



# INSTALACIÓN Y CONFIGURACIÓN DE UN ENTORNO DE TRABAJO PARA PROGRAMAR CON JAVASCRIPT

UT1-T01



AYMEDIACOÁN MAULEÓN QUINTANA  
19/9/2017  
DESARROLLO WEB EN ENTORNO CLIENTE

## Índice

1. Enunciado .....	3
2. Objetivo .....	4
3. Entornos de trabajo .....	5
3.1. Atom.....	5
3.1.1. Introducción .....	5
3.1.2. Tratamiento del código .....	5
3.1.2.1. Chequeo de sintaxis y autocompletado .....	5
3.1.2.2. Formateo del código .....	5
3.1.2.3. Selección o agrupación en bloque .....	6
3.1.3. Atajos de teclado .....	6
3.1.4. Paquetes .....	6
3.1.5. Compatibilidad .....	6
3.2. Sublime Text.....	7
3.2.1. Introducción .....	7
3.2.2. Tratamiento del código .....	7
3.2.2.1. Chequeo de sintaxis .....	7
3.2.2.2. Formateo del código .....	7
3.2.2.3. Selección o agrupación en bloque .....	8
3.2.3. Autocompletado .....	8
3.2.4. Atajos de teclado .....	8
3.2.5. Paquetes .....	8
3.2.6. Compatibilidad .....	8
3.3. CodePen .....	8
3.3.1. Introducción .....	8
3.3.2. Tratamiento del código .....	8
3.3.2.1. Chequeo de sintaxis .....	8
3.3.2.2. Formateo del código .....	9
3.3.2.3. Selección o agrupación en bloque .....	9
3.3.3. Autocompletado .....	9
3.3.4. Atajos de teclado .....	9
3.3.5. Paquetes .....	10
3.3.6. Compatibilidad .....	10
3.4. Conclusión final.....	10

4.	Validación de W3C.....	10
4.1.1.	Introducción .....	10
4.1.2.	Tipos de errores.....	10
4.1.3.	Soluciones.....	11

## 1. Enunciado

Realizar la instalación y configuración de un entorno de trabajo para programar con JavaScript.

- Para ello tendrás que buscar e instalar en tu sistema operativo 3 editores de páginas web, de los cuales, uno de ellos debe ser un editor online. Todas las aplicaciones que instales tendrán que ser gratuitas y tendrás que explicar las razones por las que has elegido tu editor web entre muchos otros.
  - o multilenguaje
  - o chequeo de la sintaxis del lenguaje
  - o formateo del código (espaciado o indentación, etc.)
  - o selección o agrupación de bloques de código (funciones, etc.)
  - o emparejamiento de llaves
  - o autocompletado
  - o atajos de teclado
  - o paquetes que pueden instalarse y cuáles recomendarías
  - o SSOO en los que puede instalarse o usarse. En el caso del editor web navegadores soportados
- Usando la dirección de validación del W3C lleva a cabo la validación de la página de Google e indica los tipos de errores encontrados, cita 3 errores diferentes detectados y la solución propuesta a cada uno de ellos.

## 2. Objetivo

El objetivo de la práctica es investigar, así como, familiarizarse con los entornos de desarrollo para programar con JavaScript de que se dispone, incluyendo tanto editores de escritorio como online, ver el tratamiento del código, las herramientas y mejoras disponibles en cada uno de ellos, eligiendo 3 y determinando cuál es el que más se ajusta a nuestras necesidades. Además de ver otras herramientas para la comprobación de errores.

## 3. Entornos de trabajo

### 3.1. Atom

#### 3.1.1. Introducción

Atom es un editor de código open source desarrollado por GitHub, del cual su mejor característica es su alta personalización.

