**PREGUNTAS**

**¿Qué herramientas consideras esenciales a la hora de empezar un proyecto frontend?**

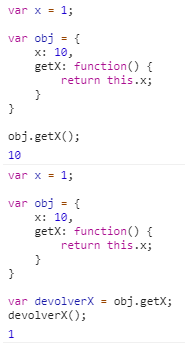
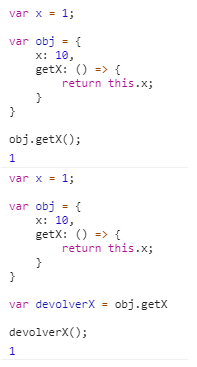
Considero que las herramientas esenciales para comenzar un desarrollo frontend son:

* Un editor de código amigable que permita instalación de addons (Visual studio code, Atom, sublime text, etc.)
* Algún sistema de control de versiones para tener controlados los cambios que se realicen (Git, Bitbucket).
* Boilerplates (No obligatorios) que pueden ayudar a reducir el tiempo de inicio del proyecto.
* Assets (Imágenes, fuentes, iconos).
* Frameworks, ej: Bootstrap. Nos ayudan a no tener que diseñar y escribir código de elementos por nuestra cuenta (Salvo que querramos hacer algo totalmente personalizado).
* Preprocesadores CSS: Como Sass o Less, para facilitarnos la visualización de estilos (Usando nesting por ejemplo).
* Bibliotecas de Javascript: Proveen mejor soporte, agregan componentes y funcionalidades que nos pueden ser útiles y nos simplifican la escritura de nuestro sitio web. Ej: React JS, jQuery, Vue Js, entre otras.
* Dev tools(Chrome Dev Tools).

**¿Puedes explicar cómo el bindeo de “this” es determinado? Dar ejemplos**

**this** es bindeado según la forma en la que se invoca a una función. No puede ser establecida mediante una asignación en tiempo de ejecución, y puede ser diferente cada vez que la función es invocada. ES5 introdujo el método **bind()** para que el valor de **this** sea independiente de la llamada, y luego surgieron las arrow functions que no proporcionan su propio binding de **this**. (Se mantiene el valor del contexto).

Ejemplos en la siguiente hoja:

**¿Qué es webpack y para que se usa? ¿Conoces otras herramientas similares?**

Webpack es un bundler de módulos que se utiliza para desarrollar aplicaciones web diseñadas en forma modular, en donde se separa el código en módulos que luego se utilizan como dependencias en otros módulos (Ya sean en javascript, html, css, e incluso con imágenes). También, se lo suele usar como minificador de código.

Como alternativas Gulp, Babel, Grunt, Browserify.

**¿Qué es Server Side Rendering y por qué es útil? ¿Puedes explicar brevemente como implementarlo en React?**

SSR consiste en que la página web se renderice del lado del servidor, para entregarle luego, la página ya construida al cliente.

Esto evita que el navegador del cliente tenga que cargar dicha aplicación, haciendo que la experiencia de usuario sea más fluida, tanto en la carga como en la interacción.

Es beneficioso, además, ya que se asiste en la carga de la página cuando el usuario tiene una conexión lenta, o también, cuando este posee un dispositivo desactualizado.

Cabe aclarar que SSR va a beneficiar al rendimiento en aplicaciones relativamente pequeñas.

En React, se debe utilizar el objeto ReactDOMServer, que se usa generalmente con un servidor Node (Podría utilizar express.js para levantarlo).

Se crearía una petición http, por ejemplo, a la url ("/"), donde se ejecutará el método ReactDOMServer.renderToString, que renderiza un elemento React a su HTML inicial.

Luego, devuelvo esta cadena HTML al cliente, quien visualiza esta página ya renderizada. Si se llama a ReactDOM.hydrate() a un nodo que ya tiene este marcado desde el servidor, React lo conservará y solo adjuntará los controladores de eventos, lo que te permitirá tener una experiencia de primera carga muy eficaz.

**¿Puedes explicar brevemente como funciona ReactJs Fiber?**

React fiber se basa en el algoritmo de "Reconciliacion" de React. Su función principal es la de permitir el renderizado incremental del Virtual DOM.

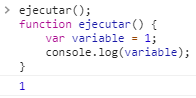
Funciona de la siguiente manera: Cuando se renderiza una aplicación, se guarda un árbol de nodos que describe nuestra app es generado y guardado en memoria. Luego, es trasladado

al entorno de renderizado (Por ejemplo, un set de operaciones del DOM). Cuando la app es actualizada, un nuevo árbol es generado. Este es comparado con el árbol previo para obtener las operaciones a realizar para actualizar la app.

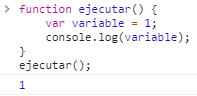
**¿Puedes explicar JS hoisting con un ejemplo?**

JS hoisting consiste en el comportamiento de mover al principio del scope, las declaraciones.

**Ejemplo:**



Esto se ejecutará correctamente ya que JS, estará haciendo lo siguiente:



**¿Cómo implementarías currying utilizando closures? Dar un ejemplo**

Son implementados mediante encadenamiento de closures definiendo y retornando sus funciones internas simultáneamente. Es decir, funciones que retornan otras funciones, y que aprovechan parámetros del closure de la función que retorna a la otra función.

**Ejemplo**:

