

Tabla de contenido

Ejercicio 1.....	1
Ejercicio 3.....	2
Conclusiones.....	3
Bibliografía y Fuentes consultadas.....	3

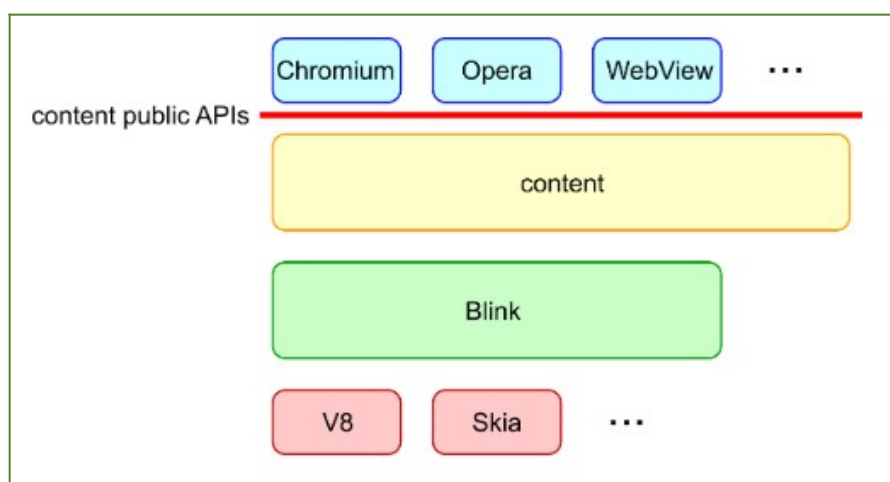
Ejercicio 1

Explica de forma concisa y clara el subsistema “Motor de renderizado” de la estructura de ejecución de un navegador. Especifica los utilizados actualmente por los navegadores principales.

El **motor de renderizado** se encarga de representar para el usuario (el cliente) una página web de manera visual, valiéndose tanto del HTML (lenguaje marcado para dar estructura), CSS (formateo y estilo de la información que se presenta) como de los demás recursos que incorpore la web (DOM, XML ...).

Cada cliente utiliza un motor de renderizado distinto, siendo los más utilizados en el mercado actualmente los siguientes:

- ➔ **Blink** : se desarrolló a partir de WebKit y esta embebido en Chromium, Android WebView y Opera a través de APIs públicas. Este motor es el más utilizado por diversos navegadores, como Google Chrome, Opera y Microsoft Edge.



- ➔ **Gecko** : es de código abierto y fue desarrollado por Netscape en 1997. Lo utiliza el cliente de Mozilla Firefox.
- ➔ **WebKit** : de software libre desde 2005, basado en el motor de renderizado KHTML (creado para el navegador Konqueror).

Ejercicio 3

Apartado 5 - averigua qué son las variables automáticamente globales. Pon un ejemplo

En JavaScript las variables pueden declararse como **var** o **let**, pero es posible declarar una variable sin una de estas palabras delante, aunque será tratada como var. Esta conversión al no poner el tipo de variable hace que esta sea global.

```
<script>

global = 3;
//Declaramos un variable global

function scope() {
  local = 4; //Declaramos una variable local dentro de la funcion
  console.log(global);
  console.log(local);
}

scope();

console.log(global);
console.log(local);

</script>
```

El siguiente código dará como resultado por consola:

- 3 //como se declara fuera de la función , dentro también la conoce
- 4 //se declara localmente dentro de la función
- 3 //esta declarada globalmente
- 4 //al no tener ninguna palabra reservada se considera **var**, por lo que no dará error

Conclusiones

En cuanto a la declaración de variables, es muy importante declarar las variables en el ámbito que deseemos (diferenciar entre variables locales y globales). Hay que tener en cuenta el scope, puesto que minimizará los problemas que podamos tener a largo plazo.

En cuanto al nombre de las variables es muy recomendable que este sea legible, evitando monosílabos (usar camelCase y mayúsculas para las constantes). En códigos de 20 líneas no habrá problemas, pero en desarrollos extenso puede ser muy tedioso saber qué significa la variable **v**.

Bibliografía y Fuentes consultadas

- Ejercicio 1:
 - Apuntes propios
 - [Definición de motor de renderizado](#)
 - [Motores de cada navegador](#)
 - [Blink](#)
 - [Imagen de blink](#)
 - [Gecko](#) y [historia](#)
 - [WebKit](#) y [KHTML](#)
- Ejercicio 2:
 - [GitHub](#)
- Ejercicio 3:
 - [GitHub](#)
 - [Variables automáticamente globales](#)
 - Imagen de ejemplo propia