

MODULO 1

Ejercicio 06* Javascript

Descripción breve

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales (HTML y CSS) hemos cubierto con mucho más detalle en otras partes del Área de aprendizaje.

Carlos Arias

carlos.arias@alumnos.udg.mx



Contenido

Contenido	2
Bibliografía	11

Contenido

1. ¿Qué es JavaScript?

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado.

2. ¿Qué hace JavaScript?

Permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás.

3. ¿Qué es el DOM?

Es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página.

4. ¿Qué es un APIs Web?

Son construcciones disponibles en los lenguajes de programación que permiten a los desarrolladores crear funcionalidades complejas de una manera simple.

5. Mencionan como minio 5 ejemplos de Apis Web

- ClassList
- Dataset
- DOM
- Drag and Drop
- EventSource

6. ¿Qué es lo que JavaScript hace en tu página web?

Permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente todo lo demás.

7. Menciona un ejemplo de cómo añadir JavaScript a tu página web

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<script>JAVASCRIPT IS USUALLY PLACED HERE</script>
<title>Time right now is: </title>
</head>
<body>
<script>JAVASCRIPT CAN ALSO GO HERE</script>
</body>
</html>
```

8. ¿Cuáles son las dos maneras de añadir código JavaScript a tu sitio web?

La primera forma de insertar JavaScript en HTML es directa. Puedes hacerlo utilizando la etiqueta `<script>` `</script>` que debe envolver todo el código JS que escribas. Se puede agregar el código JS:

- entre las etiquetas `<head>`
- entre las etiquetas `<body>`

La segunda es importando archivos externos. Se puede hacer referencia a estos archivos desde adentro de los documentos HTML, al igual que hacemos referencia a los documentos CSS. Algunos de los beneficios de agregar código JS en archivos separados son:

- Cuando los códigos HTML y JavaScript están separados, se cumple el principio de diseño de separación y hace que todo sea mucho más sostenible y reutilizable.
- La legibilidad y el mantenimiento del código es mucho más fácil.
- Los archivos JavaScript en caché mejoran el rendimiento general del sitio web al disminuir el tiempo que tardan las páginas en cargarse.

9. ¿Cómo se comenta el código en JavaScript?

```
<script> </script>
```

10. ¿Cómo se declara una variable y cuáles son sus principales características?

Var

11. ¿Cómo se declara una condición y cuáles son sus principales características?

condición

Una expresión que se evalúa como true o false.

expr1, expr2

Expresión con valores de algún tipo.

12. ¿Cómo se declara bucle y cuáles son sus principales características?

for (inicializador; condición de salida; expresión final) {

// código a ejecutar

}

La palabra reservada for, seguida por algunos paréntesis.

Dentro de los paréntesis tenemos tres ítems, separados por punto y coma (;):

Un inicializador - Este es usualmente una variable con un número asignado, que aumenta el número de veces que el bucle ha sido ejecutado. También se le llama contador o variable de conteo.

Una condición de salida - como se mencionó previamente, ésta define cuando el bucle debería detenerse. Generalmente es una expresión que contiene un operador de comparación, una prueba para verificar si la condición de término o salida ha sido cumplida.

Una expresión final - que es siempre evaluada o ejecutada cada vez que el bucle ha completado una iteración. Usualmente sirve para modificar al contador (incrementando su valor o algunas veces disminuyéndolo), para aproximarse a la condición de salida.

Algunos corchetes curvos que contienen un bloque de código - este código se ejecutará cada vez que se repita el bucle.

13. ¿Cómo se declara una función y cuáles son sus principales características?

Una definición de función (también denominada declaración de función o expresión de función) consta de la palabra clave function, seguida de:

El nombre de la función.

Una lista de parámetros de la función, entre paréntesis y separados por comas.

Las declaraciones de JavaScript que definen la función, encerradas entre llaves, { ... }.

14. ¿Cómo se declara un objeto y cuáles son sus principales características?

Un objeto de JavaScript tiene propiedades asociadas a él. Una propiedad de un objeto se puede explicar como una variable adjunta al objeto. Las propiedades de un objeto básicamente son lo mismo que las variables comunes de JavaScript, excepto por el nexos con el objeto. Las propiedades de un objeto definen las características del objeto. Accedes a las propiedades de un objeto con una simple notación de puntos:

objectName.propertyName

15. ¿Cómo se puede leer un evento con JavaScript?

Además del objeto event descrito aquí, el Gecko DOM también proporciona métodos para registrar oyentes de eventos en nodos en el DOM, quitando aquellos oyentes de eventos, y enviando eventos desde el DOM.

16. ¿Qué eventos podemos leer con JavaScript mencionales y descríbalos?

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

17. Mencionan un ejemplo donde JavaScript sea ejecutado del lado del cliente

