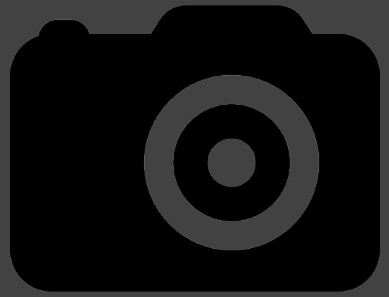
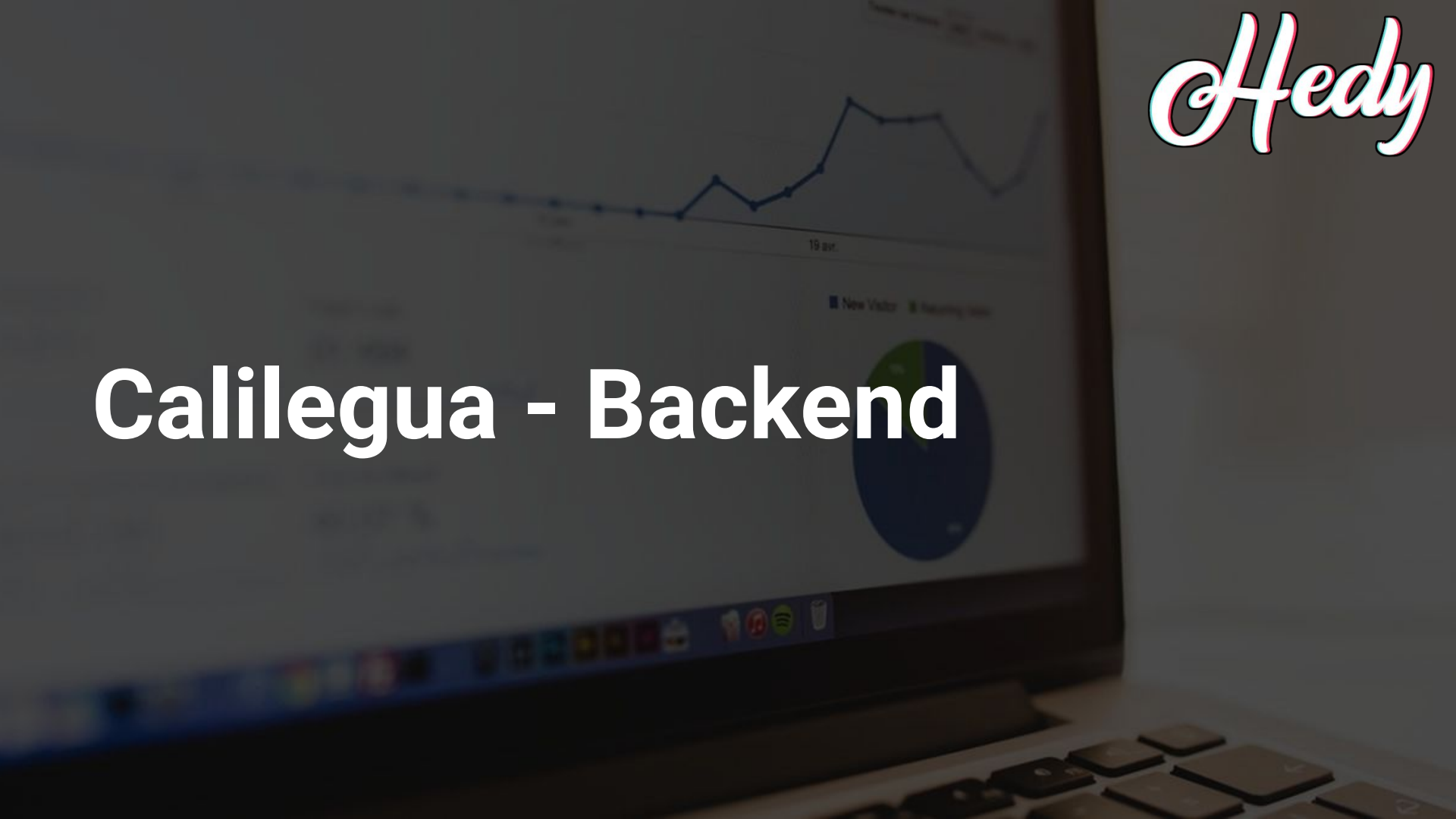
Hedy

*RE*Cordá poner a grabar la clase





Calilegua - Backend

Hedy

JavaScript

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos. Es ampliamente utilizado en el desarrollo web para agregar interactividad y funcionalidad a las páginas web. JavaScript se ejecuta en el lado del cliente, lo que significa que el código se ejecuta en el navegador web del usuario, lo que le permite interactuar dinámicamente con el contenido de la página y responder a eventos del usuario.