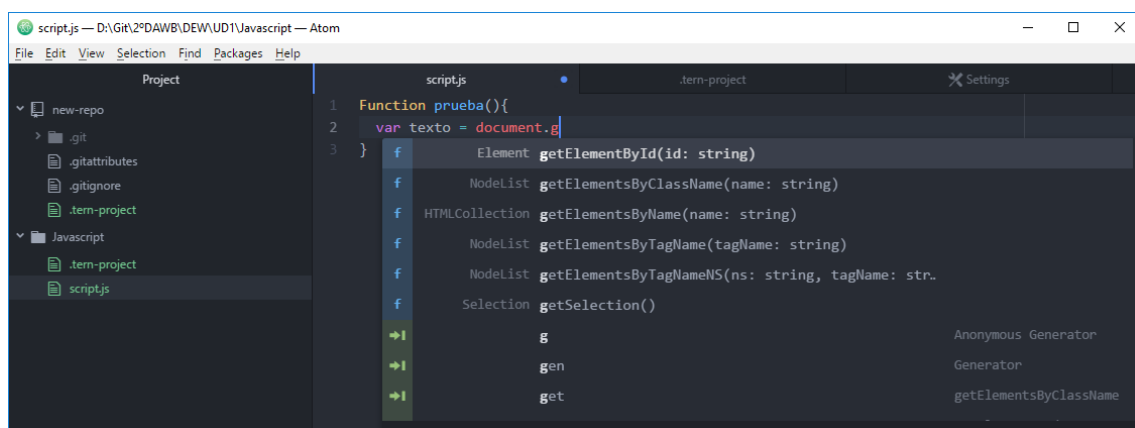
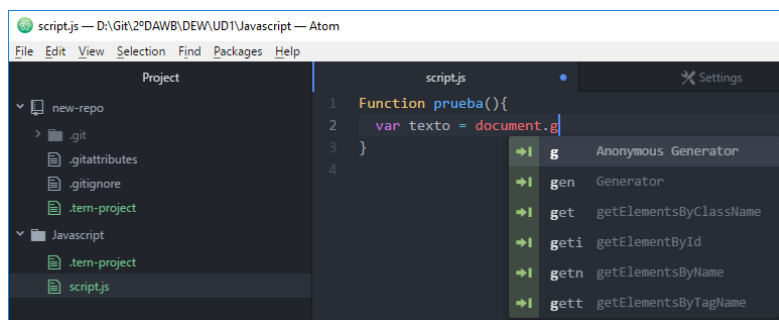
Incorpora la utilización de plugins para facilitar el desarrollo del código. Entre sus funcionalidades se encuentra la capacidad de trabajar con varios lenguajes tales como JavaScript, C/C++, C#, Java, CSS, Python entre otros muchos. Incluye la utilización de librerías o CDNs.

#### 3.1.2. Tratamiento del código

##### 3.1.2.1. Chequeo de sintaxis y autocompletado

Instalando el paquete atom-ternjs tendremos una utilidad de autocompletado de código de JavaScript y chequeo de sintaxis. Como indica su documentación es necesario añadirle un fichero de configuración donde indiquemos como queremos que nos ayude.

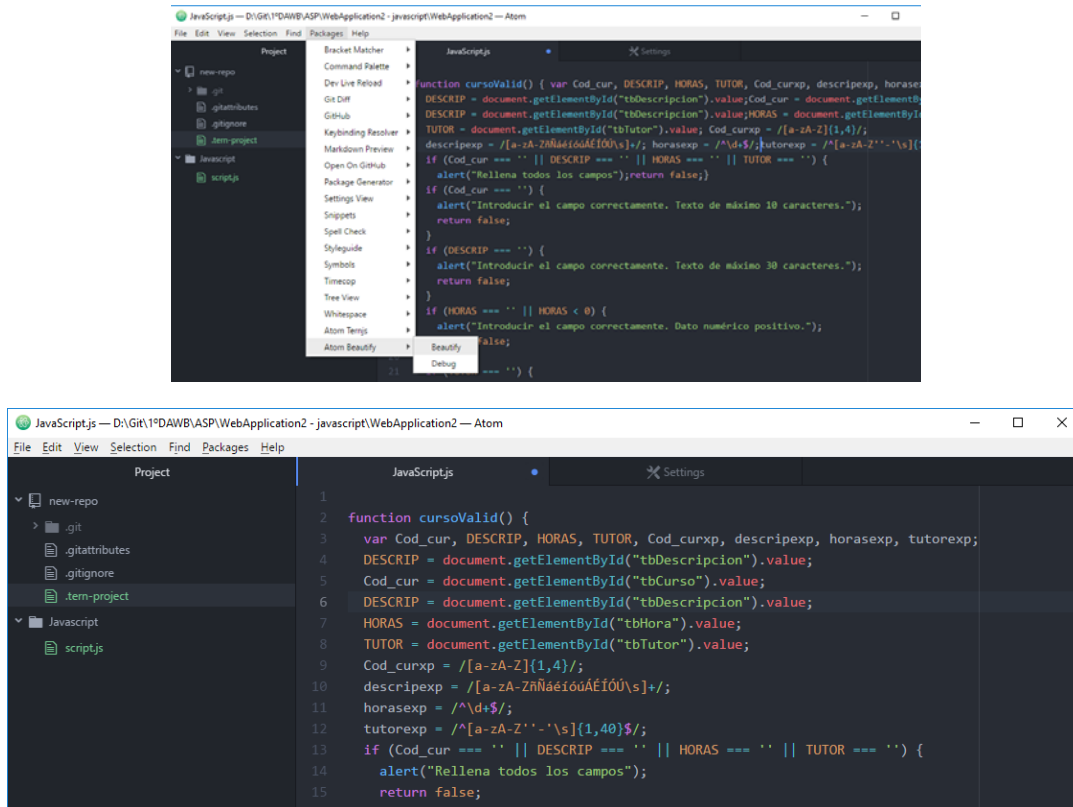
En este ejemplo podemos observar la diferencia entre las sugerencias mostradas con el paquete ternjs desactivado, primera captura, y activado, segunda captura, en ésta vemos como nos ofrece una mayor cantidad de sugerencias, que son aquellas marcadas con el icono azul.



##### 3.1.2.2. Formateo del código

Con el paquete atom-beautify podemos organizar e indentar un código mal tabulado, es decir, embellecer o dejar bonito.

En la primera captura vemos que el código es confuso ya que se encuentra muy junto y no es legible. Pulsando el paquete Beautify obtenemos como resultado la segunda captura, en la que vemos que el código es mucho más comprensible a primera vista.



### 3.1.2.3. Selección o agrupación en bloque

Para realizar estas dos acciones no es necesario instalar ningún complemento, ya que, viene implementado por defecto en el editor. Se trata de contraer u ocultar un bloque completo al pulsar la flecha que se encuentra al inicio de la instrucción padre, una vez contraído se puede seleccionar todo el bloque.

### 3.1.3. Atajos de teclado

Para consultar el listado completo de atajos del teclado disponibles en atom, deberemos presionar “ctrl + ,” y seleccionar Keybindings. Presionando “ctrl+shift+p” aparecerá un menú flotante en el que podremos escribir la función deseada.

### 3.1.4. Paquetes

A parte de los paquetes anteriormente mencionados existen otros como:

- Emmet: Es un plugin que es necesario instalar para poder usar abreviaturas que nos permiten trabajar con rapidez utilizando la tecla Tab para autocompletar la sintaxis de HTML, CSS, Sass, Scss y Less.
- JavaScript Snippets: Te permite escribir rápidamente una parte de código fuente de JS. Puedes usar abreviaturas definidas, con lo cual no es necesario escribir palabra por palabra de código. Esta funcionalidad es similar a Emmet, pero se ejecuta en código JavaScript. Este paquete Atom ofrece muchas de la sintaxis de JavaScript.

### 3.1.5. Compatibilidad

Este editor es compatible con los sistemas operativos Windows, Mac OS y Linux.

