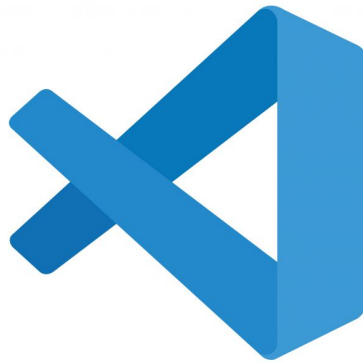


A1 Hola Mundo



Índice

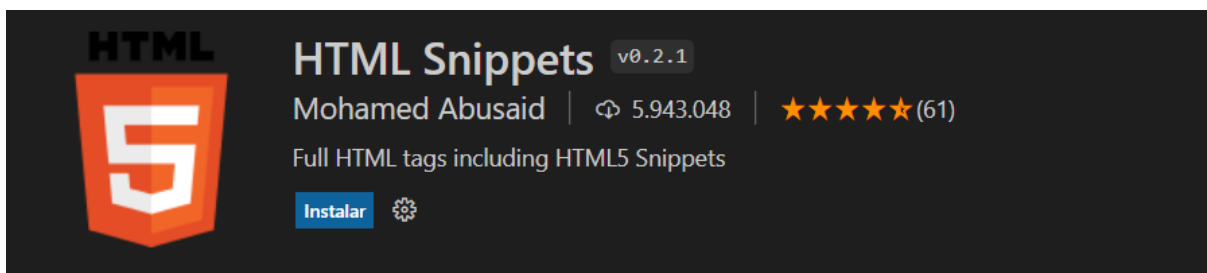
Editores de texto	3
Visual Studio Code	3
Atom	4
Sublime Text	7
Navegadores	8
Google Chrome	8
Mozilla	11
Opera GX	14

Editores de texto

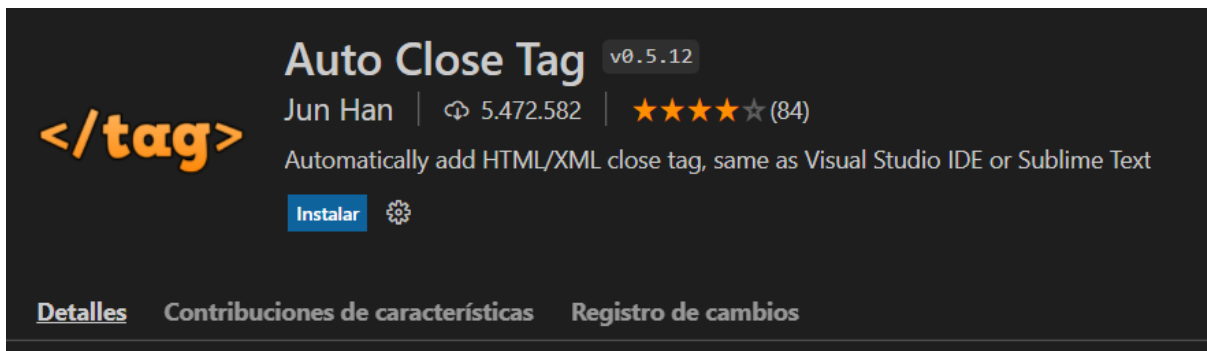
Visual Studio Code

1. Instalaciones necesarias para reconocimiento de código html y javascript

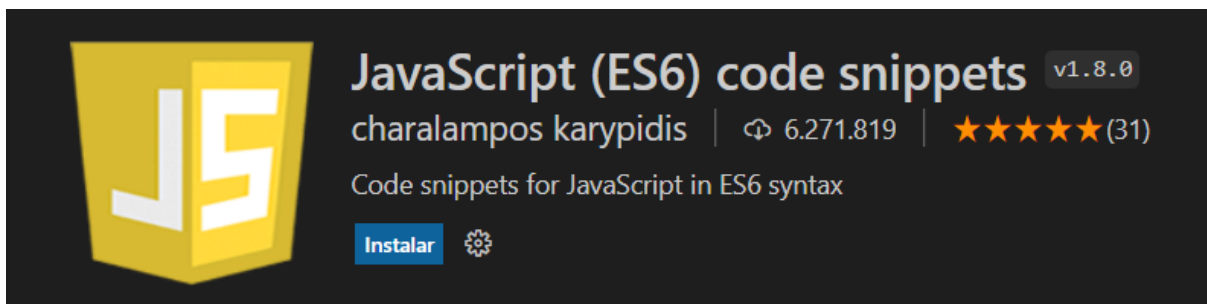
En sí no es necesario descargarse un plugin en vsc para el reconocimiento de estos lenguajes, ya que de forma nativa los reconoce, pero lo que sí podemos hacer es descargar plugins que nos ayuden a escribir el código, como estos:



Este plugin además de incorporar todas las etiquetas html, trae atajos html para codificar más rápido, y colores para diferenciar mejor las etiquetas.



Plugin que cierra de forma automática la etiqueta html.



Este plugin te ofrece atajos en JavaScript.



Por último este plugin colorea las sentencias de js y de html para que sean más visuales.

2. Atajos en ambos lenguajes. (mínimo de 5)

Alt + flechas arriba y abajo: te posicionas en una línea de código, haces este comando y podrás subir y bajar la línea de código en cuestión.

html5 + enter: este atajo se escribe directamente en el archivo html, cuando lo ejecutas te imprime un html5 básico.

Ctrl + h: este atajo html te permite sustituir todas las etiquetas en un archivo html.

imp + tab: escribe una línea de código que te permite importar una clase.

pge + tab: escribe un método get en JavaScript.

Atom

1. Instalaciones necesarias para reconocimiento de código html y javascript

Al igual que en visual studio code no es necesario un plugin para interpretar js, pero si hay varias opciones para codificar de manera más sencilla, como por ejemplo:

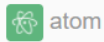


Este plugin es muy útil, la labor de este es señalar los errores sintácticos en el código, como los ; .

teletype

Share your workspace with team members and collaborate on code in real time

#collaboration #collaborative-editing #pair-programming #real-time



1,390,492

★ 715

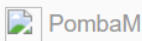
Install

Este plugin no tiene que ver nada con JavaScript ni con html, pero es muy útil para el desarrollo de ambos, ya que puedes compartir tu proyecto con tus compañeros y codificar en tiempo real.

quick-javascript

Write JavaScript quickly with over 150+ handy snippets. Fully ES6! Less typing more coding!

#javascript #snippets #es6 #productivity



2.2.0 67,898

★ 19

Install

Este plugin añade más de 150 atajos para JavaScript.

html-to-css

Generate CSS classes from HTML structure

#html #css #bem #productivity



116,220

★ 73

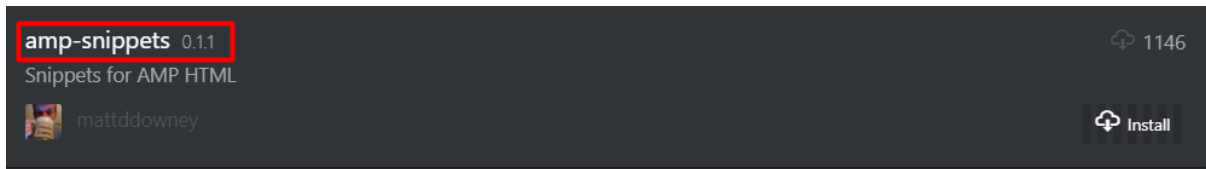
Install

Este plugin es capaz de generar una clase CSS desde la estructura de un archivo html, algo parecido a este ejemplo:

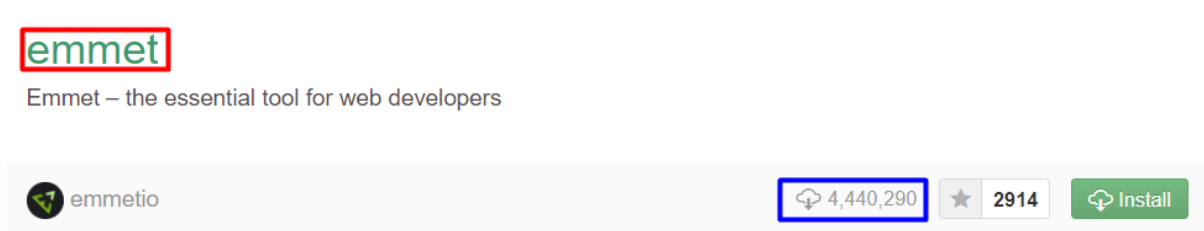
```
<section class="introduction">
  <h1 class="introduction__header"></h1>
  <p class="introduction__text"></p>
</section>
```

↓

```
.introduction {
  &__header {
  }
  &__text {
  }
}
```



Con este plugin incorporamos atajos html a atom.



Por último, podemos descargar la herramienta emmet en Atom.

2. Atajos en ambos lenguajes. (mínimo de 5)

Descargando el plugin de arriba -> **quick-javascript** tenemos:

- Ctrl + B** : Crea un bloque {...}.
- if + tab**: crea un bloque de if.
- el + tab**: crea un bloque else.
- wl + tab**: crea un bloque while.
- ty + tab**: crea un bloque try/catch.

Descargando el paquete de arriba -> **amp-snippets** podemos tener estos atajos:

- html + tab**: Imprime una estructura html básica.
- iframe + tab**: completa la sentencia html iframe.
- img + tab**: completa la sentencia html img.
- link + tab**: completa la sentencia html link.

Sublime Text

1. Instalaciones necesarias para reconocimiento de código html y javascript

Tabnine: Este plugin indexa el código html y Js, también puede predecir la escritura del código que estás escribiendo.

Emmet: Con sus atajos puedes escribir largas estructuras html de una sola vez.

JsHint: Cuando se instala y se ejecuta este plugin abre una ventana en sublime dónde puedes ver los errores que hay en tu código.

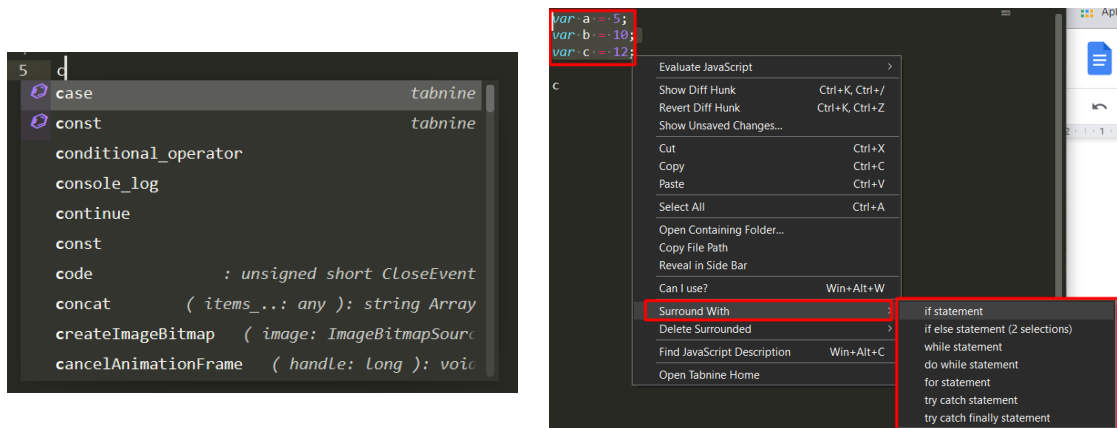
AutoFileName: Muestra una lista de archivos y directorios que puedes explorar y seleccionar.

Terminal: Puede ejecutar el terminal en la posición del archivo actualmente editado o saltar a la carpeta raíz del proyecto.

JavaScript-Completions: Con este plugin podemos tener a mano una muy buena cantidad de atajos.

2. Atajos en ambos lenguajes. (mínimo de 5)

En JavaScript con el plugin **JavaScript-Completions** tenemos una gran cantidad de “atajos”, a la hora de escribir código el plugin nos sugiere muchas opciones diferentes:



Y desde HTML tenemos el plugin emmet y todas sus opciones posibles.

Navegadores

Google Chrome

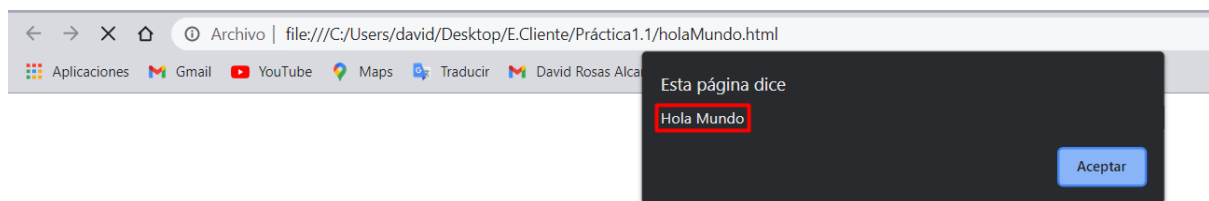
1. Teclas de acceso rápido que muestren las herramientas de desarrollo.

Ctrl + shift + i & F12 & Click derecho + inspeccionar.

2. Acceso a la consola

En primer lugar debes acceder a las herramientas para desarrolladores y después clicar en el apartado “consola”.

3. Acceso a los ficheros (mensaje.js en nuestro caso)

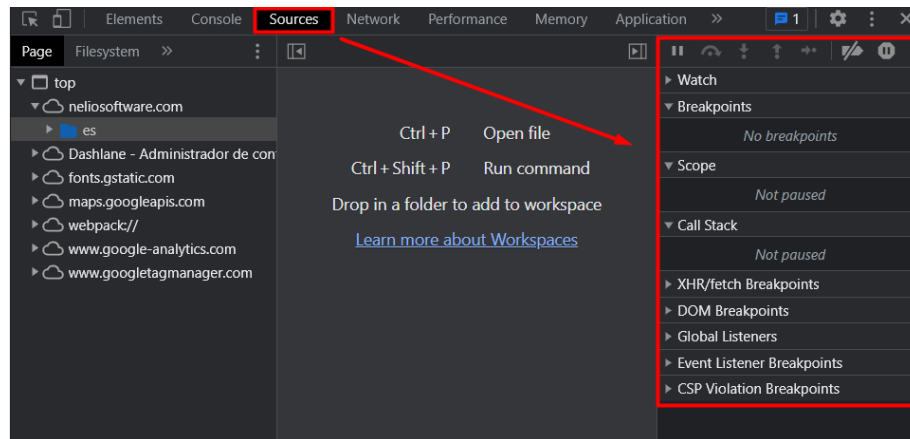


4. Reglas CSS

Debemos de darle “click derecho” y en el menú que nos sale tenemos que darle a “inspeccionar”, de esta manera nos llevará directamente al apartado de elementos en las herramientas de desarrolladores dónde podemos observar el css de la página web.

