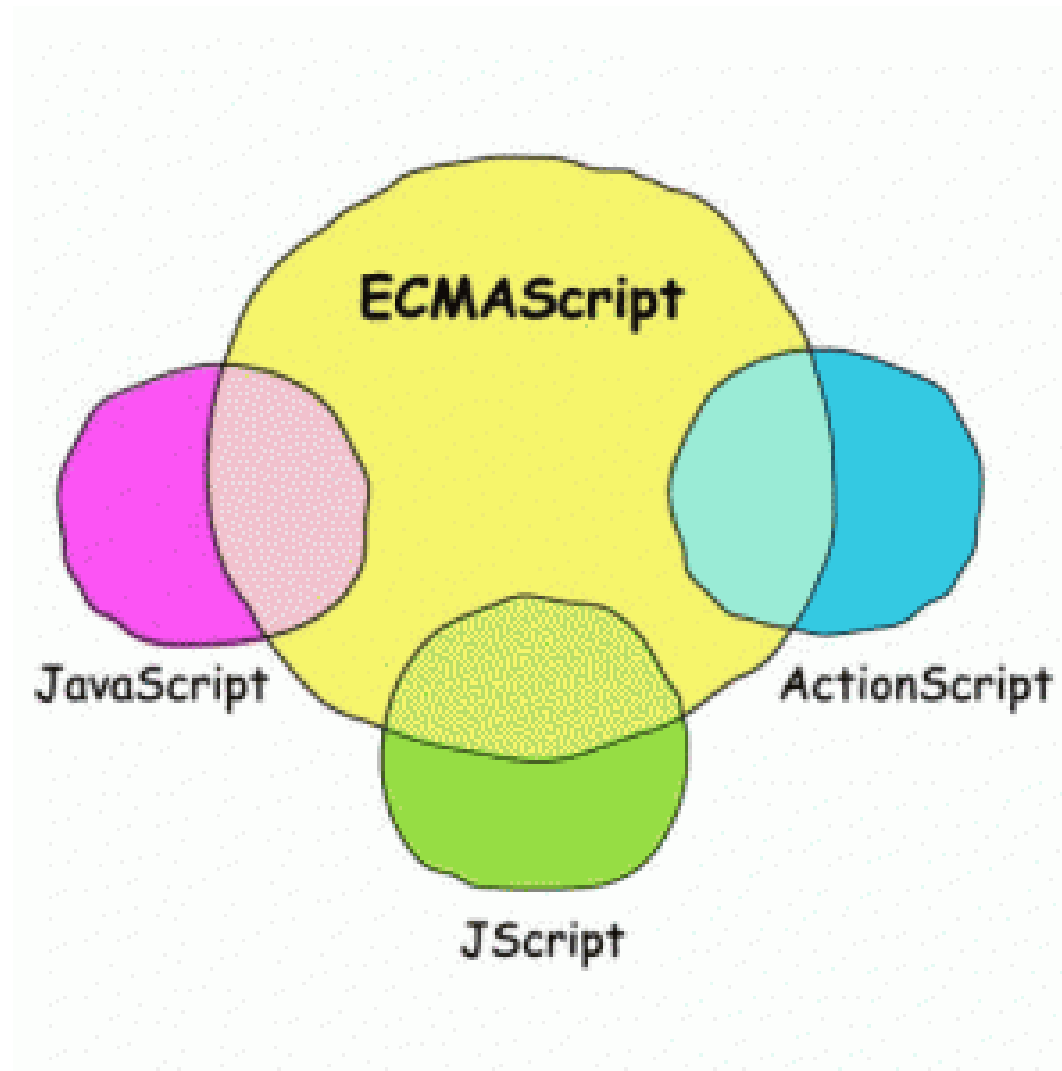


# Lenguajes y estándares en la Web

## Tema 2: Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software



# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- **Introducción**
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Introducción: ¿Qué se puede hacer?

Grado en  
Ingeniería  
Informática  
del Software

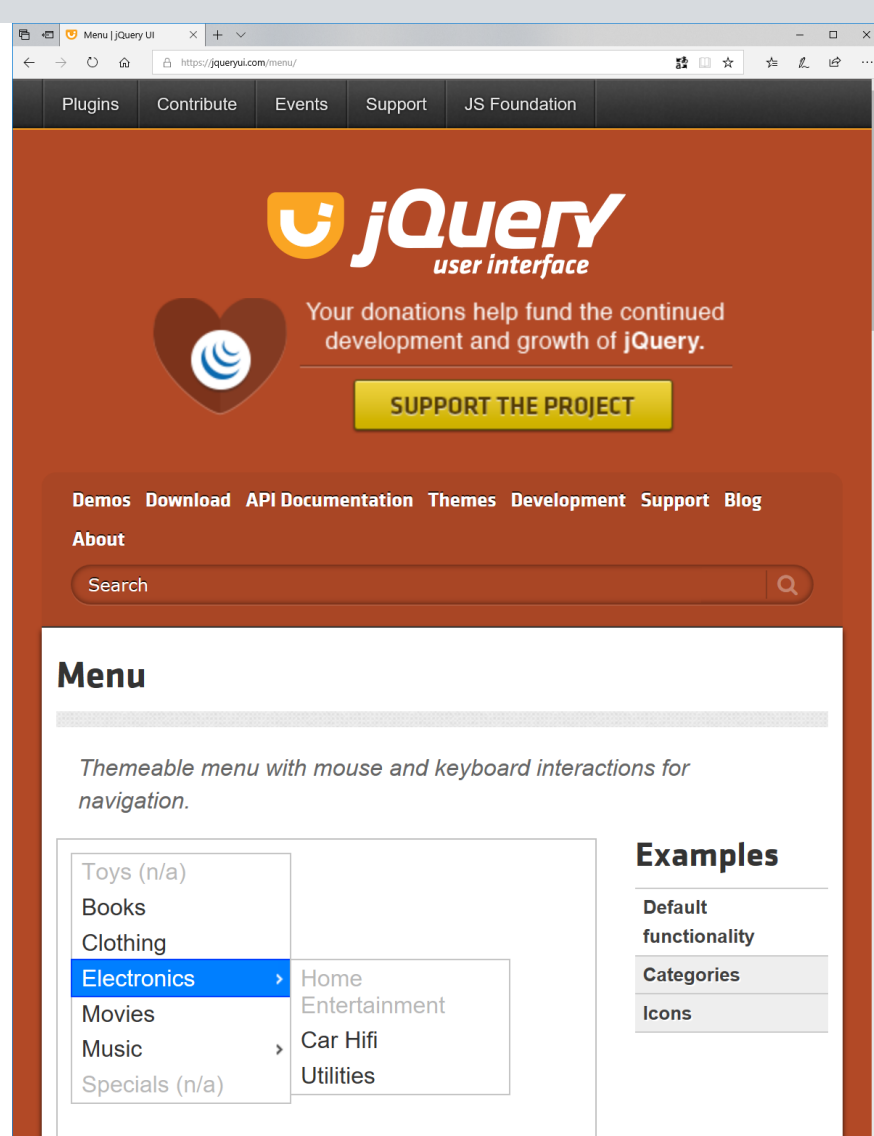
- **¿Qué se puede hacer con los lenguajes de Script?**
  - Interfaces de usuario interactivos: menús y botones
  - Validar y enviar formularios
  - Realizar cálculos (simples y complejos)
  - Personalización de las páginas Web
  - Hacer interactiva una página Web
  - Inclusión de datos del sistema cliente (hora, fecha, etc.). Crear relojes animados, calendarios, etc.
  - Manipular los elementos HTML de una página
  - Juegos, calculadoras, etc.
  - Incluir mapas y situación (geo-posicionamiento)
  - Consumir Servicios Web