## 3.2. Sublime Text

### 3.2.1. Introducción

Es un editor de textos que aporta muchas características útiles a la hora de programar o editar código. El editor tiene muchas funcionalidades útiles y cómodas de usabilidad y eficiencia. Entre sus funcionalidades se encuentra la capacidad de trabajar con varios lenguajes tales como JavaScript, C/C++, C#, Java, CSS, Python, PHP, Perl, XML, SQL, etc.

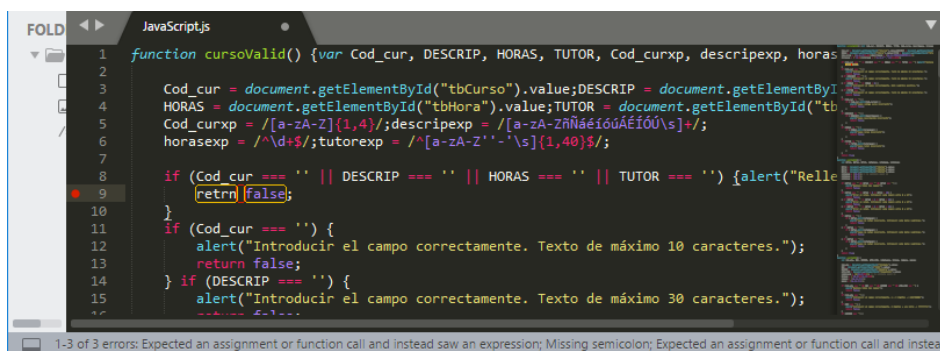
Aunque no se trata de software libre o código abierto, es posible descargarlo de forma de evaluación gratuita. Se debe obtener una licencia para su uso continuado, aunque la versión de evaluación es plenamente funcional y no tiene fecha de caducidad.

### 3.2.2. Tratamiento del código

#### 3.2.2.1. Chequeo de sintaxis

Para poder utilizar el chequeo de sintaxis en el Sublime deberemos instalar los paquetes SublimeLinter y SublimeLinter-jshint, además deberemos tener instalado el node.js y el jshint.

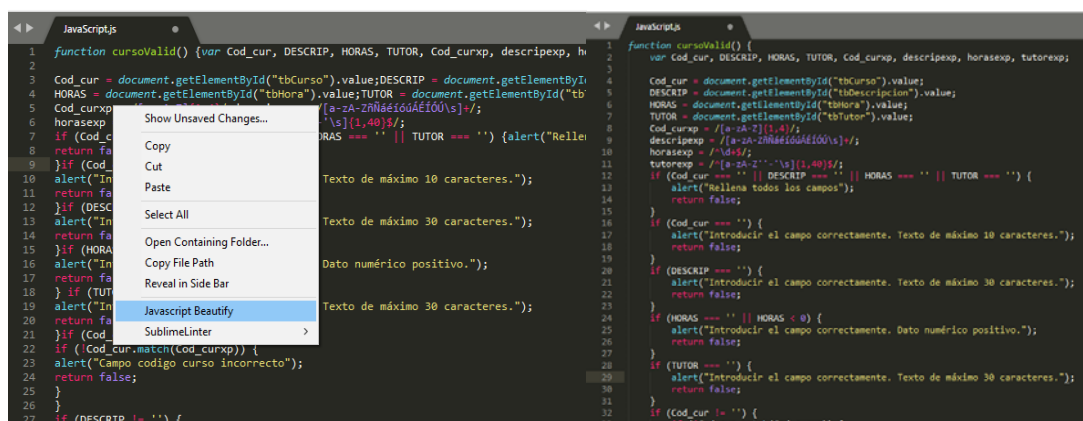
En la siguiente captura vemos el funcionamiento del chequeo, al detectar que falta un “;” nos da un aviso en forma de círculo de color donde está el error y en el pie de la ventana nos muestra el detalle del error.



#### 3.2.2.2. Formateo del código

Con el paquete JavaScript beautify podemos organizar e indentar un código mal tabulado, es decir, embellecer o dejar bonito.

