

Tabla de contenido

Ejercicio 1.....	1
Ejercicio 2.....	2
Ejercicio 3.....	6
Conclusiones.....	12
Bibliografía y Fuentes consultadas	13

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.

Motor de renderizado (Motor de diseño , motor de representación , motor de navegación) Componente de software básico de los navegadores web . Interpretando así los archivos HTML y Javascript , presentándolos en pantalla o convirtiéndolos a pixeles .

La función principal de un motor de navegación es transformar los documentos HTML y otros recursos de una página web en una representación interactiva en el dispositivo del usuario (Conocida como renderización) .

El motor de renderizado , es un software que toma contenido marcado e información de formato , mostrando el contenido ya formateado en la pantalla de aplicaciones . (Usados en navegadores web , clientes de correo electrónico u otras aplicaciones) .

El proceso que realiza para interpretación de html , css y javascript hasta la conversión a pixeles se agrupa en 4 fases . Siendo así un análisis gramatical o parsing de html a dom . Interpretación del archivo CSS para cada nodo del DOM . Creación de un nuevo árbol que incluye el DOM , con estilos y despliegue de nodos . Pintar el árbol de render .

Qué motor de renderizado usan : (Webrender , Gecko , Weebit , Blink)

Opera → Blink .

Chrome → Blink , basado en las primeras versiones del moto WebKt .

Firefox → Gecko , Webrender

Safari → Webkit

Edge → Blink

Internet explorer → Trident

Motor	Estado	Desarrollador	Licencia	Incluido en
WebKit	Activo	Apple	GNU LGPL, BSD	Safari, y todos los navegadores de la App Store
Blink	Activo	Google	GNU LGPL, BSD	Google Chrome y todos los navegadores basados en Chromium, como Microsoft Edge, Brave y Opera
Gecko	Activo	Mozilla	MPL	Mozilla Firefox y el cliente de correo Mozilla Thunderbird, además de las bifurcaciones como SeaMonkey, Waterfox y Iccat
Servo	Activo	Mozilla	MPL	navegador experimental
Goanna	Activo	M. C. Straver ⁴	MPL	Pale Moon y Basilisk
NetSurf	Activo	hobbyists ⁵	GNU GPLv2	NetSurf ⁶
KHTML	Descontinuado	KDE	GNU LGPL	Konqueror
Trident	Descontinuado	Microsoft	Privativa	Internet Explorer y el cliente de correo Microsoft Outlook
EdgeHTML	Descontinuado	Microsoft	Privativa	anteriormente Microsoft Edge
Presto	Descontinuado	Opera Software	Privativa	anteriormente Opera

Sistemas operativos [\[editar \]](#)

Los sistemas operativos en los cuales el motor de renderizado puede ejecutarse de forma nativa.

Engine	Windows	macOS	iOS	Android	Linux	BSD
WebKit	Si	Si	Si	Si	Si	Si
Blink ⁷	Si	Si	Desconocido	Si	Si	Si
Gecko	Si	Si	Desconocido	Si	Si	Si
Servo ⁸	Si	Si	Desconocido	Si	Si	Desconocido
Goanna ⁹	Si	Si	No	No	Si	Si
NetSurf ¹⁰	Si	Si	No	No	Si	Si

Ejercicio 2

Declara seis variables utilizando nombres acordes a su contenido:

- el precio de una sola rosa (8) y el número de rosas que tienes (70)
- el precio de un solo lirio (10) y el número de lirios que tienes (50)
- el precio de un solo tulipán (2) y la cantidad de tulipanes que tienes (120)
 - a) Ahora declara tres variables, una para cada una de las rosas, lirios y tulipanes que tienes, en las que colocas su precio total. Inserta los valores correspondientes en las variables utilizando las variables declaradas en el paso anterior.
 - b) Finalmente, declara una variable en la que almacenes el precio de todas tus flores (nuevamente, usa las variables anteriores para la inicialización). Muestra toda la información del inventario en la consola de la siguiente forma:

Rosa: precio unitario: 8 , cantidad: 70 , valor: 560

Lirio: precio unitario: 10 , cantidad: 50 , valor: 500

Tulipán: precio unitario: 2 , cantidad: 120 , valor: 240

Total: 1300

- Declaración de las variables que vamos a usar , con los valores que obtendrán realizando la multiplicación del número total por el valor (precio) de la flor .

```
<title>Ejercicio 2 </title>
<script>

// DECLARACIÓN DE VARIABLES : ( ROSAS , LIRIOS , TULIPAN , VALORES Y PRECIOS ( DE TODAS ))
// ROSAS :
let precioRosa = 8;
let numeroRosas = 70;

// LIRIOS :
let precioLirio = 10;
let numeroLirios = 50;

// TULIPAN :
let precioTulipan = 2;
let numeroTulipan = 120;

// VALORES ( MULTIPLICANDO EL VALOR POR EL NUMERO TOTAL)
let valorRosas = precioRosa * numeroRosas;
let valorLirios = precioLirio * numeroLirios;
let valorTulipan = precioTulipan * numeroTulipan;

// VARIABLE TOTAL ( SUMA DE VALORES DE TODAS LAS FLORES ):
let total = valorRosas + valorLirios + valorTulipan;
```

// DECLARACIÓN DE VARIABLES : (ROSAS , LIRIOS , TULIPAN , VALORES Y PRECIOS (DE TODAS))

// ROSAS :

let precioRosa = 8;

let numeroRosas = 70;

// LIRIOS :

let precioLirio = 10;

let numeroLirios = 50;

```
// TULIPAN :
```

```
let precioTulipan = 2;
```

```
let numeroTulipan = 120;
```

```
// VALORES ( MULTIPLICANDO EL VALOR POR EL NUMERO TOTAL)
```

```
let valorRosas = precioRosa * numeroRosas;
```

```
let valorLirios = precioLirio * numeroLirios;
```

```
let valorTulipan = precioTulipan * numeroTulipan;
```

```
// VARIABLE TOTAL ( SUMA DE VALORES DE TODAS LAS FLORES ):
```

```
let total = valorRosas + valorLirios + valorTulipan;
```

- Continuamos mostrando con console.log , los valores y concatenando cadena de texto con las variable ya declaradas .

```
// MOSTRANDO POR CONSOLA EL VALOR , CANTIDAD Y PRECIO DE TODAS LAS FLORES ( USANDO CONSOLE.LOG ):  
// CONCATENACIÓN DE CADENA DE TEXTO CON VARIABLES , SIEMPRE USANDO ""+VARIABLE+"" :  
console.log("Rosa: precio unitario: " + precioRosa + ", cantidad: " + numeroRosas + ", valor: " + valorRosas);  
console.log("Lirio: precio unitario: " + precioLirio + ", cantidad: " + numeroLirios + ", valor: " + valorLirios);  
console.log("Tulipan: precio unitario: " + precioTulipan + ", cantidad: " + numeroTulipan + ", valor: " + valorTulipan);  
  
// MOSTRANDO EL TOTAL .  
console.log("Total: " + total);
```

```
// MOSTRANDO POR CONSOLA EL VALOR , CANTIDAD Y PRECIO DE TODAS LAS  
FLORES ( USANDO CONSOLE.LOG ):
```

```
// CONCATENACIÓN DE CADENA DE TEXTO CON VARIABLES , SIEMPRE USANDO  
""+VARIABLE+"" :
```

```
console.log("Rosa: precio unitario: " + precioRosa + ", cantidad: " + numeroRosas +  
", valor: " + valorRosas);
```

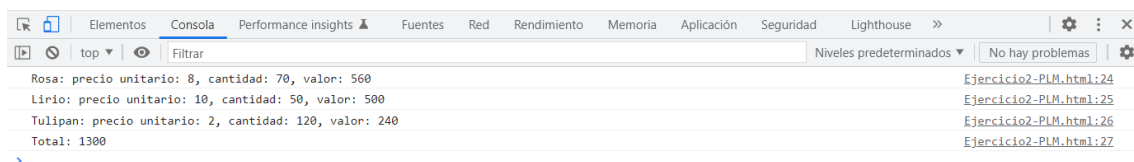
```
console.log("Lirio: precio unitario: " + precioLirio + ", cantidad: " + numeroLirios + ",  
valor: " + valorLirios);
```

```
console.log("Tulipan: precio unitario: " + precioTulipan + ", cantidad: " +  
numeroTulipan + ", valor: " + valorTulipan);
```

```
// MOSTRANDO EL TOTAL .
```

```
console.log("Total: " + total);
```

- El resultado obtenido lo podemos ver en nuestro navegador , en este caso Google Chrome . Pulsando botón derecho inspeccionar , o f12 y abrimos DEVS TOOLS .
Buscamos en consola , y observamos el resultado :



Como podemos ver , nos sale directamente le texto concatenado de las variables , y en el último el valor del total , es decir el resultado de la operación que declaramos en la variable arriba .