```
GET https://developer.mozilla.org/en-US/search?q=client+server+overview&topic=apps&topic=html&topic=css&topic=js&topic=api&topic=webdev HTTP/1.1
Host: developer.mozilla.org
Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: https://developer.mozilla.org/en-US/
Accept-Encoding: gzip, deflate, sdch, br
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7
Accept-Language: en-US,en;q=0.8,es;q=0.6
Cookie: sessionid=6ynxs23n521lu21b1t136rhbv7ezngie;
csrftoken=zIPUJsAZv6pcgCBJSCj1zU6pQZbfMUAT; dwf_section_edit=False;
dwf_sg_task_completion=False; _gat=1; _ga=GA1.2.1688886003.1471911953;
ffo=true
```

18. Mencionan un ejemplo donde JavaScript sea ejecutado del lado del servidor

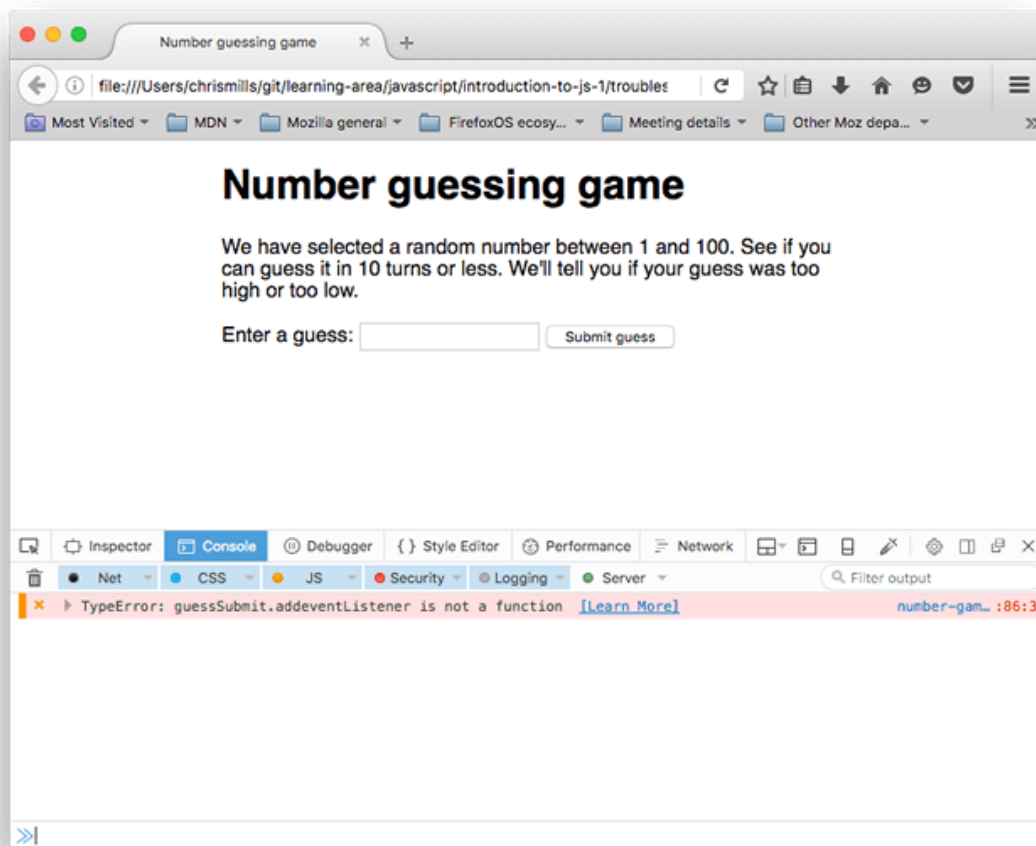
```
HTTP/1.1 200 OK
Server: Apache
X-Backend-Server: developer1.webapp.scl3.mozilla.com
Vary: Accept, Cookie, Accept-Encoding
Content-Type: text/html; charset=utf-8
Date: Wed, 07 Sep 2016 00:11:31 GMT
Keep-Alive: timeout=5, max=999
Connection: Keep-Alive
X-Frame-Options: DENY
Allow: GET
X-Cache-Info: caching
Content-Length: 41823
```

```
<!DOCTYPE html>
<html lang="en-US" dir="ltr" class="redesign no-js" data-ffo-
opensanslight=false data-ffo-opensans=false >
<head prefix="og: http://ogp.me/ns#">
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=Edge">
  <script>(function(d) { d.className = d.className.replace(/\bno-js/,
  ''); }) (document.documentElement);</script>
  ...
```

19. ¿Qué tipo de errores podemos encontrar en JavaScript?

Errores de sintaxis, y errores lógicos

20. ¿De qué manera podemos ver los errores de JavaScript en el navegador?



21. ¿Cómo puedo convertir Strings a Objetos?