# Lenguajes y estándares en la Web

## Introducción. Ejemplo: Menús

Grado en  
Ingeniería  
Informática  
del Software

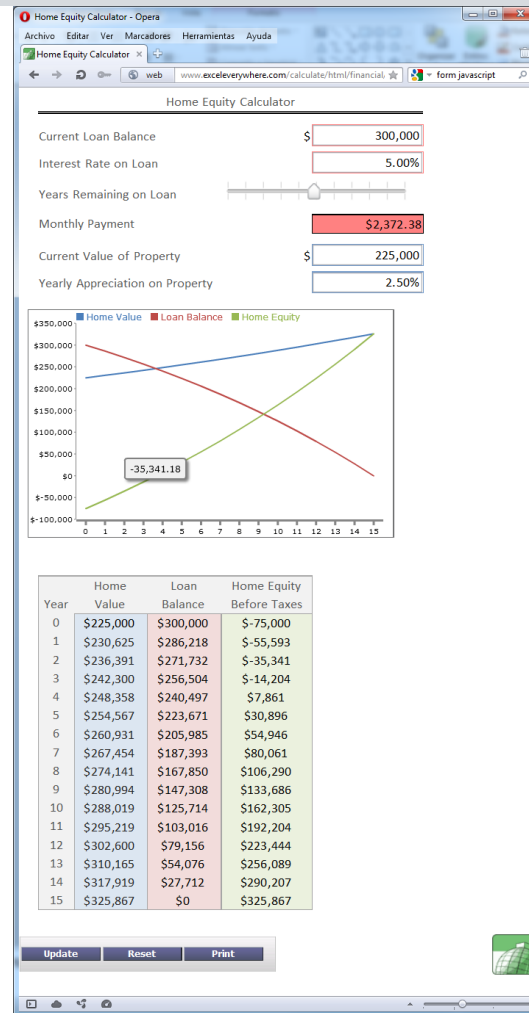
<https://jqueryui.com/menu/>



# Lenguajes y estándares en la Web

## Introducción. Ejemplo: Cálculos, gráficos e informes

Grado en  
Ingeniería  
Informática  
del Software



[http://www.exceleverywhere.com/calculate/html/financial/Home Loan Calculator/Home Loan Calculator.htm](http://www.exceleverywhere.com/calculate/html/financial/Home%20Loan%20Calculator/Home%20Loan%20Calculator.htm)

# Lenguajes y estándares en la Web

## Introducción. Ejemplos: Puzzle

Grado en  
Ingeniería  
Informática  
del Software



<https://javascript.plainenglish.io/build-an-8-puzzle-game-with-pure-javascript-efe424bc252a>

# Lenguajes y estándares en la Web

## Introducción. Ejemplo: Calculadora HP-35

Grado en  
Ingeniería  
Informática  
del Software

### HP-35 RPN Calculator



*Click the image to launch it.*

This JavaScript simulator is a very faithful simulation with the exception of some rounding errors. Both the HP-35 and the JavaScript produce rounding errors, but they do so in different situations. It was written by [Neil Fraser](#). If you are using Internet Explorer you may find that the display sometimes updates quite slowly across the network. You may want to [download](#) your own copy. The ZIP file has all the code and graphics (50KB).