Ejercicio 3

Uso de “var” en JavaScript

1. Define una variable saludo=“Hola” como variable local dentro de una función saludar() que muestre una ventana emergente con el valor de la variable saludo
 - Invoca la función
 - Muestra la variable desde fuera de la función mediante un window.alert
2. Define una variable global despedida=“Adiós” fuera de una función despedir() que muestre una ventana emergente con el valor de la variable despedida.
 - Invoca la función
 - Muestra la variable desde fuera de la función
3. Define dos variables con el mismo nombre y distinto valor: una fuera y otra dentro de una función ámbito() que muestre un window.alert de la misma. Muestra el valor de las dos mediante window.alert e invocando a la función.
4. Declara y define una variable global. Redefínela dentro de una función que la muestre mediante un window.alert. Muestra el valor de la variable mediante window.alert e invocando a la función.
5. Averigua qué son las variables automáticamente globales. Pon un ejemplo

- APARTADO 1 :

```
// APARTADO 1 :  
// FUNCIÓN SALUDAR (MOSTRANDO UNA VARIABLE LOCAL CON UN WINDOW.ALERT(VARIABLE)):  
function saludar ()  
{  
    // VARIABLE LOCAL SALUDO EN LA FUNCION SALUDAR :  
    var saludo = "Hola";  
  
    // VENTANA EMERGENTE CON LA VARIABLE LOCAL SALUDO .  
    window.alert(saludo);  
    // PARA COMPROBAR VALORES DESDE DENTRO DE LA CONSOLE DEL PROPIO INSPECCIONAR DEL NAVEGADOR WEB  
    console.log(saludo);  
}  
  
// INTENTAMOS MOSTRAR LA VARIABLE LOCAL DE LA FUNCIÓN DESDE FUERA .  
window.alert(saludo);  
// LLAMADA A LA FUNCIÓN SALUDAR PARA QUE SE EJECUTE EN EL HTML .  
saludar();
```

Y en el intento de mostrar la variable desde fuera , nos salta un error en consola undefined .

Esta página dice

Hola

Aceptar

```
// APARTADO 1 :  
// FUNCIÓN SALUDAR (MOSTRANDO UNA VARIABLE LOCAL CON UN  
WINDOW.ALERT(VARIABLE));  
function saludar ()  
{  
    // VARIABLE LOCAL SALUDO EN LA FUNCION SALUDAR :  
    var saludo = "Hola";  
  
    // VENTANA EMERGENTE CON LA VARIABLE LOCAL SALUDO .  
    window.alert(saludo);  
    // PARA COMPROBAR VALORES DESDE DENTRO DE LA CONSOLE DEL PROPIO  
    INSPECCIONAR DEL NAVEGADOR WEB  
    console.log(saludo);  
}  
  
// INTENTAMOS MOSTRAR LA VARIABLE LOCAL DE LA FUNCIÓN DESDE FUERA .  
window.alert(saludo);  
// LLAMADA A LA FUNCIÓN SALUDAR PARA QUE SE EJECUTE EN EL HTML .  
saludar();
```

En esta primera parte , declaramos una función que tiene una variable local , que mostraremos con un window.alert en la página .Intentamos ejecutar la variable local desde fuera de la función .
Y por último ejecutamos la función .

- APARTADO 2 :

```
// APARTADO 2 :  
// VARIABLE GLOBAL FUERA DE LA FUNCIÓN (DESPEDIDA)  
var despedida = "Adiós";  
  
// FUNCIÓN DESPEDIR QUE TOMA EL VALOR DE LA VARIABLE DECLARADA FUERA DE DICHA FUNCIÓN :  
function despedir(despedida)  
{  
    window.alert(despedida);  
}  
  
// LLAMADA A LA FUNCIÓN PARA QUE SE EJECUTE LA FUNCIÓN DESPEDIR .  
despedir(despedida);  
// MOSTRAR LA VARIABLE DESDE FUERA DE LA FUNCIÓN .  
console.log(despedida);
```

```
// APARTADO 2 :  
// VARIABLE GLOBAL FUERA DE LA FUNCIÓN (DESPEDIDA)  
var despedida = "Adiós";  
  
// FUNCIÓN DESPEDIR QUE TOMA EL VALOR DE LA VARIABLE DECLARADA FUERA  
DE DICHA  
function despedir(despedida)  
{  
    window.alert(despedida);  
}  
  
// LLAMADA A LA FUNCIÓN PARA QUE SE EJECUTE LA FUNCIÓN DESPEDIR .  
despedir(despedida);  
// MOSTRAR LA VARIABLE DESDE FUERA DE LA FUNCIÓN .  
console.log(despedida);
```

En este apartado declaramos una variable global fuera de la función en la que pasaremos como parámetros en la función despedir , y mostraremos con un window.alert .