En la primera captura vemos que el código es confuso ya que se encuentra muy junto y no es legible. Pulsando el paquete Beautify obtenemos como resultado la segunda captura, en la que vemos que el código es mucho más comprensible a primera vista.





#### 3.2.2.3. Selección o agrupación en bloque

El editor permite realizar una agrupación en bloque por medio de flechas en el margen izquierdo, sin embargo, no permite la selección en bloque.

#### 3.2.3. Autocompletado

Instalando el paquete ternjs tendremos una utilidad de autocompletado de código de JavaScript.

#### 3.2.4. Atajos de teclado

El editor no trae un menú de atajos de teclado, pero hay una gran documentación en internet. Posee la utilidad de grabar macros para crear nuestros propios atajos de teclado.

#### 3.2.5. Paquetes

A parte de los paquetes anteriormente mencionados existen otros como:

- Emmet: Es un plugin que es necesario instalar para poder usar abreviaturas que nos permiten trabajar con rapidez utilizando la tecla Tab para autocompletar la sintaxis de HTML, CSS, Sass, Scss y Less.
- BracketHighlighter: Resalta la apertura y cierre de las llaves en un bloque de código. Funciona con corchetes, llaves, paréntesis, comillas simples y dobles, etc. Una herramienta que nos ahorrará mucho tiempo buscando etiquetas mal cerradas o sin cerrar.
- jQuery: Nos ayuda coloreando y resaltando las funciones y métodos de jQuery, además de encargarse del autocompletado.

#### 3.2.6. Compatibilidad

Este editor es compatible con los sistemas operativos Windows, Mac OS y Linux.

### 3.3. CodePen

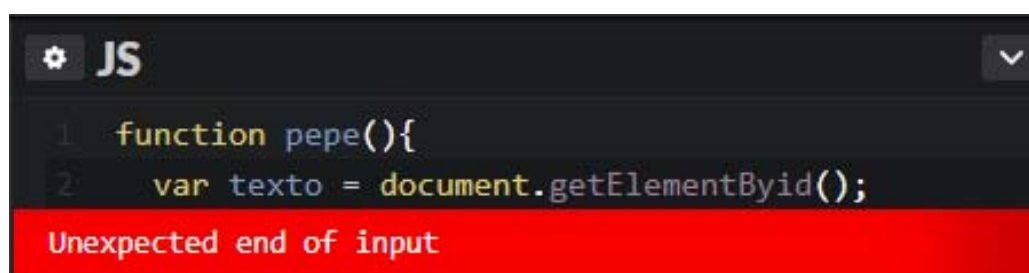
#### 3.3.1. Introducción

CodePen es una de las mejores herramientas de front-end, donde se puede practicar con HTML5, CSS3, JavaScript y otras herramientas relacionadas, es una plataforma de desarrollo donde puedes publicar pequeñas páginas donde hacer ejemplos con fragmentos de código.