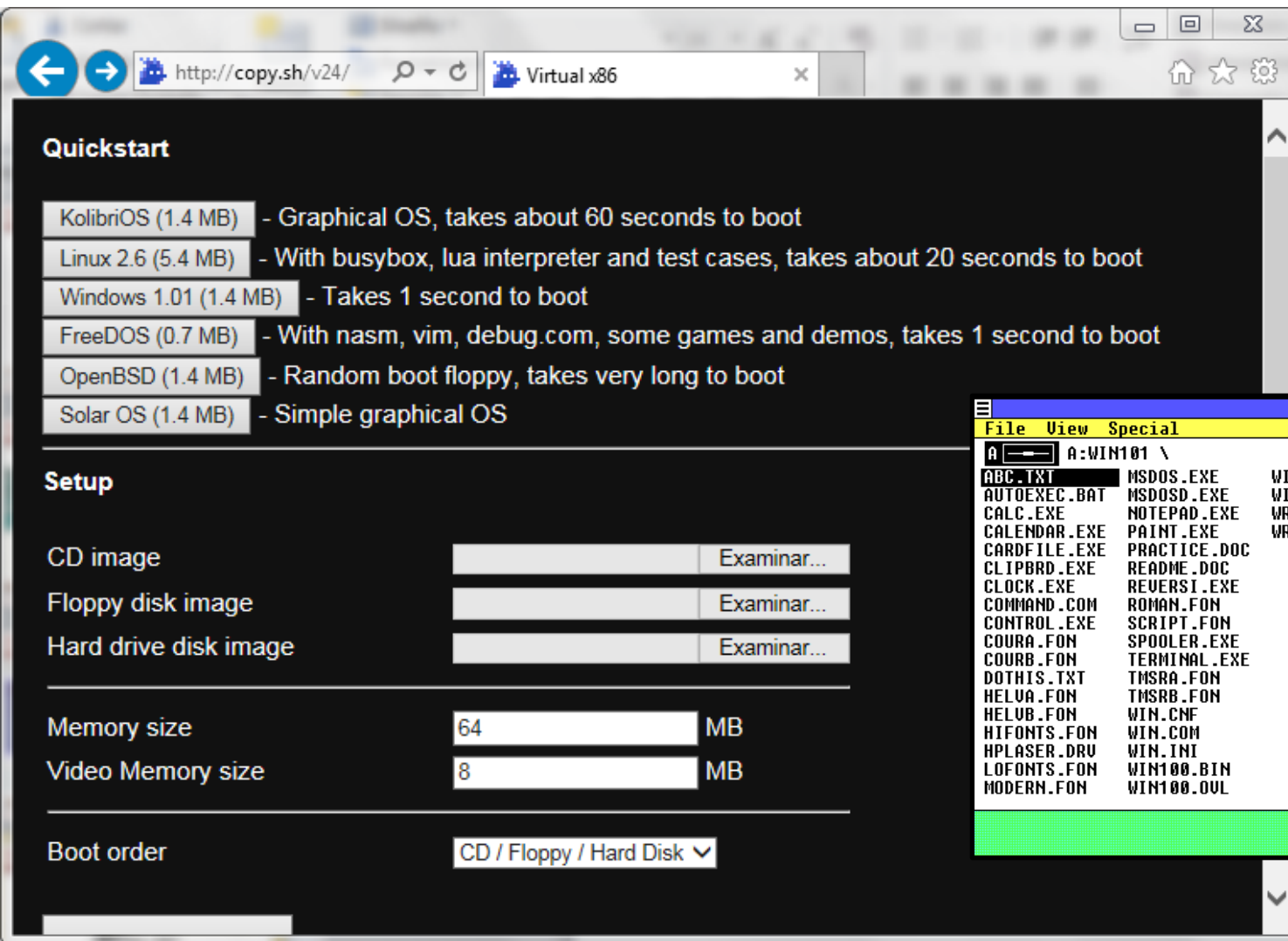
<http://www.hpmuseum.org/simulate/hp35sim/hp35sim.htm>