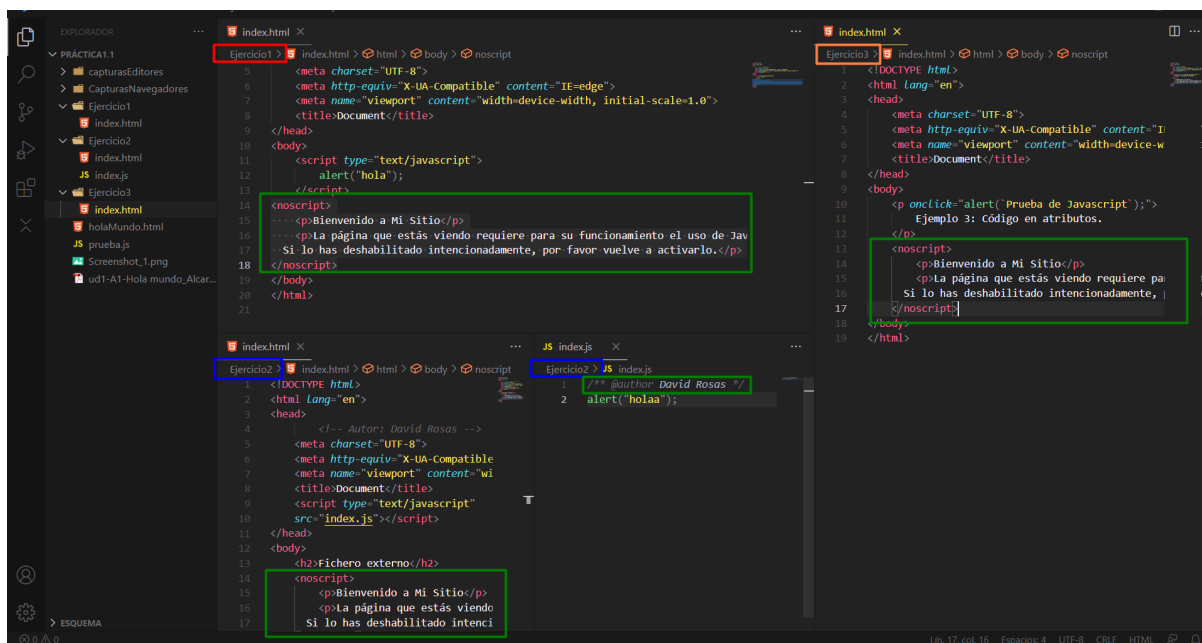
5. Depurador

De nuevo desde las herramientas de desarrollador, accedemos al apartado source:

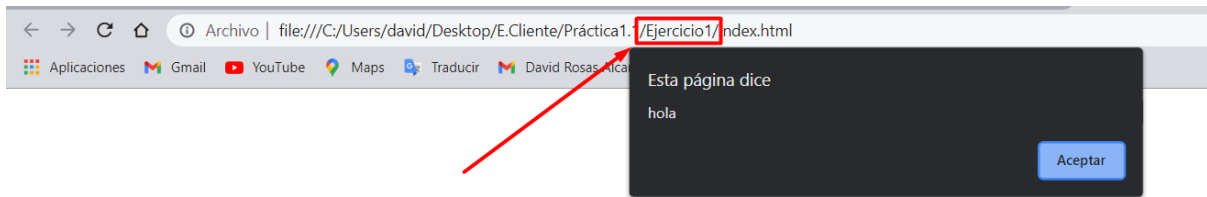


6. Implementa los ejemplos propuestos en el apartado 1.4 del tema y prueba con los distintos navegadores web (Chrome, Edge, Firefox, Opera...).

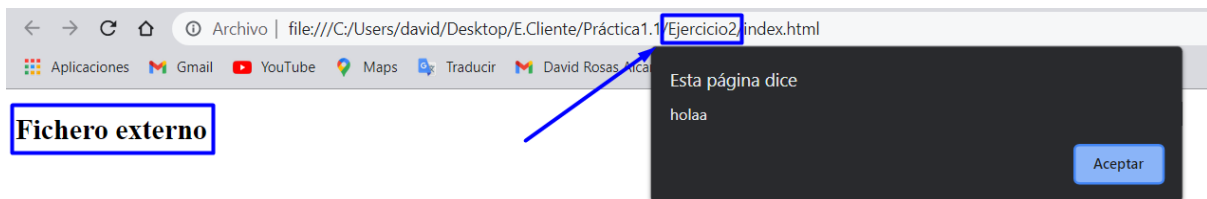
Los códigos son los siguientes:



Ejercicio 1:

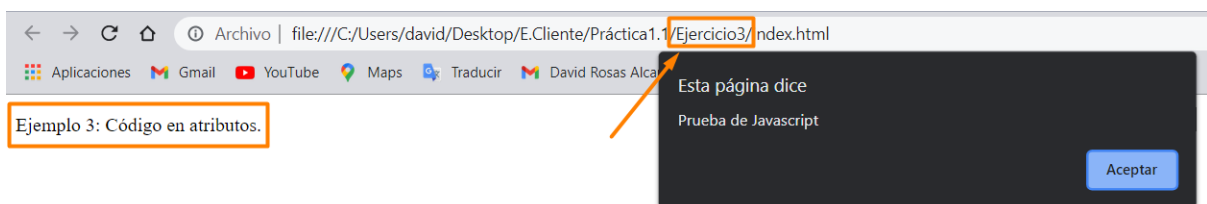


Ejercicio 2:



Fichero externo

Ejercicio 3:



Ejemplo 3: Código en atributos.