#### 3.3.2. Tratamiento del código

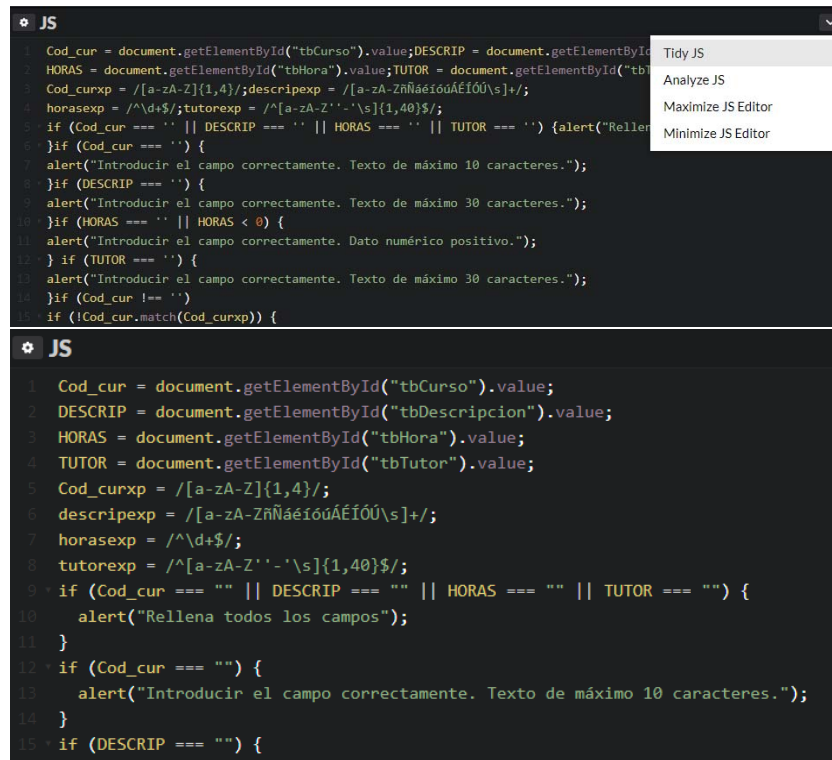
##### 3.3.2.1. Chequeo de sintaxis

Sólo detecta los errores de aperturas y cierres. Como vemos en la captura, detecta que falta la llave del cierre de la función, pero no la "i" minúscula del método.



### 3.3.2.2. Formateo del código

En la flecha desplegable del editor aparece un menú en el cual deberemos elegir Tidy JS, el cual ordenará el código.

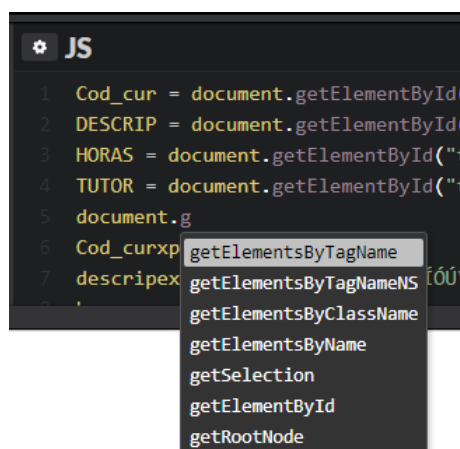


### 3.3.2.3. Selección o agrupación en bloque

El editor permite realizar una agrupación en bloque por medio de flechas en el margen izquierdo, sin embargo, no permite la selección en bloque.

### 3.3.3. Autocompletado

El editor incorpora la utilidad de autocompletado que, pulsando ctrl+espacio autocompleta la instrucción o nos muestra sugerencia de autocompletado, como podemos observar en la captura.



### 3.3.4. Atajos de teclado

El editor tiene un apartado en su configuración para poder usar los atajos de teclado propios, los del editor Sublime Text o del editor Vim. Si optamos por los propios del editor, nos indica en la ayuda dónde podemos consultarlos.

### 3.3.5. Paquetes

No permite la instalación de paquetes, pero en la configuración nos permite crear nuestros propios Snippet HTML y Snippet CSS.

### 3.3.6. Compatibilidad

Codepen puede trabajar en todos los navegadores.

## 3.4. Conclusión final

He decidido optar por estos editores por ser de los más conocidos y utilizados. Viendo las diferencias de los editores de escritorio y el escaso uso que realizado en Atom, creo que de momento prefiero usar el editor de Sublime Text, ya que es bastante más rápido que el Atom.

En caso del editor online, he decidido el Codepen ya que es uno de los más usados, ofrece muchas utilidades y en definitiva es de los más completos.

