

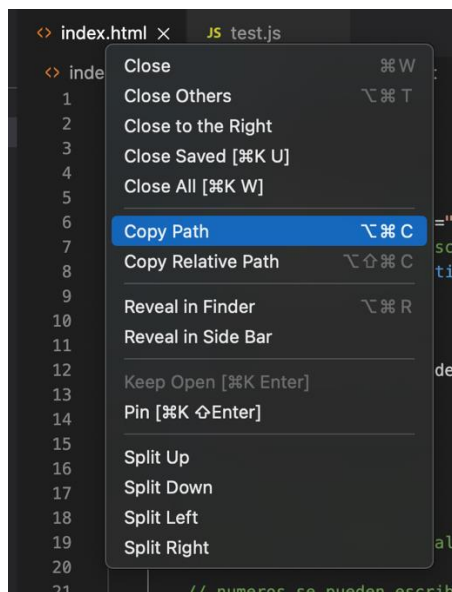
HINTS

CÓDIGO EN VIVO

Si quieres ver los resultados de tu código JS en vivo, es recomendable descargar la extensión Live Server, para VS code.

CÓMO DESPLEGAR UN HTML EN EL NAVEGADOR

Para visualizar un HTML en nuestro navegador, necesitamos de la ruta completa del proyecto, por lo que si buscamos entre todas nuestras carpetas, esto puede resultar un poco tedioso. Un atajo para poder obtenerla desde Visual Studio Code, es dirigirnos al nombre de nuestro proyecto, y hacer **CLIC** derecho > *Copy Path*.



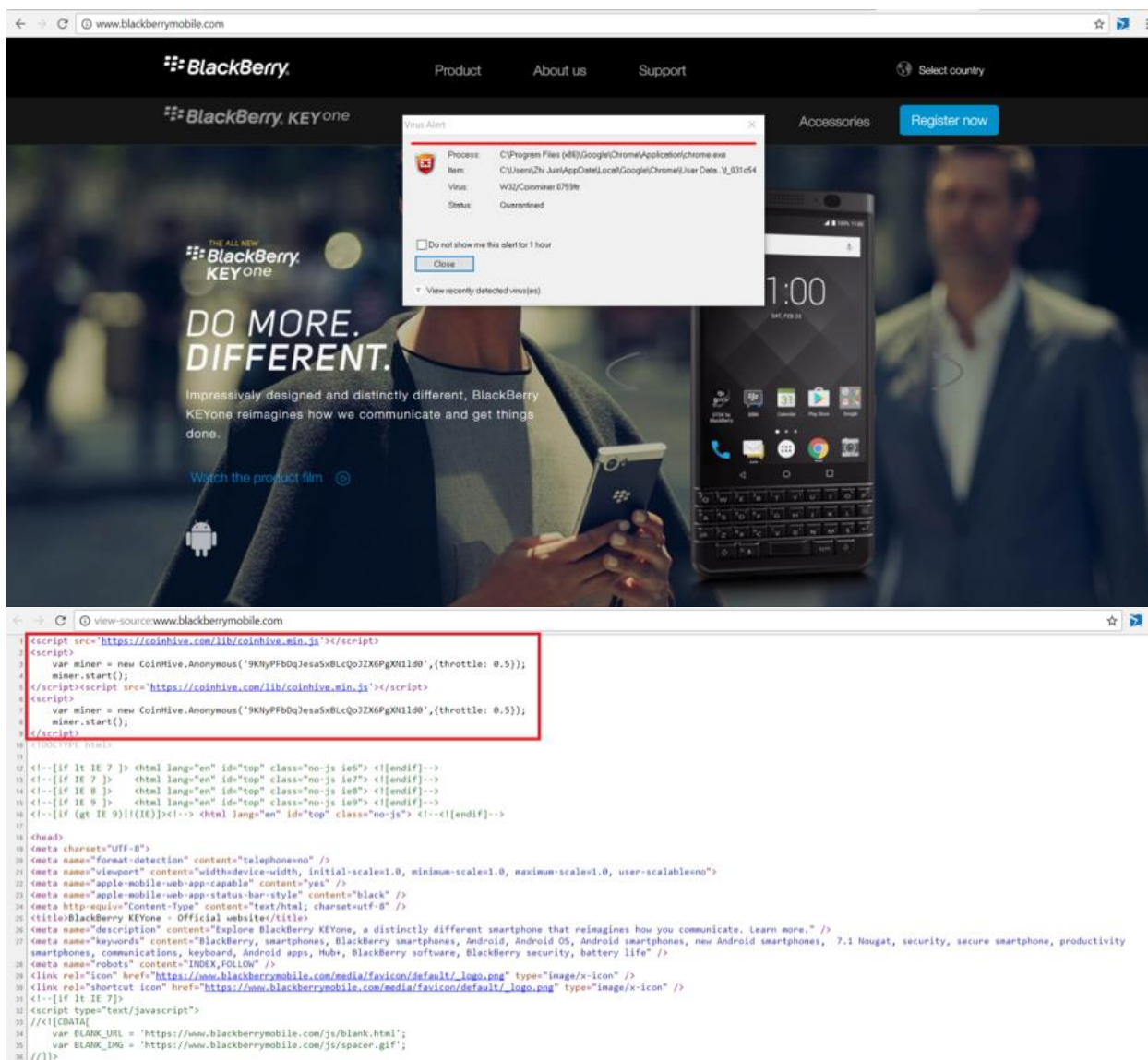
LIMITACIONES DE JAVASCRIPT EN EL NAVEGADOR

Aunque JavaScript es un lenguaje de programación muy poderoso, no deja de tener sus puntos débiles. En este caso, mencionaremos algunas de sus limitaciones.

