

Front-end Developer Interview Questions Review

General

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing semantics needed by assistive technologies such as screen readers. ARIA enables

developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

[Explain some of the pros and cons for CSS animations versus JavaScript animations.](#)

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

[What does CORS stand for and what issue does it address?](#)

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

[What's a doctype do?](#)

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0`, `overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

[Describe pseudo-elements and discuss what they are used for.](#)

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

[Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.](#)

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){
  foo = function() {}
})();
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
  return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]
(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

`GET` is used to retrieve remote data, and `POST` is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

domains:

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

HTML

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0, overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

[Describe pseudo-elements and discuss what they are used for.](#)

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

[Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.](#)

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){  
    foo = function() {}  
})();  
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
    return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]
(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

GET is used to retrieve remote data, and **POST** is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

domains:

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

CSS

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0`, `overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

Describe pseudo-elements and discuss what they are used for.

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){
  foo = function() {}
})();
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
  return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]
(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

`GET` is used to retrieve remote data, and `POST` is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

domains:

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

JavaScript

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

[Explain some of the pros and cons for CSS animations versus JavaScript animations.](#)

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

[What does CORS stand for and what issue does it address?](#)

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

[What's a doctype do?](#)

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0`, `overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

Describe pseudo-elements and discuss what they are used for.

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){  
    foo = function() {}  
})();  
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
    return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]

(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

GET is used to retrieve remote data, and **POST** is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

domains:

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

jQuery

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0, overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

Describe pseudo-elements and discuss what they are used for.

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){  
    foo = function() {}  
})();  
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
  return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]

(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

GET is used to retrieve remote data, and **POST** is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```


Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

Testing

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0`, `overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

Describe pseudo-elements and discuss what they are used for.

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){
  foo = function() {}
})();
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
    return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]

(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

GET is used to retrieve remote data, and **POST** is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

domains:

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

Performance

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0, overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

[Describe pseudo-elements and discuss what they are used for.](#)

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

[Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.](#)

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){  
    foo = function() {}  
})();  
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
    return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]
(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

`GET` is used to retrieve remote data, and `POST` is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

domains:

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

Network

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0`, `overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

Describe pseudo-elements and discuss what they are used for.

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){
  foo = function() {}
})();
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
    return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]
(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

GET is used to retrieve remote data, and **POST** is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

domains:

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

Coding

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0`, `overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

[Describe pseudo-elements and discuss what they are used for.](#)

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

[Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.](#)

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){
  foo = function() {}
})();
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
    return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]
(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

GET is used to retrieve remote data, and **POST** is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

Fun

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0`, `overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

Describe pseudo-elements and discuss what they are used for.

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){
  foo = function() {}
})();
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
    return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]
(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

`GET` is used to retrieve remote data, and `POST` is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>

Links

What UI, Security, Performance, SEO, Maintainability or Technology considerations do you make while building a web application or site?

Answer

Semantics, performant CSS, performant JavaScript, independent patterns, code quality (linters), emphasis on the "above the fold" content.

Which version control systems are you familiar with?

Answer

Git.

If you have 5 different stylesheets, how would you best integrate them into the site?

Answer

Concatenate, minify and link as one file.

Can you describe the difference between progressive enhancement and graceful degradation?

Answer

Progressive enhancement and graceful degradation are essentially the same techniques, but with opposite "directions". Progressive enhancement is the process of enhancing baseline experience for newer, more advanced platforms; graceful degradation — providing fallbacks for older ones.

How would you optimize a website's assets/resources?

Answer

Consolidate (concatenate CSS and JavaScript, compile images into sprites), minify, optimize images, GZip, optimize fonts, serve everything in a non blocking way.

How many resources will a browser download from a given domain at a time?

Answer

- Internet Explorer 7 — 2
- More current browsers — from 6 to 8.

Name 3 ways to decrease page load (perceived or actual load time).

Answer

- Compress and optimize resources
- Stylesheets in the `<head>`
- Scripts at the bottom of the page
- Minimize HTTP requests
- Setup caching
- No redirects
- Serve content in a non blocking way

If you jumped on a project and they used tabs and you used spaces, what would you do?