Mozilla

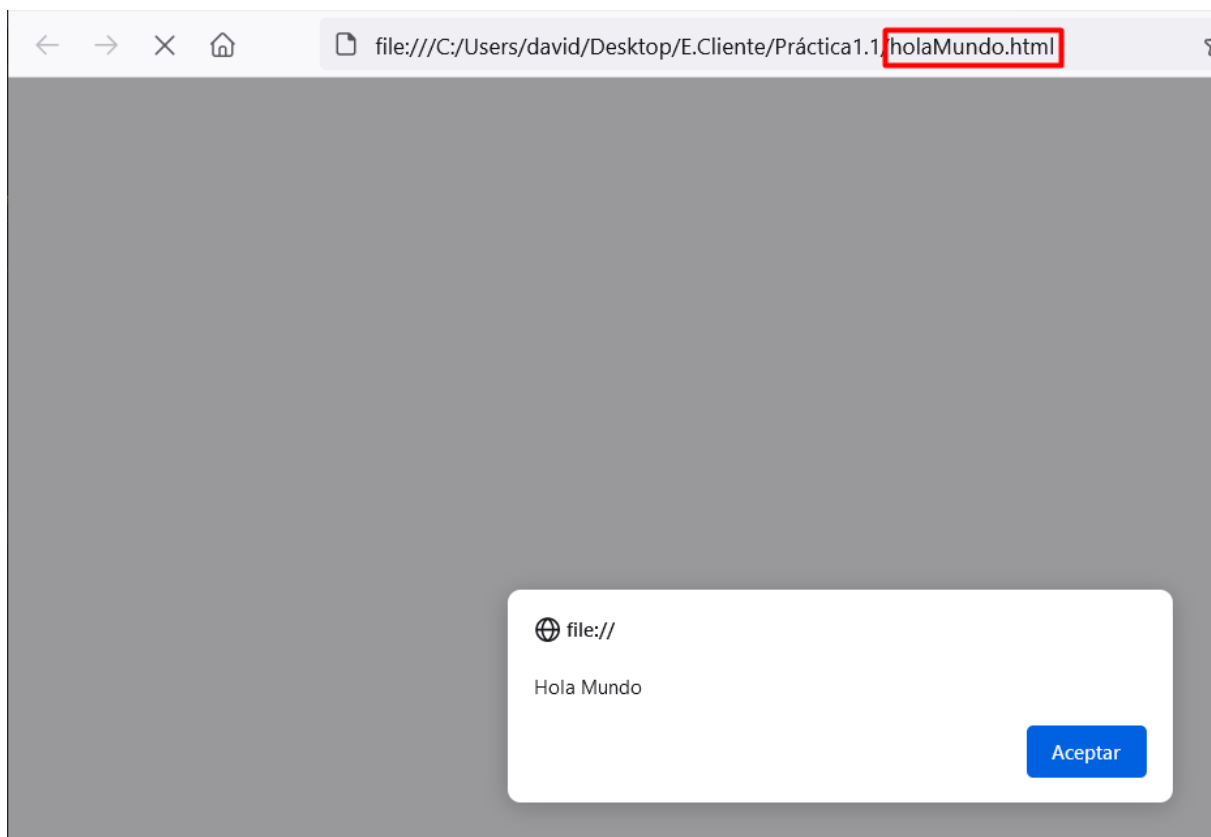
1. Teclas de acceso rápido que muestren las herramientas de desarrollo.

Ctrl + shift + i & F12 & Click derecho + inspeccionar.

2. Acceso a la consola

En primer lugar debes acceder a las herramientas para desarrolladores y después clicar en el apartado "consola".