1. Seguridad del lado del cliente

Dado que el usuario de un sitio web puede ver el código JavaScript, otros pueden usarlo con fines maliciosos; por ejemplo: utilizar el código fuente sin autenticación. Además, es muy fácil colocar algún

código en el sitio que comprometía la seguridad de los datos; así fue como algunas personas se infectaron con código maligno, que utilizaba dispositivos de víctimas para extraer criptomonedas.



Como se observa en estas imágenes, una vez el sitio oficial de BlackBerry fue infectado con código JavaScript maligno, se puede apreciar en el cuadro rojo del código de fuente. Las imágenes, y más información, provienen del siguiente [sitio](#).

2. Representación del código detenido

Un solo error de código puede detener la representación de todo el JavaScript en un sitio web, causando que una página no logre verse como debería. Para el usuario, parecerá que JavaScript no estuviera presente.

ARREGLOS EN JAVASCRIPT

A lo largo de este CUE, hemos aprendido acerca de una variedad de datos disponibles para usar en JavaScript, pero hay otro tipo que no se ha mencionado aún: los arreglos.

Sobre este tipo de datos abarcaremos más información en un CUE siguiente de este curso, pero es importante tener una idea general de lo que son. Los arreglos son variables que pueden contener una serie de valores en forma de una lista. En JavaScript, las matrices se declaran como cualquier otra variable, y son iguales a los valores colocados entre dos corchetes. Por ejemplo:

```
1 var tres = "hola";  
2  
3 //Los arreglos pueden contener distintos tipos de datos.  
4 var arrayEjemplo = [1, "dos", tres, 4];
```

También llamados matrices, o "Arrays" en inglés, son muy útiles cuando necesitamos guardar y agrupar diferentes piezas de información.

CÓDIGO LIMPIO

Es fácil de entender y cambiar. Si tu código funciona bien, puede que te preguntes por qué ocuparse de limpiarlo; la respuesta es que, incluso si tu código es funcional en un momento, debes tener en cuenta el que estará en el futuro. Si el código no está limpio para empezar, será mucho más difícil mantenerlo, actualizarlo o ampliarlo.

La idea de escribir código limpio es subjetiva. Existen algunas mejores prácticas, pero es difícil encontrar una guía en la que todos los desarrolladores puedan estar de acuerdo, de manera consistente. Por este motivo, se recomiendan las siguientes:

1. Usa nombres claros de variables y funciones.
2. Escribe funciones cortas que solo hagan una cosa.

3. Escribe una buena documentación que explique lo que hace tu código.
4. Modulariza tu código, dividiendo funciones grandes en otras más pequeñas.
5. Usa el principio DRY: "No te repitas".

CONVENCIONES

Son pautas de estilo para la programación, que están destinadas a mantener el código limpio, y fácilmente legible para ti y otros desarrolladores que trabajen en éste. Generalmente incluyen:

- Reglas de nomenclatura, y declaración de variables y funciones.
- Reglas para el uso de espacios en blanco, sangría y comentarios.
- Prácticas y principios de programación.

Las convenciones de codificación aseguran la calidad:

- Mejora la legibilidad del código.
- Facilitar el mantenimiento del código.

Éstas pueden ser reglas documentadas, diseñadas para ser seguidas por los equipos, o simplemente servir como práctica de codificación individual. Independientemente, varían de una organización a otra, por lo que es importante consultar las convenciones de codificación de JavaScript de [Mozilla](#), [Google](#) y [w3school](#), para tener una mejor idea.

POR QUÉ SEGUIR UNA GUÍA Y CONVENCIONES

Nos puede facilitar mucho nuestro trabajo desarrollando, e incluso nos puede permitir mejorar nuestra forma de codificar. Por ejemplo, veamos los beneficios de aplicar la convención de mantener el código limpio; uno de los principales es que nos ayudará a minimizar el tiempo que necesitamos para leerlo y tratar de entenderlo.

Un código desordenado puede ralentizar a cualquier desarrollador, y hacer que su trabajo sea mucho más difícil. Cuanto más desordenado está, más tiempo se necesita para comprenderlo; y si el código es demasiado complicado, el desarrollador puede decidir detenerse y comenzar desde cero.

Este es el primer beneficio de un código en armonía con los consejos que revisaremos; y aplica tanto para nuestros propios proyectos, como para el trabajo de otros desarrolladores.