Answer

Use my .editorconfig

Explain the importance of standards and standards bodies.

Answer

Web standards are the open specifications to enhance the interoperability of web-related products, achieve a more stable web, reduce development and maintenance time, allow backward compatibility and validation.

Standards specifications basically are roadmaps for developing instruments.

Standards bodies (namely W3C) are consortiums of companies and independent developers assembled to work on standards specifications.

What is Flash of Unstyled Content? How do you avoid FOUC?

Answer

- Inlined critical CSS
- Load JavaScript in a non blocking way & asynchronously
- Load fonts in a non blocking way

Explain what ARIA and screenreaders are, and how to make a website accessible.

Answer

ARIA — Accessible Rich Internet Applications — provides a way to add the missing

semantics needed by assistive technologies such as screen readers. ARIA enables developers to describe their widgets in more detail by adding special attributes to the markup. Designed to fill the gap between standard HTML tags and the desktop-style controls found in dynamic web applications, ARIA provides roles and states that describe the behavior of most familiar UI widgets.

Explain some of the pros and cons for CSS animations versus JavaScript animations.

Answer

Pros:

- Hardware acceleration
- Declarative

Cons:

- Less control
- Only in newer browsers.

What does CORS stand for and what issue does it address?

Answer

Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources (e.g. fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain from which the resource originated.

CORS defines a way in which a browser and server can interact to safely determine whether or not to allow the cross-origin request. It allows for more freedom and functionality than purely same-origin requests, but is more secure than simply allowing all cross-origin requests. It is a recommended standard of the W3C.

What's a doctype do?

Answer

Doctype declaration is not an HTML tag; it is an instruction for the browser telling it what version of the markup language the page is written in.

What's the difference between standards mode and quirks mode?

Answer

In quirks mode, layout emulates nonstandard behavior in Navigator 4 and Internet Explorer 5 for Windows that is required not to break existing content on the Web.

In full standards mode, browser's behavior is (hopefully) the behavior described by the HTML and CSS specifications.

What are the limitations when serving XHTML pages?

Answer

- Browser support
- HTML5 is far more developed

Are there any problems with serving pages as application/xhtml+xml?

Answer

- Some browser won't accept XML
- Syntax errors are critical

How do you serve a page with content in multiple languages?

Answer

- Set primary language (e.g. `<html lang="en">`)
- Mark change in language (e.g. `...`)
- Mark change in language on the linked page (e.g. `<a href lang="fr">...`)

What kind of things must you be wary of when design or developing for multilingual sites?

Answer

- Language directionality
- Font sizes
- Length of words

What are data- attributes good for?

Answer

For binding data to the HTML elements.

Consider HTML5 as an open web platform. What are the building blocks of HTML5?

Answer

Different standards and APIs.

- More semantic text markup
- New form elements
- Video, audio and canvas
- New JavaScript APIs (Communication, geolocation, Web workers, WebGL, Storage, & etc.)

Describe the difference between cookies, sessionStorage and localStorage.

Answer

- **localStorage** — persistent and scoped to the domain, allows to store up to 5 mb of data.
- **sessionStorage** — non persistent and scoped only to the current window, allows to store up to 5 mb of data.
- **cookies** — stores name/value pairs per domain, may expire, allows to store up to 4 kb of data.

Describe the difference between <script>, <script async> & <script defer>.

Answer

- Scripts with **async** directive execute right away regardless of status of other declared scripts
- Scripts with **defer** directive will wait until HTML is parsed and then executed in order

Why is it generally a good idea to position CSS link tags in the head and JavaScript script tags just before /body? Do you know any exceptions?

Answer

<link>'s in the **<head>** so that CSS resources will be downloaded right away, and site will be rendered (and generally available) as soon as possible.

<script>'s just before **</body>** so that page rendering isn't blocked while downloading, parsing and running JavaScript resources, also because that guarantees that DOM will be fully constructed before it could be potentially changed by those scripts.

This technique could be disregarded when impact of loading and executing scripts before rendering the page is negligible, but styling or/and some aspects of performance (eg. extra resources) are depending on it (Modernizr classes).

What is progressive rendering?

Answer

Progressive rendering is the name given to techniques used to render content for display as quickly as possible.

It used to be much more prevalent in the days before broadband internet but it's still useful in modern development as mobile data connections are becoming increasingly popular (and unreliable).

Examples of such techniques :

- Lazy loading of images where (typically) some javascript will load an image when it comes into the browser's viewport instead of loading all images at page load.
- Prioritizing visible content (or above the fold rendering) where you include only the minimum css/content/scripts necessary for the amount of page that would be rendered in the user's browser first to display as quickly as possible, you can then use deferred javascript (domready/load) to load in other resources and content.

What is the difference between classes and IDs in CSS?

Answer

IDs are unique.

What's the difference between "resetting" and "normalizing" CSS? Which would you choose, and why?

Answer

Establishes shared baseline between browser by removing all style inconsistencies.

Describe floats and how they work.

Answer

Remove blocks from the flow of the document and position them to the left or right inside of the container. Content in the flow positions itself around content in the floated blocks.

Describe z-index and how stacking context is formed.

Answer

Natural stacking order

1. Background and borders of the element that establish stacking context
2. Elements with negative stacking contexts, in order of appearance
3. Non-positioned, non-floated, block-level elements, in order of appearance
4. Non-positioned, floated elements, in order of appearance
5. Inline elements, in order of appearance
6. Positioned elements, in order of appearance

z-index property changes stacking order of the element within its stacking context and can be applied only to the explicitly positioned elements.

Describe BFC (Block Formatting Context) and how it works

Answer

A block formatting context is a part of a visual CSS rendering of a web page. It is the region in which the layout of block boxes occurs and in which floats interact with each other.

A block formatting context is created by one of the following:

- the root element or something that contains it

- floats (elements where float is not none)
- absolutely positioned elements (elements where position is absolute or fixed)
- inline-blocks (elements with display: inline-block)
- table cells (elements with display: table-cell, which is the default for HTML table cells)
- table captions (elements with display: table-caption, which is the default for HTML table captions)
- elements where overflow has a value other than visible
- flex boxes (elements with display: flex or inline-flex)

A block formatting context contains everything inside of the element creating it that is not also inside a descendant element that creates a new block formatting context.

Block formatting contexts are important for the positioning (see float) and clearing (see clear) of floats. The rules for positioning and clearing of floats apply only to things within the same block formatting context. Floats do not affect the layout of things in other block formatting contexts, and clear only clears past floats in the same block formatting context.

What are the various clearing techniques and which is appropriate for what context?

Answer

- `overflow:hidden`
- Float
- Clearing block or pseudo-element.

Explain CSS sprites, and how you would implement them on a page or site.

Answer

Compiled file consists of smaller images used as parts of site's design. Useful to reduce

time and requests needed to load graphic resources. Usually implemented using `background` CSS property.

What are your favourite image replacement techniques and which do you use when?

Answer

- Using `padding-top`
- Using `text-indent` (when height varies)

[CSS Tricks Article](#)

How would you approach fixing browser-specific styling issues?

Answer

- Conditionally included CSS-files
- Modernizr classes in rare cases
- User agent sniffing, but very-very rarely
- `@supports`

How do you serve your pages for feature-constrained browsers? What techniques/processes do you use?

Answer

Ways

- Progressive enhancement
- Graceful degradation

Techniques

- HTML5 doctype
- Modernizr
- IE conditional comments (stylesheets)
- Widely supported defaults
- Autoprefixer
- CSS preprocessor mixins

What are the different ways to visually hide content (and make it available only for screen readers)?

Answer

- `text-indent: -9000px`
- `height: 0`, `overflow: hidden` and `padding-top`

How do you optimize your webpages for print?

Answer

- Print styles & typography
- Turn off all colors and backgrounds
- Hide online-only elements (widgets, social links & etc.)
- Output URLs using pseudo-elements

What are some of the "gotchas" for writing efficient CSS?

Answer

- Avoid a universal key selector.
- In general keep selector specificity low
- Don't use ids

- Avoid CSS expressions
- Use efficient CSS selectors
- Put CSS at the top of the document

What are the advantages/disadvantages of using CSS preprocessors? Describe what you like and dislike about the CSS preprocessors you have used.

Answer

Advantages:

- efficiency
- speed
- flexibility

Disadvantages:

- need to setup an environment
- may output bad CSS
- potential future support limitations

How would you implement a web design comp that uses non-standard fonts?

Answer

- Optimize using Fontsquirrel
- Load using FontFaceObserver
- Don't like using Google Fonts because of the additional requests

Explain how a browser determines what elements match a CSS selector

Answer

As the browser parses HTML, it constructs an internal document tree representing all the elements to be displayed. It then matches elements to styles specified in various stylesheets, according to the standard CSS cascade, inheritance, and ordering rules. In Mozilla's implementation (and probably others as well), for each element, the CSS engine searches through style rules to find a match. The engine evaluates each rule from right to left, starting from the rightmost selector (called the "key") and moving through each selector until it finds a match or discards the rule.

[Describe pseudo-elements and discuss what they are used for.](#)

Answer

Just like pseudo-classes, pseudo-elements are added to selectors but instead of describing a special state, they allow you to style certain parts of a document. For example, the `::first-line` pseudo-element targets only the first line of an element specified by the selector.

- `::after` – matches a virtual last child of the selected element
- `::before` – matches a virtual first child of the selected element
- `::first-letter` – selects the first letter of the first line of a block
- `::first-line` – applies styles only to the first line of an element
- `::selection` – portion of a document that has been highlighted

[Explain your understanding of the box model and how you would tell the browser in CSS to render your layout in different box models.](#)

Answer

- `box-sizing`
- Appropriate `doctype` for older Internet Explorer versions

What does * { box-sizing: border-box; } do? What are its advantages?

Answer

Switches all the block elements to the new box model.

Don't have to worry about paddings and borders breaking layout.

What's the difference between inline and inline-block?

Answer

Elements with `display:inline-block` can be used like `display:block` while flowing within text. `display:inline` ignore width and height properties.

What's the difference between a relative, fixed, absolute and statically positioned element?

Answer

`position: static` — default positioning of the element.

`position: relative` — element remains in the natural flow of the document, but can be (visually) 'nudged' from its position (using `top`, `right`, `bottom` and `left` properties) and moved in the current stacking context (using `z-index` property).

`position: absolute` — element positions itself independently from the natural flow of the document within nearest relatively positioned parent. Its position in the current stacking order can be changed explicitly.

`position: fixed` — same as `position: absolute`, only element positions itself within window boundaries.

The 'C' in CSS stands for cascading. How is priority determined in assigning styles (a few examples)? How can you use this system to your advantage?

Answer

"Cascading" in this context means that because more than one stylesheet rule could apply to a particular piece of HTML, there has to be a known way of determining which specific stylesheet rule applies to which piece of HTML.

The rule used is chosen by cascading down from the more general rules to the specific rule required. The most specific rule is chosen.

How is responsive design different from adaptive design?

Answer

Adaptive design "adapts" for different specifically set resolutions (devices), responsive design uses available space fluidly.

Have you ever worked with retina graphics? If so, when and what techniques did you use?

Answer

- Scaling graphics with `background-size`
- Using SVG images for CSS-images
- Responsive images polyfills for content graphics

Is there any reason you'd want to use `translate()` instead of absolute positioning, or vice-versa? And why?

Answer

When block's position isn't animated it is better to use absolute positioning because its more precise.

When block's position is animated it is better to use `transform` (`translate()`) for positioning because it performs much better (saves extra layouts and paints).

Explain event delegation.

Answer

Attach event listener to the parent node and then check if this particular event gets triggered through event propagation.

Explain how this works in JavaScript.

Answer

`this` points to the current execution context (scope).

Explain how prototypal inheritance works.

Answer

When it comes to inheritance, JavaScript only has one construct: objects. Each object has an internal link to another object called its prototype. That prototype object has a prototype of its own, and so on until an object is reached with null as its prototype. null, by definition, has no prototype, and acts as the final link in this prototype chain.

JavaScript objects are dynamic "bags" of properties (referred to as own properties). JavaScript objects have a link to a prototype object. When trying to access a property of an object, the property will not only be sought on the object but on the prototype of the object, the prototype of the prototype, and so on until either a property with a matching name is found or the end of the prototype chain is reached.

Explain why the following doesn't work as an IIFE: `function foo(){ }();`.

Answer

`function foo(){ }` — is the declaration of the `foo()`. Everything else is syntax error.

What needs to be changed to properly make it an IIFE?

What needs to be changed to properly make it an IIFE?

Answer

```
(function foo()){ }
```

By wrapping the function inside of parentheses, the JavaScript parser knows to treat the anonymous function as a function expression instead of a function declaration.

What's the difference between a variable that is: null, undefined or undeclared?

Answer

- `null` is a value that represents no value
- `undefined` shows that declared variable wasn't assigned any value
- variable is undeclared when it does not use the `var` keyword. It gets created on the global object (the window)

How would you go about checking for any of these states?

Answer

For `null` and `undefined` you can check if value is falsey — `if (!variable) {}`

Null and undefined are the same value but different types:

```
null == undefined // true
null === undefined // false
```

What is a closure, and how/why would you use one?

Answer

A closure is an inner function that has access to the outer (enclosing) function's variables — scope chain.

For example you can create closures using module pattern:

```
var module = (function(){
  foo = function() {}
})();
module.foo();
```

Closure is a way to encapsulate module and to not pollute global scope.

What's a typical use case for anonymous functions?

Answer

- Callbacks
- Closures

What's the difference between host objects and native objects?

Answer

Native object is the object in an ECMAScript implementation whose semantics are fully defined by this specification rather than by the host environment.

Host object supplied by the host environment to complete the execution environment of ECMAScript.

Difference between (code is in the description)

Code

1. `function Person(){},`
2. `var person = Person()`
3. `var person = new Person()`

Answer

1. Declaration of a function
2. Function gets executed and result is assigned to the variable
3. Browser creates new instance of a "class", and assigns it to a variable

What's the difference between .call and .apply?

Answer

Both `.apply` and `.call` bind certain context to the function (`this`) they called upon — first parameter. But `.apply` gets other parameters as a single array, `.call` — as usual.

Explain Function.prototype.bind.

Answer

The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called

- `call()` and `apply()` execute function
- `bind()` creates new to execute later

When would you use document.write()?

Answer

Pros

- it's the easiest way to embed inline content from an external script (third party libraries).

Cons

- it only works while the page is loading; after the page is done loading, it will

overwrite the whole page.

- it doesn't work in XHTML
- it's slow

What's the difference between feature detection, feature inference, and using the UA string

Answer

Feature detection checks a feature for existence:

```
if (window.XMLHttpRequest) {  
    new XMLHttpRequest();  
}
```

Feature inference checks for a feature just like feature detection, but uses another function because it assumes it will also exist:

```
if (document.getElementsByTagName) {  
    element = document.getElementById(id);  
}
```

Checking the UA string is an old practice and should not be used anymore. You have to keep changing the UA checks:

```
if (navigator.userAgent.indexOf("MSIE 7") > -1){  
    //do something  
}
```

Explain AJAX in as much detail as possible.

Answer

AJAX stands for Asynchronous JavaScript and XML.

Used to send requests and handle responses from the server without disrupting reloading current page. Requires an URL to send the request to, which can't point to the

different domain (same origin policy).

Requests use the method `get` by default, but can be changed by changing the type.

Have success and error callbacks.

AJAX Steps:

1. A client event occurs.
2. An XMLHttpRequest object is created.
3. The XMLHttpRequest object is configured.
4. The XMLHttpRequest object makes an asynchronous request to the Webserver.
5. The Webserver returns the result containing XML document.
6. The XMLHttpRequest object calls the `callback()` function and processes the result.
7. The HTML DOM is updated.

Explain how JSONP works (and how it's not really AJAX).

Answer

JSONP is a simple trick to overcome the **XMLHttpRequest** same domain policy (you can't make an **AJAX** request to a different domain).

Instead of using **XMLHttpRequest** **JSONP** uses **script** HTML tags.

```
script = document.createElement('script');
script.type = 'text/javascript';
script.src = 'http://www.someWebApiServer.com/some-data
?callback=my_callback';
```

When **JSONP** server receives your request and finds callback parameter, it'll return:

```
my_callback({['data', 'data', ...]});
```

Explain "hoisting".

Answer

Within its current scope, regardless of where a variable is declared, it will be, behind the scenes, hoisted to the top. However, only the declaration will be hoisted. If the variable is also initialized, the current value, at the top of the scope, will initially be set to undefined.

Describe event bubbling.

Answer

Event bubbling and capturing are two ways of event propagation in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. The event propagation mode determines in which order the elements receive the event.

With bubbling, the event is first captured and handled by the innermost element and then propagated to outer elements.

With capturing, the event is first captured by the outermost element and propagated to the inner elements.

What's the difference between an "attribute" and a "property"?

Answer

Attributes are defined by HTML. Properties are defined by DOM.

Some HTML attributes have 1:1 mapping onto properties. `id` is one example of such.

Some do not – the `value` attribute specifies the initial value of an `input`, but the `value` property specifies the current value.

Why is extending built in JavaScript objects not a good idea?

Answer

Other libraries, and even client, can easily be affected. This is especially true for the Object prototype as everything in Javascript extends from it.

Browsers that don't support element extensions — like IE 6, 7, Safari 2.x, etc. — require manual object extension. The problem is that manual extension is slow, inconvenient and doesn't scale.

Difference between document load event and document ready event?

Answer

The **ready** event occurs after the HTML document has been loaded, while the **onload** event occurs later, when all content also has been loaded.

The **onload** event is a standard event in the DOM, while the **ready** event is specific to jQuery.

document.ondomcontentready / **document.ondomcontentloaded** — a new event which fires when the document's DOM is loaded. It's not available in all browsers.

What is the difference between == and ===?

Answer

The equality operator **==** converts the operands if they are not of the same type, then applies strict comparison.

```
1 == 1           // true
'1' == 1         // true
1 == '1'         // true
0 == false       // true
0 == null        // false
0 == undefined   // false
null == undefined // true
```

The identity operator **===** returns true if the operands are strictly equal (see above) with no type conversion.

Explain the same-origin policy with regards to JavaScript.

Answer

It is a critical security mechanism for isolating potentially malicious documents.

Only scripts that are executed on the same domain can access each others objects and properties without restriction. If scripts are served from different domains some restriction do apply (can't make an XHR request originating from one domain to the other).

The "origin" is the same if three things are the same: the protocol, the domain, and the port.

Make this work (code is in description):

Code

```
duplicate([1,2,3,4,5]); // [1,2,3,4,5,1,2,3,4,5]
```

Answer

```
function duplicate(array) {  
    return array.concat(array);  
}
```

Why is it called a ternary expression, what does the word "ternary" indicate?

Answer

It's a conditional operator that takes 3 parameters

```
condition ? expression 1 : expression 3
```

What is "use strict";? what are the advantages and disadvantages to

using it?

Answer

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program, or a function, in a "strict" operating context. This strict context prevents certain actions from being taken and throws more exceptions.

- it catches some common coding bloopers, throwing exceptions.
- it prevents, or throws errors, when relatively "unsafe" actions are taken (such as gaining access to the global object).
- it disables features that are confusing or poorly thought out.

Browsers not supporting strict mode will run strict mode code with different behavior from browsers that do, so don't rely on strict mode without feature-testing for support for the relevant aspects of strict mode.

Create a for loop that iterates up to 100 while outputting "fizz" at multiples of 3, "buzz" at multiples of 5 and "fizzbuzz" at multiples of 3 and 5.

Answer

```
for (i = 0; i < 100; i++) {  
  if (i >= 3) {  
    if (i % 3 === 0 && i % 5 === 0) {  
      console.log("fizzbuzz");  
    } else if (i % 3 === 0) {  
      console.log("fizz");  
    } else if (i % 5 === 0) {  
      console.log("buzz");  
    }  
  }  
}
```

Why is it, in general, a good idea to leave the global scope of a website as-is and never touch it?

Answer

- It's harder to read the code and reason about.
- Global variable name clashes and conflicts.

What are the pros and cons of using Promises instead of callbacks?

Answer

Promises make it easier for more than one function to be called at once. Promises allow functions to be called anywhere in code, subject only to the promise being within the scope.

For code that relies on short local reactions to asynchronous events, callbacks are the better option. For more complex interactions and data flows, promises bring order to what could potentially be callback hell.

What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?

Answer

Pros

Compiler can validate the integrity of your code before runtime and large scale refactorings with integrity.

Cons

Potentially complex workflow.

What language constructions do you use for iterating over object properties and array items?

Answer

Objects

- `for in` loop
- `Object.keys()`

Arrays

- `for` loop
- `for each` loop
- `Array.map()`

Explain the difference between mutable and immutable objects.

Answer

A mutable object is an object whose state is allowed to change over time. An immutable object is an object that can never change after it has been created

What is an example of an immutable object in JavaScript?

Answer

An example of an immutable object in JavaScript is strings. Strings are immutable because they cannot change. We can only create new strings.

Numbers are also immutable objects.

What are the pros and cons of immutability?

Answer

Pros:

- programs with immutable objects are less complicated to think about, since you don't need to worry about an object may evolve over time

- immutable objects are good for sharing information between threads in a multithreaded environment
- in languages where functions are first class values, operations like map, reduce, filter, etc. are basic operations on collections. These can be combined in many ways, and can replace most loops in a program.

Cons:

- allocating lots and lots of small objects rather than modifying ones you already have can have a performance impact. Usually the complexity of either the allocator or the garbage collector depends on the number of objects on the heap.
- cyclic data structures such as graphs are difficult to build. If you have two objects which can't be modified after initialization, how can you get them to point to each other?

How can you achieve immutability in your own code?

Answer

In ES2015 `const` could be used.

In earlier versions of JavaScript there are no immutable lists and maps, so 3rd party libraries are used.

What is event loop?

Answer

The browser has an inner loop, called the event loop, which checks the queue and processes events, executes functions and other tasks.

Explain "chaining".

Answer

Chaining allows to run multiple jQuery methods (on the same element) within a single statement.

Explain "deferreds"

Answer

Deferred object provides a way to register multiple callbacks into self-managed callback queues, invoke callback queues as appropriate, and relay the success or failure state of any synchronous or asynchronous function.

Basically it is a jQuery implementation of the promises concept.

[HTML5 Rocks Tutorial](#)

What are some jQuery specific optimizations you can implement?

Answer

- Caching jQuery objects
- Don't create extra jQuery objects. Use `this` instead of `$(this)`
- Descending from the ID selector is better for performance

What does .end() do?

Answer

Returns set of matched elements in filtering operation to its previous state.

Name 4 different values you can pass to the jQuery method.

Answer

- jQuery object
- DOM element
- Array of DOM elements
- String

[What is the difference between .get(), [], and .eq()?]
(<https://trello.com/c/9l6wXV0l/113-what-is-the-difference-between-get-and-eq>)

Answer

`get()` and `[]` return DOM object

`eq()` returns jQuery object

What tools would you use to find a performance bug in your code?

Answer

- Chrome DevTools
- ySlow
- WebPagetest
- analyze-css

Can you explain the difference between GET and POST?

Answer

GET is used to retrieve remote data, and **POST** is used to insert/update or remote data.

Traditionally, why has it been better to serve site assets from multiple domains?

domains:

Answer

To circumvent browser limitation on opening simultaneous connections to one URI. And to serve resources from cookieless domains.

Do your best to describe the process from the time you type in a website's URL to it finishing loading on your screen.

Answer

- Browser resolves domain name and sends a request to open connection; server responds and, if connection is allowed, browser sends request for the specific data
- Server, if data is found, sends it back to the browser in the form of HTML-file
- Browser parses HTML and, while parsing it, requests more resources (CSS, JavaScript, images) referenced in it from the server
- As CSS and images become available browser starts rendering page
- When JavaScript become available browser stops rendering process, executes JavaScript and then continues rendering page

What are the differences between Long-Polling, Websockets and SSE?

Answer

Ajax Long-Polling:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which requests a file from the server
3. The server does not immediately respond with the requested information but waits until there's new information available
4. When there's new information available, the server responds with the new information
5. The client receives the new information and immediately sends another request to the server, re-starting the process

HTML5 Server Sent Events (SSE) / EventSource:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection to the server
3. The server sends an event to the client when there's new information available

HTML5 Websockets:

1. A client requests a webpage from a server using regular HTTP
2. The requested webpage executes JavaScript which opens a connection with the server
3. The server and the client can now send each other messages when new data (on either side) is available

[StackOverflow Answer](#)

Explain difference between Expires, Date, Age and If-Modified-... Headers

Answer

Expires – sets a date from which the cached resource should no longer be considered valid.

Date – date and time at which the message was originated.

Age – gives the sender's estimate of the amount of time since the response (or its revalidation) was generated at the origin server.

If-Modified-... – Used with a method to make it conditional: if the requested variant has not been modified since the time specified in this field, the server will not return the entity but information about this fact.`

Explain the DNT (Do Not Track) Header

Answer

Do not track. Requests a web application to disable their tracking of a user

Explain the Cache-Control Header

Answer

Controls content caching mechanisms – if something should be cached, for how long, etc..

Explain the Transfer-Encoding Header

Answer

Instead of processing the whole page, generating all of the html and sending it to the client, we can split the html into 'chunks' and send one after the other, without telling the browser how big the response will be ahead of time. To notify the browser about the chunked response, you need to omit the **Content-Length** header, and add the header **Transfer-Encoding: chunked**. Giving this information to the browser, the browser will now expect to receive the chunks in a very specific format.

Explain the ETag Header

Answer

Entity tag – used to check if cached resource has changed.

Explain the X-Frame-Options Header

Answer

Clickjacking protection. The header, when set by website owner, declares its preferred framing policy.

What is the value of foo (code is in description)?

Code

```
var foo = 10 + '20';
```

Answer

```
'1020'
```

How would you make this work (code is in description)?

Code

```
add(2, 5); // 7  
add(2)(5); // 7
```

Answer

```
function add(a, b) {  
  if (arguments.length > 1) {  
    return a+b;  
  } else {  
    return function(c) {  
      return a + c;  
    }  
  }  
}
```

What value is returned from the following statement (code is in description)?

Code

```
"i'm a lasagna hog".split("").reverse().join("");
```

Answer

"goh angasal a m'i"

What is the value of window.foo (code is in description)?

Code

```
(window.foo || ( window.foo = "bar" ));
```

Answer

bar

What is the outcome of the two alerts below (code is in description)?

Code

```
var foo = "Hello";  
(function() {  
    var bar = " World";  
    alert(foo + bar);  
})();  
alert(foo + bar);
```

Answer

"Hello World"

Error – **bar** in undefined (out of scope)

What is the value of foo.length (code is in description)?

Code

```
var foo = [];  
foo.push(1);  
foo.push(2);
```

Answer

2

What is the value of foo.x (code is in description)?

Code

```
var foo = {n: 1};  
var bar = foo;  
foo.x = foo = {n: 2};
```

Answer

undefined

What does the following code print (code is in description)?

Code

```
console.log('one');  
setTimeout(function() {  
  console.log('two');  
}, 0);  
console.log('three');
```

Answer

one
three
two

Explanation

`setTimeout(function() {...}, 0)` is pausing JavaScript execution (useful to let the rendering threads catch up). Although it seems to say "run this immediately" it actually gives the browser a chance to finish doing some non-JavaScript things that have been waiting to finish before attending to this new piece of JavaScript.

Front-end Developer Interview Questions

<https://github.com/h5bp/Front-end-Developer-Interview-Questions>