3. Acceso a los ficheros (mensaje.js en nuestro caso).



4. Reglas CSS

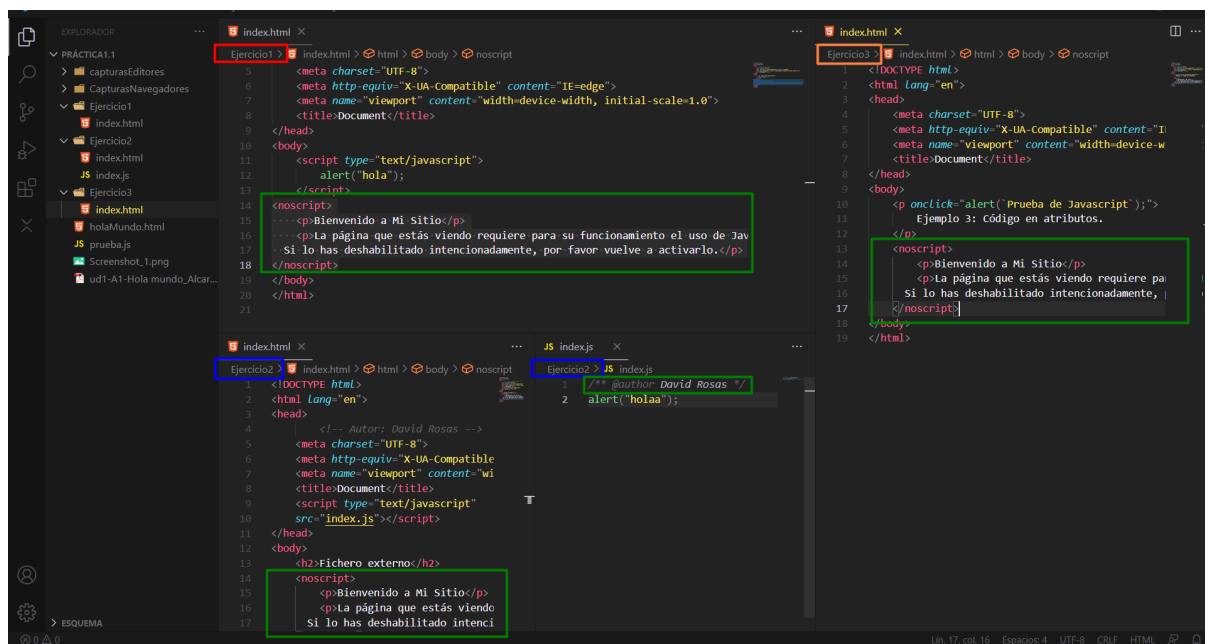
La forma de acceder a las reglas CSS es la misma que en google. Accedemos a las herramientas del desarrollador mediante los atajos visto anteriormente y nos vamos al apartado de "Inspector".

5. Depurador

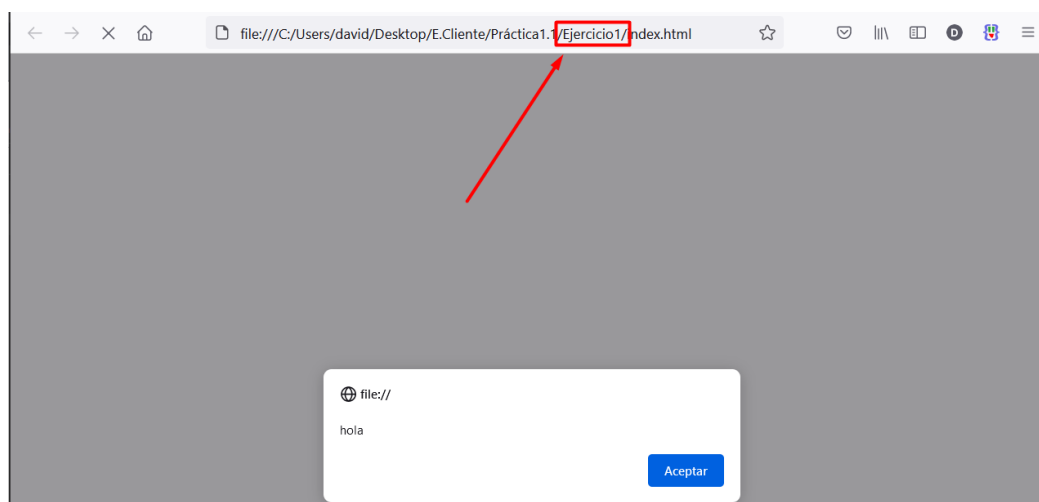
En Mozilla debemos acceder a las herramientas del desarrollador y dirigirnos directamente al apartado: “Depurador”.

- Implementa los ejemplos propuestos en el apartado 1.4 del tema y prueba con los distintos navegadores web (Chrome, Edge, Firefox, Opera...).

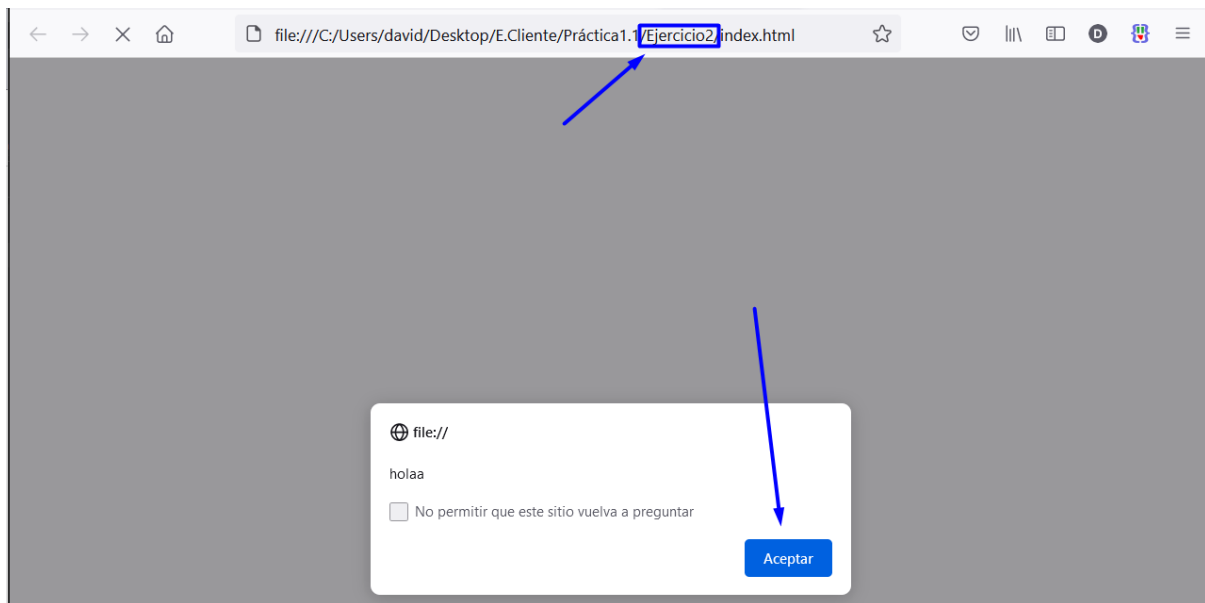
El código js y html es el mismo:



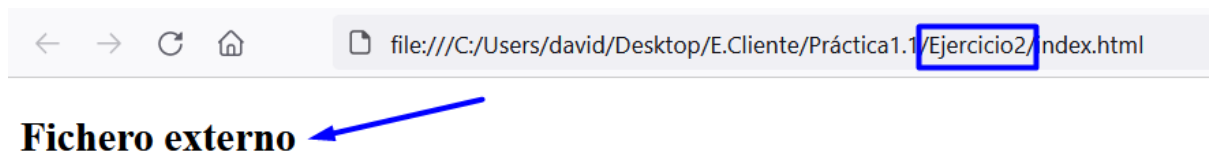
Ejercicio 1:



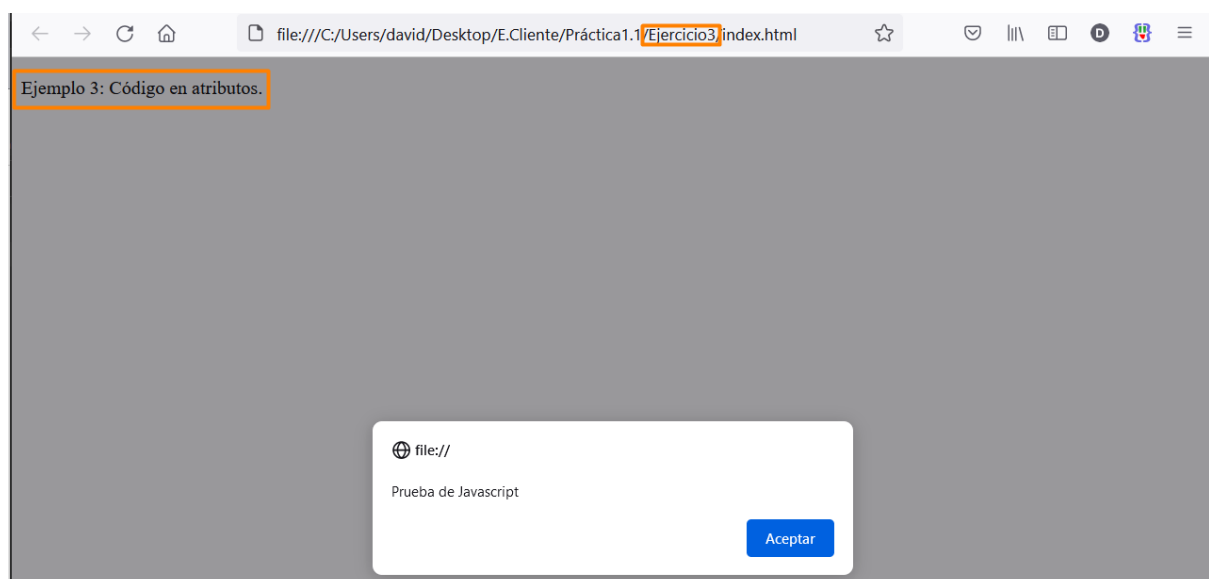
Ejercicio 2:



Cuando le damos a aceptar sale el texto “Fichero externo”.



Ejercicio 3:



Opera GX

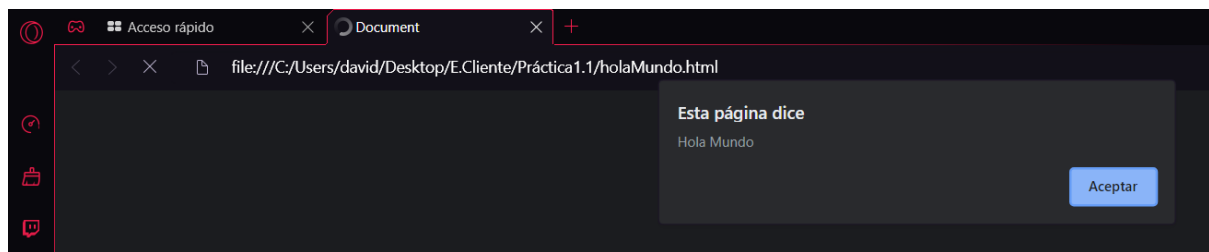
1. Teclas de acceso rápido que muestren las herramientas de desarrollo.

En este navegador solo podemos acceder a las herramientas de desarrollo con el Ctrl + Shift + i.

2. Acceso a la consola

Al acceder a las herramientas de desarrollo debemos dirigirnos al apartado de consola.

3. Acceso a los ficheros (mensaje.js en nuestro caso).

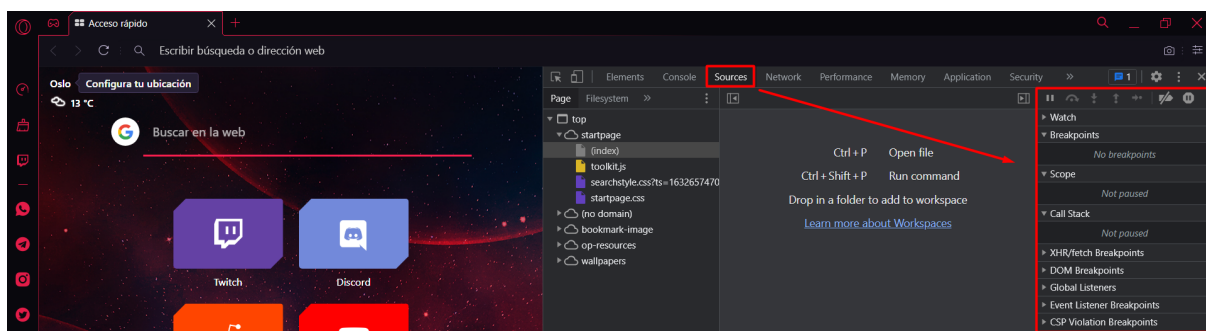


4. Reglas CSS

Accedemos a las herramientas y nos dirigimos al apartado de "Elements"

5. Depurador

Accedemos a las herramientas y nos dirigimos al apartado de "Source".



6. Implementa los ejemplos propuestos en el apartado 1.4 del tema y prueba con los distintos navegadores web (Chrome, Edge, Firefox, Opera...).

El código sigue siendo el mismo:

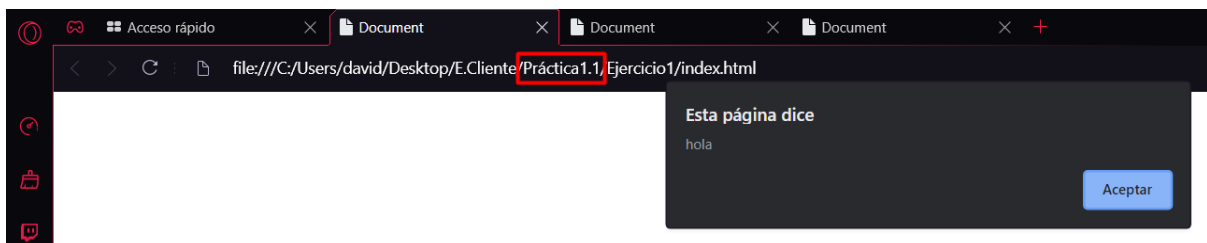
```

Ejercicio1:
<script type="text/javascript">
  alert("hola");
</script>

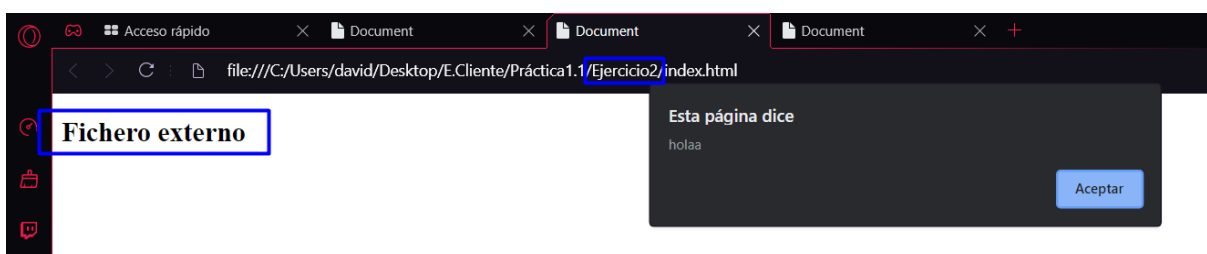
Ejercicio2:
<script type="text/javascript" src="index.js"></script>
<noscript>
  <p>Bienvenido a Mi Sitio</p>
  <p>La página que estás viendo requiere para su funcionamiento el uso de Jav
  <p>Si lo has deshabilitado intencionadamente, por favor vuelve a activarlo.</p>
</noscript>

Ejercicio3:
<p onclick="alert('Prueba de Javascript');">
  Ejemplo 3: Código en atributos.
</p>
<noscript>
  <p>Bienvenido a Mi Sitio</p>
  <p>La página que estás viendo requiere pa
  <p>Si lo has deshabilitado intencionadamente,
</noscript>
  
```

Ejercicio 1:



Ejercicio 2:



Ejercicio 3:

