**COSAS QUE RESALTAR DESDE EL S0-S6**

**S3:**

Id = ” a” 🡪 #a

Class= ”a “ 🡪 .a

1. Cuando se aplican valores **contradictorios** a la misma propiedad de un elemento, el **último estilo** definido es el que prevalece.

**<p style="color:blue; color:green">Párrafo</p>**

Prevalece el último que es el green. (Lo mismo pasaría en una hoja de estilo css)

2. El **estilo** aplicado directamente sobre las **etiquetas HTML**, en el fichero html, tiene **PRIORIDAD** sobre el estilo aplicado en la **hoja de estilo. (HTML >>>>> CSS)**

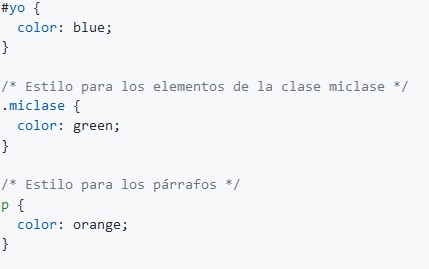
3. El que no tiene color hereda del más cercano.

4. Regla

<body>

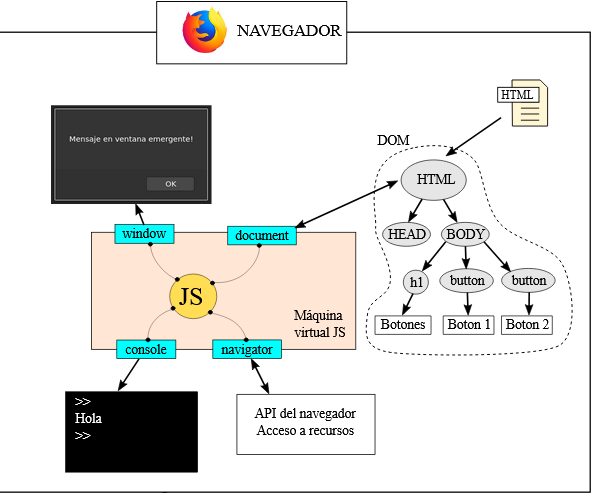
<p class="miclase" id="yo">Párrafo. ¿De qué color soy? </p>

</body>



El color sería blue, ya que es en html el último que se define el que tiene la prioridad, además hay que tener en cuenta que html tiene prioridad sobre css.

**S4:**

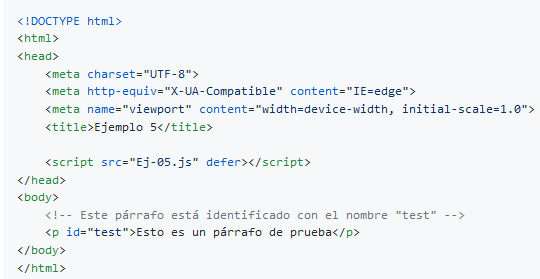


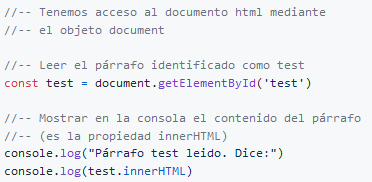
-Objeto console, pone el navegador a nuestra disposición, contiene el método log() para imprimir mensajes en la consola:

console.log("¡¡Hola Mundo!!")

-Para garantizar que nuestro código js se empiece a ejecutar una vez que el fichero html ha sido procesado completamente, y que por tanto tenemos acceso a él, utilizaremos el atributo **defer** en la etiqueta <script>.

Para localizar el párrafo identificado como **"test"** usamos el método **getElementById('test')**:

.html

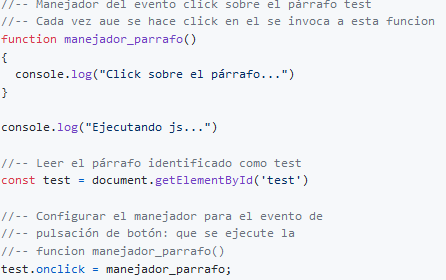
.js

Efectivamente, al ejecutarse el código javascript, ha impreso en la consola lo mismo que había en el párrafo en el HTML.

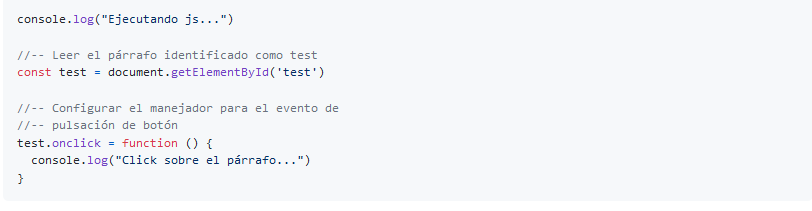
Estamos usando la función manejador\_parrafo **como un objeto,** y por eso **NO** usamos manejador\_parrafo() con los paréntesis al final. Si se ponen los paréntesis lo que se hace es llamar a la función primero y asignar a onclick el valor devuelto por ella. Es un **ERROR MUY COMUN, habría que hacer lo siguiente:**

test.onclick = manejador\_parrafo;

Pasamos de :



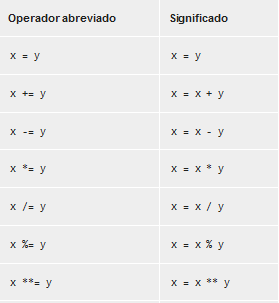
A:



El **comportamiento** es **exactamente el mismo** que antes: cada vez que se hace click sobre el párrafo, se ejecuta el código console.log("Click sobre el párrafo..."). La diferencia es que esta **notación** es **más compacta**, y nos **ahorramos** el tener que definir el nombre de una función. Lo cual tiene mucho sentido, porque esta función nunca será llamada desde otro lugar, sino que sólo se ejecuta al hacer click en el párrafo.

Realizamos el ejercicio 9…

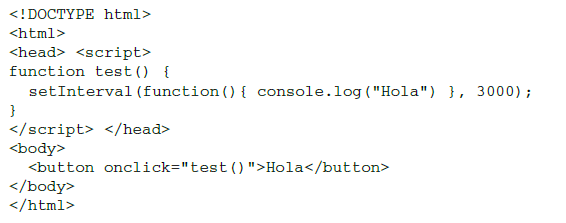
Operadores:



**Dom**=

**Hay tres tipos de conexiones en las que HTTP puede funcionar y son**: Conexión no persistente, conexión persistente sin pipelining y conexión persistente con pipelining. La conexión no persistente es la menos eficiente y la conexión no persistente con pipelining la mayor.

Al pinchar en el botón test ejecutas la función, y en la consola te sale hola, cada 3sh(3000ms):

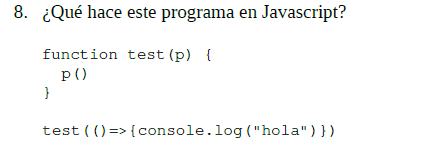


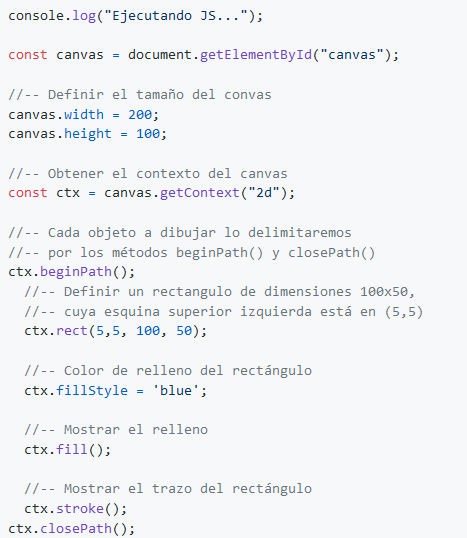
**Playout**= Es el tiempo transcurrido desde que se envía un paquete en el emisor y se empieza a reproducir en el receptor.

**DOM**= Es el modelo de objetos del documento. Es una interfaz estándar para acceder a todos los objetos que constituyen la página web, y poder manipularlos.

A los objetos del DOM se les asigna una identificación dando un valor al campo id en las etiquetas HTML. Ejemplo: <section id=”darth\_vader”> Se crea una sección nueva cuya identificación es la cadena darth\_vader.

**P FUNCION QUE LE PASAS COMO PARÁMETRO:**





Son coordenadas en el canvas.