## 4. Validación de W3C

### 4.1.1. Introducción

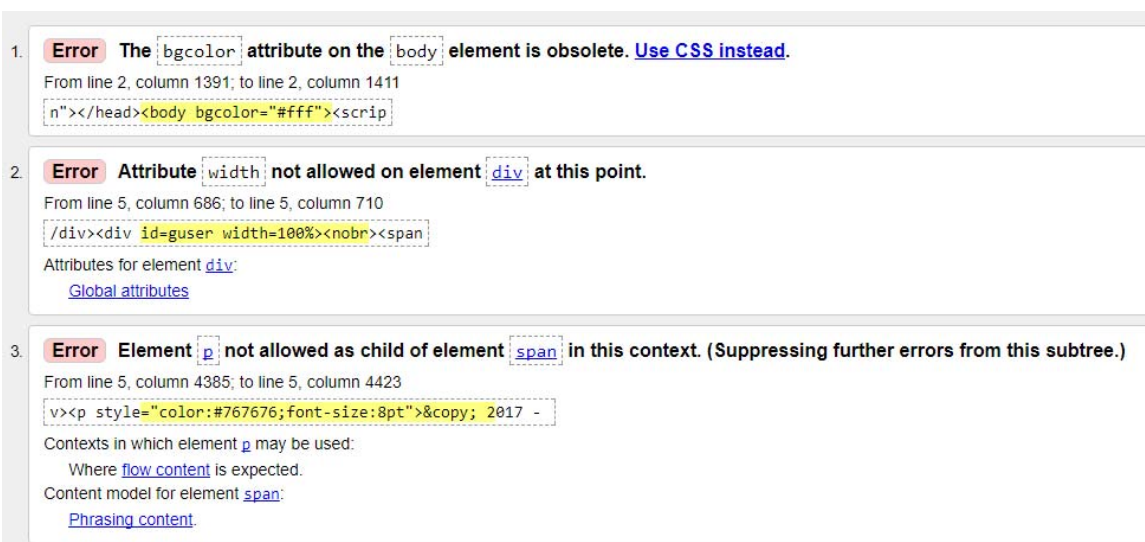
El servicio de Validación de CSS del W3C es un software libre creado por el W3C para ayudar a los diseñadores y desarrolladores web a validar Hojas de Estilo en Cascada (CSS). También ayuda a encontrar errores comunes, errores tipográficos, o usos incorrectos de CSS, además avisará cuando el CSS presente algún riesgo en cuanto a usabilidad.

Para el ejemplo se utilizará la url de google, y se mostraran 3 entre los errores que se encuentren.

### 4.1.2. Tipos de errores

Haciendo un chequeo de [www.google.es](http://www.google.es) en la web de validación, se presentan varios tipos de errores, tales como, declaración errónea de la codificación, atributos obsoletos dentro de los elementos y elementos no permitidos como hijos de otros en un contexto específico.

Para el análisis se han elegido los 3 errores que se pueden ver en la captura.



The screenshot displays three error messages from the W3C CSS Validator. Each error is numbered and includes a description, the location of the error in the CSS file, and a code snippet showing the problematic code.

- Error 1:** The `bgcolor` attribute on the `body` element is obsolete. [Use CSS instead.](#)  
From line 2, column 1391; to line 2, column 1411  
`<body bgcolor="#fff"><scrip`
- Error 2:** Attribute `width` not allowed on element `div` at this point.  
From line 5, column 686; to line 5, column 710  
`<div id=guser width=100%><nobr><span`  
Attributes for element `div`:  
[Global attributes](#)
- Error 3:** Element `p` not allowed as child of element `span` in this context. (Suppressing further errors from this subtree.)  
From line 5, column 4385; to line 5, column 4423  
`<p style="color:#767676;font-size:8pt">&copy; 2017 -`  
Contexts in which element `p` may be used:  
Where [flow content](#) is expected.  
Content model for element `span`:  
[Phrasing content](#).

#### 4.1.3. Soluciones

En referencia a la captura anterior, en el primer error se puede observar cómo indica que el tributo “bgcolor” dentro del elemento “body” está obsoleto, por lo que, la solución sería, como también indica en el error, incluir la etiqueta de “background-color” en una hoja de estilos. El segundo dice que el atributo “width” no está permitido en el elemento “div”, por lo que, una solución podría ser cambiarle el tamaño desde el CSS. Y por último en el tercer error indica que el elemento “p” no puede ser un elemento hijo de “span”, una posible solución sería eliminar la etiqueta “span”.