JavaScript

Con JavaScript, los desarrolladores pueden crear efectos visuales, validar formularios, cargar contenido de manera asíncrona, crear juegos en línea, entre muchas otras cosas. Además de su uso en el desarrollo web, JavaScript también se ha expandido a otros ámbitos, como el desarrollo de aplicaciones móviles y de escritorio, gracias a frameworks como React Native y Electron. En resumen, JavaScript es un lenguaje versátil y poderoso que ha sido fundamental para la evolución de la web moderna.

TypeScript

TypeScript es un lenguaje de programación de código abierto desarrollado y mantenido por Microsoft. Se basa en JavaScript pero agrega características adicionales que facilitan el desarrollo y mantenimiento de aplicaciones a gran escala.

TypeScript

Algunas de las características principales de TypeScript incluyen:

- **Tipado estático opcional**: TypeScript permite especificar tipos de datos para variables, parámetros de funciones y otros elementos del código. Esto ayuda a detectar errores en tiempo de compilación y mejora la legibilidad y mantenibilidad del código.
- **Adopción gradual:** Como TypeScript es un superconjunto de JavaScript, significa que todo código JavaScript válido también es código TypeScript. Esto facilita la adopción gradual de TypeScript en proyectos existentes.
- **Soporte para características de ECMAScript más recientes:** TypeScript brinda soporte para las últimas características de ECMAScript (el estándar en el que se basa JavaScript), lo que permite a los desarrolladores utilizar sintaxis moderna incluso si el entorno de destino no la admite directamente.

TypeScript

- **Mejoras en el desarrollo de software:** TypeScript ofrece características adicionales como interfaces, clases, módulos, decoradores, y más, que no están presentes en JavaScript estándar. Estas características facilitan la organización y estructuración del código, así como la implementación de patrones de diseño comunes.
- **Herramientas de desarrollo:** TypeScript se integra bien con una variedad de herramientas de desarrollo, incluidos editores de código como Visual Studio Code, lo que proporciona sugerencias de autocompletado, detección de errores en tiempo real y refactoring más sólidos.

Hedy

TypeScript

¿Por qué se debería TypeScript?

Es posible encontrar en diferentes sitios muchos argumentos a favor o en contra de TypeScript. Mucho depende de las necesidades y del uso de las funciones que ofrece TypeScript. Algunas razones a nuestro criterio por las que pensamos que el uso de TypeScript puede tener algunas ventajas son:

- En primer lugar, TypeScript ofrece verificación de tipos y análisis de código estático. Podemos requerir que los valores sean de cierto tipo y que el compilador advierta sobre su uso incorrecto. Esto puede reducir los errores durante el tiempo de ejecución, e incluso es posible que pueda reducir la cantidad de pruebas unitarias requeridas en un proyecto, al menos en lo que respecta a las pruebas de tipo puro. El análisis de código estático no sólo advierte sobre el uso incorrecto de tipos, sino también sobre otros errores, como escribir mal el nombre de una variable o función o intentar usar una variable más allá de su scope.

TypeScript

- La segunda ventaja de TypeScript es que las anotaciones de tipo en el código pueden funcionar como una especie de documentación a nivel de código. Es fácil verificar a partir de la firma de una función qué tipo de argumentos puede consumir y qué tipo de datos devolverá. Esta forma de documentación vinculada con anotaciones de tipo siempre estará actualizada y facilita a los nuevos programadores comenzar a trabajar en un proyecto existente. También es útil al volver a trabajar en un proyecto antiguo.
- Los tipos se pueden reutilizar en toda la base de código, y un cambio en una definición de tipo se reflejará automáticamente en todos los lugares donde se use el tipo. Se podría argumentar que puedes lograr una documentación de nivel de código similar con, por ejemplo, JSDoc, pero no está conectado al código tan estrechamente como los tipos de TypeScript y, por lo tanto, puede salir de sincronización más fácilmente y también es más verboso.

TypeScript

- La tercera ventaja de TypeScript es que los IDE pueden proporcionar IntelliSense más específico e inteligente cuando saben exactamente qué tipos de datos estás procesando.

Todas estas características son extremadamente útiles cuando necesitas refactorizar tu código. El análisis de código estático te advierte sobre cualquier error en tu código, y el IntelliSense puede darte pistas sobre las propiedades disponibles e incluso posibles opciones de refactorización. La documentación a nivel de código te ayuda a comprender el código existente. Con la ayuda de TypeScript, también es muy fácil comenzar a usar las funciones más nuevas del lenguaje JavaScript en una etapa temprana, simplemente modificando su configuración.

GIT

Git es un sistema de control de versiones distribuido ampliamente utilizado en el desarrollo de software. Fue creado por Linus Torvalds en 2005 para el desarrollo del kernel de Linux, pero desde entonces se ha convertido en una herramienta fundamental en la industria del software para el seguimiento de cambios en el código fuente de proyectos.
En términos simples, Git permite a los desarrolladores trabajar en colaboración en un mismo proyecto, registrando los cambios realizados en el código a lo largo del tiempo. Esto facilita la gestión de versiones, la colaboración entre equipos de desarrollo, la reversión de cambios y la implementación de nuevas características de manera organizada y controlada.

GIT

Algunas de las características clave de Git incluyen:

- **Sistema distribuido:** Cada usuario que trabaja en un repositorio de Git tiene una copia completa del historial de cambios, lo que permite trabajar de forma descentralizada y sin depender de un servidor central.
- **Ramificación (branching):** Git permite a los desarrolladores crear ramas independientes del código principal para trabajar en nuevas características o arreglar errores sin afectar la rama principal. Posteriormente, estas ramas pueden fusionarse de nuevo en la rama principal (o en otra rama) cuando estén listas.
- **Control de versiones eficiente:** Git almacena los cambios de manera eficiente, utilizando técnicas como la compresión y el almacenamiento de diferencias, lo que hace que el repositorio de Git no ocupe demasiado espacio en disco.

GIT

- **Integridad de datos:** Cada archivo y cambio en Git está verificado por un hash criptográfico, lo que garantiza la integridad de los datos y facilita la detección de errores o corrupción.
- **Compatibilidad con múltiples plataformas:** Git es compatible con diversas plataformas, incluyendo Linux, macOS y Windows, lo que facilita su adopción en una amplia gama de entornos de desarrollo.

GitHub

GitHub es una plataforma en línea que utiliza el sistema de control de versiones Git para el alojamiento y la colaboración en proyectos de desarrollo de software. Fundada en 2008, GitHub ha ganado una enorme popularidad y se ha convertido en un centro importante para la comunidad de desarrollo de software.
El 4 de junio de 2018 Microsoft compró GitHub por la cantidad de 7500 millones de dólares. Al inicio, el cambio de propietario generó preocupaciones y la salida de algunos proyectos de este sitio; sin embargo, no fueron representativos. GitHub continúa siendo la plataforma más importante de colaboración para proyectos de código abierto

GitHub

Las características principales de GitHub incluyen:

- **Repositorios públicos y privados:** GitHub permite a los usuarios crear repositorios tanto públicos como privados para alojar su código fuente. Los repositorios públicos son accesibles para cualquiera, mientras que los privados están restringidos a los colaboradores autorizados.
- **Colaboración:** GitHub facilita la colaboración entre desarrolladores al permitirles trabajar juntos en un mismo proyecto. Los usuarios pueden clonar un repositorio, realizar cambios en su propia copia y luego enviar solicitudes de extracción (pull requests) para proponer cambios al proyecto principal.
- **Gestión de problemas y seguimiento de incidencias:** GitHub proporciona herramientas para el seguimiento de problemas y la gestión de incidencias. Los usuarios pueden informar de problemas, sugerir nuevas características o realizar preguntas a través de un sistema de tickets integrado.

GitHub

- **Control de acceso:** GitHub permite a los propietarios de repositorios controlar quién puede acceder y contribuir a sus proyectos mediante el uso de permisos y roles.
- **Integraciones y automatización:** GitHub es compatible con una amplia gama de herramientas y servicios de integración continua (CI), lo que permite automatizar procesos como pruebas de código, compilación y despliegue.
- **Comunidad y networking:** GitHub es un lugar donde los desarrolladores pueden mostrar su trabajo, colaborar con otros en proyectos de código abierto y aprender de la comunidad global de desarrolladores.

Práctica

Modificar el siguiente script para que:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>El primer script</title>

<script type="text/javascript">
  console.log("Hola Mundo!");
</script>
</head>

<body>
<p>Esta página contiene el primer script</p>
</body>
</html>
```

Práctica

1. Todo el código JavaScript se encuentre en un archivo externo llamado `codigo.js` y el script siga funcionando de la misma manera.
2. Después del primer mensaje, se debe mostrar otro mensaje que diga "Soy el primer script"
3. Añadir algunos comentarios que expliquen el funcionamiento del código

Práctica 2

Completar las condiciones de los if del siguiente script para que los mensajes se muestren siempre de forma correcta:

```
var numero1 = 5;
var numero2 = 8;

if(...) {
  console.log("numero1 no es mayor que numero2");
}
if(...) {
  console.log("numero2 es positivo");
}
if(...) {
  console.log("numero1 es negativo o distinto de cero");
}
if(...) {
  console.log("Incrementar en 1 unidad el valor de numero1 no lo hace mayor o igual que numero2");
}
```

Práctica extras JS

- Escribir el código de una función a la que se pasa como parámetro un número entero y devuelve como resultado una cadena de texto que indica si el número es par o impar. Mostrar por pantalla el resultado devuelto por la función.

- Definir una función que muestre información sobre una cadena de texto que se le pasa como argumento. A partir de la cadena que se le pasa, la función determina si esa cadena está formada sólo por mayúsculas, sólo por minúsculas o por una mezcla de ambas.

Ejercicios resueltos TS

En el siguiente enlace encontrarás ejercicios resueltos en TypeScript:

https://github.com/nestorkx/Typescript/


Hedy