Posteriormente llamamos a la función para ejecutarla .

Y después mostramos la variable global con un console.log , para verlo en la consola .

Esta página dice

Adiós

Aceptar

- APARTADO 3:

```
// APARTADO 3 :
// DEFINIR UNA VARIABLE GLOBAL , QUE VA A TENER EL MISMO NOMBRE QUE LA QUE VAMOS A DECLARAR EN LA FUNCIÓN ÁMBITO
// PERO TENDRAN VALORES DISTINTOS .

// DECLARACIÓN DE VARIABLE GLOBAL NOMBRE (PABLO):
var nombre = "Pablo";

// DECLARACIÓN DE LA FUNCIÓN ÁMBITO QUE MOSTRARÁ CON UN WINDOW.ALERT LOS VALORES
// DE DICHAS VARIABLES QUE HEMOS Y VAMOS A DECLARAR
function ambito(nombre)
{
    window.alert(nombre);

    // DECLARACIÓN DE VARIABLE LOCAL EN LA FUNCIÓN QUE SE SOBREESCRIBE EN EL ANTERIOR VALOR .
    var nombre = "Alex" ;

    window.alert(nombre);
}

// PARA QUE FUNCIONE EL PASO DE LOS DATOS DE LA VARIABLE GLOBAL EN LA FUNCIÓN ÁMBITO ,
// DEBEMOS PASARLE COMO PARAMETRO LA VARIABLE O DIRECTAMENTE EL VALOR POR LO QUE SINO ,
// LA VARIABLE EN EL MOMENTO DE MOSTRARLA SERÍA UNDEFINED , LUEGO AL DECLARAR LA NUEVA VARIABLE ,
// AUTOMATICAMENTE SE SOBREESCRIBE EL VALOR , Y MUESTRA EL NUEVO NOMBRE EN ESTE CASO ALEX .
ambito(nombre);
```

// APARTADO 3 :

// DEFINIR UNA VARIABLE GLOBAL , QUE VA A TENER EL MISMO NOMBRE QUE LA
QUE VAMOS A DECLARAR EN LA FUNCIÓN ÁMBITO
// PERO TENDRAN VALORES DISTINTOS .

// DECLARACIÓN DE VARIABLE GLOBAL NOMBRE (PABLO):
var nombre = "Pablo";

// DECLARACIÓN DE LA FUNCIÓN ÁMBITO QUE MOSTRARÁ CON UN
WINDOW.ALERT LOS VALORES

// DE DICHAS VARIABLES QUE HEMOS Y VAMOS A DECLARAR
function ambito(nombre)
{

 window.alert(nombre);

 // DECLARACIÓN DE VARIABLE LOCAL EN LA FUNCIÓN QUE SE SOBREESCRIBE EN
EL ANTERIOR VALOR .

 var nombre = "Alex" ;

 window.alert(nombre);

}

// PARA QUE FUNCIONE EL PASO DE LOS DATOS DE LA VARIABLE GLOBAL EN LA
FUNCIÓN ÁMBITO ,
// DEBEMOS PASARLE COMO PARAMETRO LA VARIABLE O DIRECTAMENTE EL
VALOR POR LO QUE SINO ,
// LA VARIABLE EN EL MOMENTO DE MOSTRARLA SERÍA UNDEFINED , LUEGO AL
DECLARAR LA NUEVA VARIABLE ,

```
// AUTOMATICAMENTE SE SOBREScribe EL VALOR , Y MUESTRA EL NUEVO  
NOMBRE EN ESTE CASO ALEX .  
ambito(nombre);
```

Esta página dice

Pablo

Aceptar

Declaramos una variable global desde fuera que pasaremos como parámetros en la función ámbito , que en primer lugar mostraremos con un window.alert su valor , y posteriormente declararemos otra variable local a la que le modificaremos su valor , y por consiguiente cambiara su valor a mostrarlo de nuevo .

Esta página dice

Alex

Aceptar

- APARTADO 4 :

```
// APARTADO 4 :  
  
// DECLARACIÓN DE LA VARIABLE GLOBAL , QUE PASAREMOS A LA FUNCIÓN REMODELAR , A LA QUE LE AÑADIREMOS  
// MÁS CONTENIDO QUE MOSTRAREMOS CON UN WINDOW.ALERT  
var variableGlobal = "Hoy es jueves" ;  
  
// DECLARACIÓN DE LA FUNCIÓN REMODELAR , QUE RECIBE LA VARIABLEGLOBAL :  
function remodelar(variableGlobal)  
{  
    // AÑADIMOS CONTENIDO A LA VARIABLE GLOBAL QUE TENIAMOS EN EL PRINCIPIO  
    variableGlobal += ", y mañana es viernes ( PABLO )";  
  
    // MOSTRAMOS EL CONTENIDO DE LA VARIABLE .  
    window.alert(variableGlobal);  
}  
  
// LLAMAMOS A LA FUNCIÓN REMODELAR QUE NOS RETORNA UN WONDOW.ALERT CON EL CONTENIDO ( VALOR )  
// DE LA VARIABLEGLOBAL .  
remodelar(variableGlobal);  
  
</script>
```

```
// APARTADO 4 :

// DECLARACIÓN DE LA VARIABLE GLOBAL , QUE PASAREMOS A LA FUNCIÓN
REMODELAR , A LA QUE LE AÑADIREMOS
// MÁS CONTENIDO QUE MOSTRAREMOS CON UN WINDOW.ALERT
var variableGlobal = "Hoy es jueves" ;

// DECLARACIÓN DE LA FUNCIÓN REMODELAR , QUE RECIBE LA VARIABLEGLOBAL :
function remodelar(variableGlobal)
{
    // AÑADIMOS CONTENIDO A LA VARIABLE GLOBAL QUE TENIAMOS EN EL
    PRINCIPIO
    variableGlobal += ", y mañana es viernes ( PABLO )";

    // MOSTRAMOS EL CONTENIDO DE LA VARIABLE .
    window.alert(variableGlobal);
}

// LLAMAMOS A LA FUNCIÓN REMODELAR QUE NOS RETORNA UN
WONDO.ALERT CON EL CONTENIDO ( VALOR )
// DE LA VARIABLEGLOBAL .
remodelar(variableGlobal);
```

Esta página dice

Hoy es jueves, y mañana es viernes (PABLO)

Aceptar

Observando que es una addición a la variable ya definida con un valor , también valdría con hacer un = y cambiarle el valor , y se redefiniría igualmente .

En este apartado declaramos una variable global , a la que cuando declaremos la función remodelar , lo que hará es , añadirle contenido a la variable global que encontramos fuera de la variable , y por último mostrarla con un window.alert .

- APARTADO 5 :

Las variables automáticamente globales las encontramos declaradas normalmente en la parte superior , en el comienzo de lo que sería el script , en ocasiones podemos encontrarlo entre código ,o incluso dentro de una función. Por norma general se encuentra en la parte superior debido a que se suelen declarar las variables ahí para generalizar , el código este ordenado , aunque no influye que está debajo de funciones , o en funciones , debido a que la lectura la realizará igual .

Todo esto es debido al trato del ámbito de la variable , en el que podemos diferenciar con una variable local , que esta definida dentro de la función , siendo el caso de que en el caso de que no estén declaradas con “var” pasan a ser globales , como por ejemplo el uso de let .

Y por último una variable global es la que podemos encontrar en cualquier parte del programa.

Concluyendo que las variables que se encuentran definidas fuera de las funciones siempre van a ser globales , y en el caso de encontrarlas dentro de la función pueden ser locales o globales .

Conclusiones

Para finalizar esta práctica , el primer ejercicio me ha servido un poco de información , de entender lo que es un motor de renderizado . Continuando con el segundo ejercicio , en el que hemos pasado a la parte un poco más práctica , utilizando consultas importantes como pueden ser ver en la consola del propio devs tolls mensajes , haciendo un ejercicio de declaración de variables , para mostrarlas directamente por consola , y creando una ultima variable que nos proporciona y vemos que se pueden realizar sumas de operadores . En el tercer ejercicio he comprendido el uso de las variables locales y globales , observando su diferencia y las distintas maneras de tratar las funciones dependiendo de sus ámbitos , viendo así las facilidades y los inconvenientes que nos pueden surgir dependiendo de donde declaremos o de donde usemos las variables . Todo dependiendo de los ámbitos , ya sea uno externo o interno de la función .

Para concluir , esta práctica , me ha servido para informarme y adquirir los conceptos básicos de los comienzos de Javascript , siendo así una manera de iniciarme y ver las diferencias de ámbitos , variables.

Bibliografía y Fuentes consultadas

- Enlaces :
 - [Ambito de las variables en Javascript \(desarrolloweb.com\)](https://desarrolloweb.com/es-es/variables-en-javascript/)
 - <https://learn.microsoft.com/es-es/managed-desktop/get-started/edge-browser-app>
 - [https://es.wikipedia.org/wiki/Anexo:Comparativa de motores de renderizado](https://es.wikipedia.org/wiki/Anexo:Comparativa_de_motores_de_renderizado)