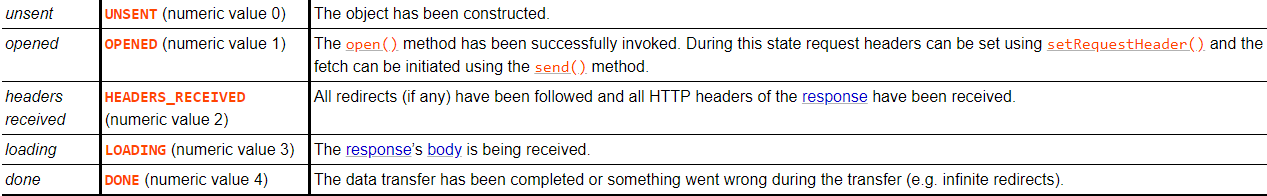
1. What is the DOM? Explain its structure.
   1. Document Object Model
   2. Tree representation of our HTML page with JS objects



1. What is JSON? Why is it important?
   1. JSON: JavaScript Object Notation
      1. Language agnostic data formatted (very similar to a JavaScript object literal)
      2. Basically a string representation of our JS object
   2. Provides standardized method for data transportation
2. Name some methods and properties we can work with DOM elements.

|  |  |
| --- | --- |
| **Method** | **Description** |
| *element*.innerHTML =  *new html content* | Change the inner HTML of an element |
| *element*.*attribute = new value* | Change the attribute value of an HTML element |
| *element*.setAttribute*(attribute, value)* | Change the attribute value of an HTML element |
| *element*.style.*property = new style* | Change the style of an HTML element |

1. What are the different ready states of an XHR object?



1. What types of information can we get from an Event object?
   1. Target
   2. Type
   3. Srcelement
   4. currenttarget
2. What are some methods which are provided by a String object in JavaScript?

**Method Description**

* **charAt()**- Returns the character at the specified index (position)
* **charCodeAt()**- Returns the Unicode of the character at the specified index
* **concat()**- Joins two or more strings, and returns a new joined strings
* **endsWith()**- checks whether a string ends with specified string/characters
* **fromCharCode()**- Converts Unicode values to characters
* **includes()**- Checks whether a string contains the specified string/characters
* **indexOf()**- Returns the position of the first found occurrence of a specified value in a string
* **lastIndexOf()**- Returns the position of the last found occurrence of a specified value in a string
* **localeCompare()**- Compares two strings in the current locale
* **match() -** Searches a string for a match against a regular expression, and returns the matches
* **repeat()**- Returns a new string with a specified number of copies of an existing string
* **replace()**- Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced
* **search() -** Searches a string for a specified value, or regular expression, and returns the position of the match
* **slice()**- Extracts a part of a string and returns a new string
* **split()**- Splits a string into an array of substrings
* **startsWith()**- Checks whether a string begins with specified characters
* **substr()** - Extracts the characters from a string, beginning at a specified start position, and through the specified number of character
* **substring()**- Extracts the characters from a string, between two specified indices
* **toLocaleLowerCase()**- Converts a string to lowercase letters, according to the host's locale
* **toLocaleUpperCase()**- Converts a string to uppercase letters, according to the host's locale
* **toLowerCase()**- Converts a string to lowercase letters
* **toString()**- Returns the value of a String object
* **toUpperCase()**- Converts a string to uppercase letters
* **trim()**- Removes whitespace from both ends of a string
* **valueOf()**- Returns the primitive value of a String object

1. Describe different ways we can create JavaScript objects.
   1. Define and create a single object, using an object literal.
   2. Define and create a single object, with the keyword new.
   3. Define an object constructor, and then create objects of the constructed type.
2. Describe flow of control statements available for use with JavaScript?
   1. If: Executes a statement if a specified condition is true. If the condition is false, another statement can be executed.
      1. If-else
      2. Else
   2. Switch
   3. For: Creates a loop that consists of three optional expressions, enclosed in parentheses and separated by semicolons, followed by a statement executed in the loop.
   4. While: Creates a loop that executes a specified statement as long as the test condition evaluates to true. The condition is evaluated before executing the statement.
3. What are the uses of the continue and break keywords?