```
var myobj = JSON.parse('{ "hello":"world" }');  
alert(myobj.hello); // 'world'
```

22. ¿Cómo se usan los arreglos en JavaScript?

Las matrices se construyen con corchetes, que contiene una lista de elementos separados por comas.

- Digamos que queríamos almacenar una lista de compras en una matriz — haríamos algo como lo siguiente. Ingresas las siguientes líneas en la consola:

```
let shopping = ['bread', 'milk', 'cheese', 'hummus', 'noodles'];  
shopping;
```


23. ¿Cuántas versiones de JavaScript han existido menciona sus principales características?

- Javascript 1: nació con el Netscape 2.0 y soportaba gran cantidad de instrucciones y funciones, casi todas las que existen ahora ya se introdujeron en el primer estandar.
- Javascript 1.1: Es la versión de Javascript que se diseñó con la llegada de los navegadores 3.0. Implementaba poco más que su anterior versión, como por ejemplo el tratamiento de imágenes dinámicamente y la creación de arrays.
- Javascript 1.2: La versión de los navegadores 4.0. Esta tiene como desventaja que es un poco distinta en plataformas Microsoft y Netscape, ya que ambos navegadores crecieron de distinto modo y estaban en plena lucha por el mercado.
- Javascript 1.3: Versión que implementan los navegadores 5.0. En esta versión se han limado algunas diferencias y asperezas entre los dos navegadores.
- Javascript 1.5: Versión actual, en el momento de escribir estas líneas, que implementa Netscape 6.
- Por su parte, Microsoft también ha evolucionado hasta presentar su versión 5.5 de JScript (así llaman al javascript utilizado por los navegadores de Microsoft).
- En 2009 se publica la versión de ECMAScript 5. Esta es la versión que vamos a tratar principalmente en el manual, que funciona en todos los navegadores del mercado.
- En 2015 se lanza la sexta versión de ECMAScript, con una cantidad muy importante de novedades. Esta versión se puede usar completamente, excepto en Internet Explorer. Sus mejoras son tan importantes que todos los desarrolladores profesionales las usan y, si fuera necesario, se traduce el código para soporte a navegadores antiguos. Esta versión se explica en el Manual de ES6.
- Actualmente se ha adquirido un compromiso de liberar una versión del estándar ECMAScript por año. Por ello existen diversos estándares con pequeños cambios incrementales, que se han ido incorporando a distintos ritmos entre los navegadores.

24. ¿Cuántas versiones de ECMAScript han existido menciona sus principales características?

Edición	Fecha de publicación	Cambios desde la edición anterior
1	Junio de 1997	Primera edición
2	Junio de 1998	Cambios editoriales para mantener la especificación completa alineada con el estándar internacional ISO/IEC 16262
3	Diciembre de 1999	Se agregaron expresiones regulares, mejor manejo de strings, nuevo control de declaraciones, manejo de excepciones con try/catch, definición más estricta de errores, formato para la salida numérica y otras mejoras.
4	Abandonado	La cuarta edición fue abandonada debido a diferencias políticas respecto a la complejidad del lenguaje. Muchas características propuestas para la cuarta edición fueron completamente abandonadas; algunas fueron propuestas para la edición ECMAScript Harmony.
5	Diciembre de 2009	Agrega el modo estricto ("strict mode"), un subconjunto destinado a proporcionar una mejor comprobación de errores y evitar constructores propensos a errores. Aclara varias ambigüedades de la tercera edición, y afina el comportamiento de las implementaciones del "mundo real" que difieren consistentemente desde esa especificación. Agrega algunas nuevas características, como getters y setters, librería para el soporte de JSON, y una más completa reflexión sobre las propiedades de los objetos. ¹¹
5.1	Junio de 2011	Esta edición 5.1 de la ECMAScript Standard está completamente alineada con la tercera edición del estándar internacional ISO/IEC 16262:2011.
6	Junio de 2015 ¹²	La sexta edición agrega cambios significativos en la sintaxis para escribir aplicaciones complejas, incluyendo clases y módulos, definiéndolos semánticamente en los mismos términos del modo estricto de la edición ECMAScript 5. Otras nuevas características incluyen iteradores <code>for/of</code> loops, generadores y generador de expresiones estilo <code>Python</code> , funciones de dirección, datos binarios, colecciones (mapas, sets, mapas débiles), y proxies (?metaprogramación para objetos virtuales y wrappers). Al ser la primera especificación "ECMAScript Harmony", es también conocida como "ES6 Harmony".
7	Junio de 2016	La séptima edición fue una mera actualización de la versión 6. Incorpora el método <code>Array.prototype.includes()</code> y el operador exponencial (**).
8	Junio de 2017	La 8ª edición, oficialmente conocida como ECMAScript 2017, fue finalizada en junio de 2017 ¹¹ . Incluye constructores <code>async/await</code> , los cuales funcionan usando generadores y promesas.
9	Junio de 2018	La 9ª edición, oficialmente conocida como ECMAScript 2018, incluye operadores rest/spread para variables (tres puntos: <code>...identificador</code>), iteración asincrónica, <code>Promise.prototype.finally()</code> .
10	Enero de 2019	La 10ª edición, oficialmente conocida como ECMAScript 2019, incorporó <code>Array.flat()</code> , <code>Array.flatMap()</code> , <code>String.trimStart()</code> , <code>String.trimEnd()</code> , errores opcionales en el bloque <code>catch</code> , <code>Object.fromEntries()</code> , <code>Symbol.description</code> .

25. ¿Qué es Firefox Developer Edition?

Es el primer navegador web especialmente pensado para desarrolladores e incluye las últimas funcionalidades y herramientas para desarrollo y depuración de aplicaciones web

26. ¿Cuáles son las principales funciones y características de Firefox Developer Edition?

- Explorador de HTML y CSS.
- Consola de depuración.
- Depurador de Javascript: posibilidad de establecer puntos de parada, ejecución paso a paso, etc.
- Editor de estilos: edición de estilos CSS en vivo.
- Diseño web adaptable: visualización de páginas con diferentes tamaños de pantalla.
- Monitorización de red y rendimiento: inspección y edición de cabeceras, tiempos de respuesta, etc.
- Scratchpad: edición y ejecución de Javascript en vivo.

27. ¿Qué es WebGL?

WebGL es la tecnología más reciente para visualizar contenido 3D en la red.

28. ¿Qué APIs existen para APIs for game development?

Call of Duty: Modern Warfare	Best for COD & Warzone Data
RAWG Video Games Database	Best for Video Game Information
Chicken Coop	Best for Metacritic Data
Pokemon Go	Best for Pokemon Go Data
PokeAPI	Best for General Pokemon Data
IGDB	Best for Video Game Database
SteamWeb	Best for Steam Stats & User Data
Dota2	Best for Dota2 League & Player Data
League of Legends	Best for LoL (League of Legends) Game Data
Minecraft	Best for Minecraft Server Data

29. ¿Qué es un webSocket?

Es una tecnología avanzada que hace posible abrir una sesión de comunicación interactiva entre el navegador del usuario y un servidor.

30. ¿Cómo gestiona la memoria JavaScript?

Sin importar el lenguaje de programación, el ciclo de memoria es casi siempre parecido al siguiente:

Reservar la memoria necesaria

Utilizarla (lectura, escritura)

Liberar la memoria una vez ya no es necesaria.

El primer y el segundo paso son explícitos en todos los lenguajes. El último paso es explícito en lenguajes de bajo nivel, pero es mayormente implícito en lenguajes de alto nivel como JavaScript

Bibliografía

https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

<https://desarrolloweb.com/articulos/que-es-el-dom.html>

https://developer.mozilla.org/es/docs/Learn/JavaScript/Client-side_web_APIs/Introduction

<https://www.htmlcinco.com/15-apis-de-html5/>

<https://www.hostinger.mx/tutoriales/insertar-javascript-en-html/>

<https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/var>

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Conditional_Operator

https://developer.mozilla.org/es/docs/Learn/JavaScript/Building_blocks/Looping_code

<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Functions>

https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Working_with_Objects

https://www.w3schools.com/js/js_events.asp

https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Client-Server_overview

https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_went_wrong

<https://stackoverflow.com/questions/1086404/string-to-object-in-js>

https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Arrays

<https://desarrolloweb.com/articulos/494.php>

<https://es.wikipedia.org/wiki/ECMAScript>

<https://www.conasa.es/blog/mozilla-lanza-su-navegador-web-para-desarrolladores-firefox-developer-edition/>

<http://learn.sculpteo.com/qu%C3%A9-es-webgl-c%C3%B3mo-se-activa-webgl>

<https://rapidapi.com/blog/top-video-game-apis/>

https://developer.mozilla.org/es/docs/Web/API/WebSockets_API

https://developer.mozilla.org/es/docs/Web/JavaScript/Memory_Management