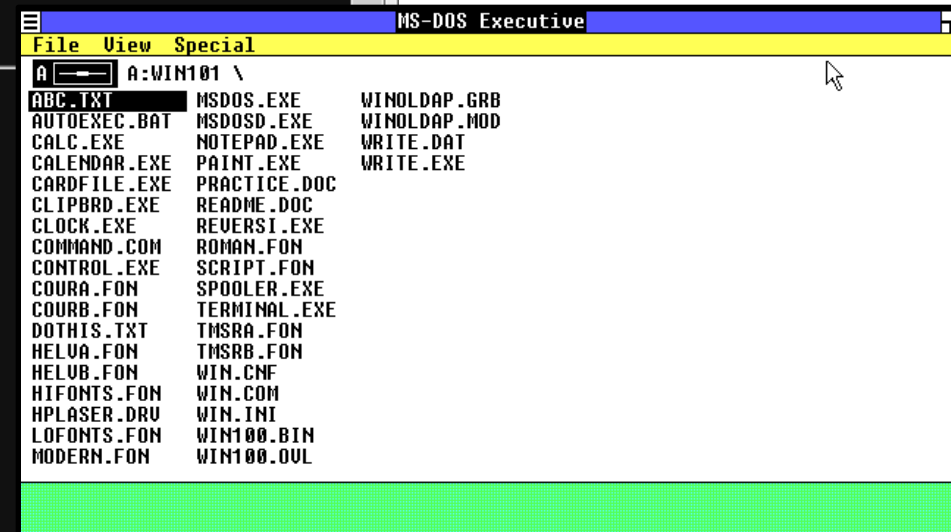
# Lenguajes y estándares en la Web

## Introducción. Ejemplo: Emuladores de sistemas operativos

Grado en  
Ingeniería  
Informática  
del Software



<http://copy.sh/v24/>

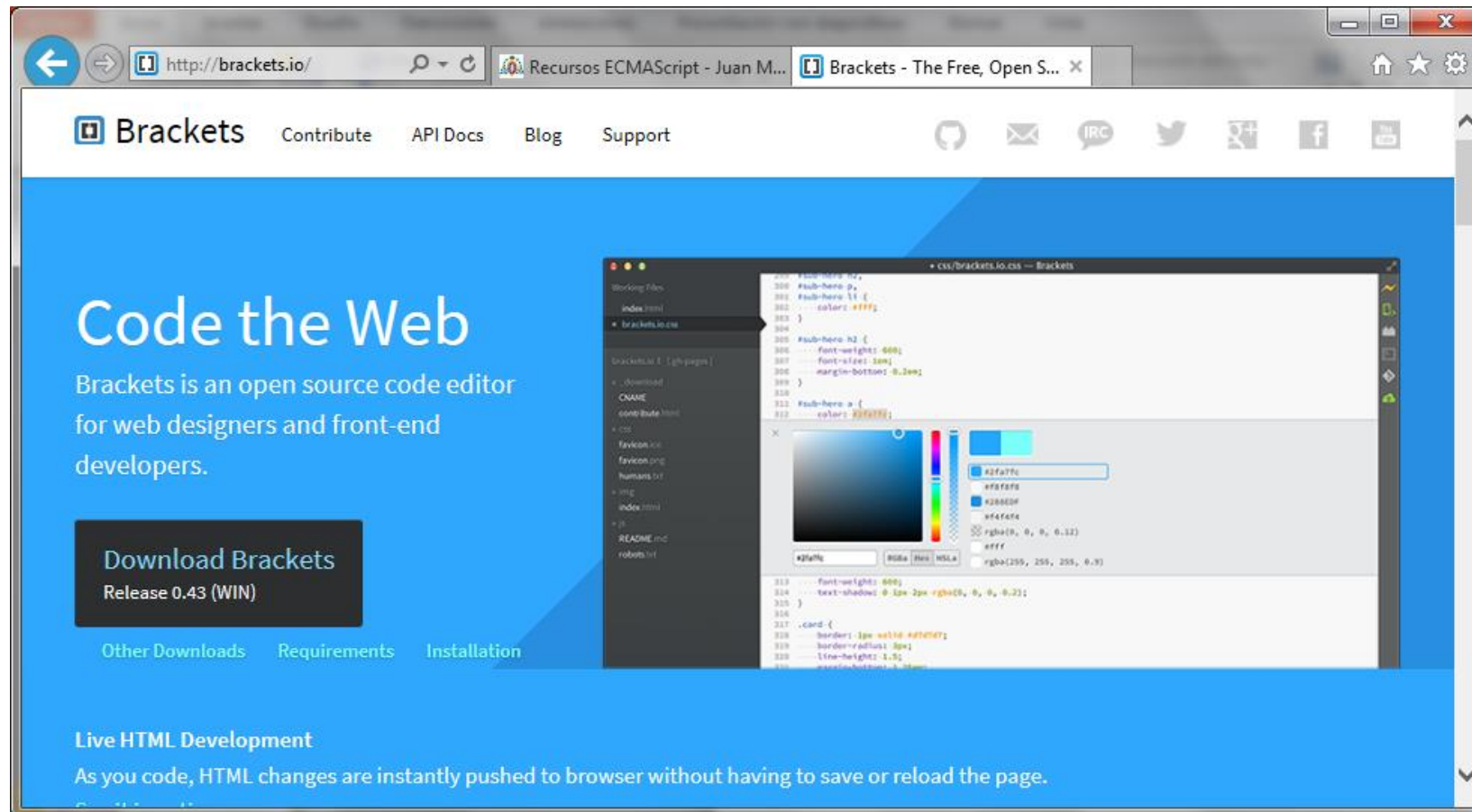


# Lenguajes y estándares en la Web

## Introducción. Ejemplo: Editores

<http://brackets.io/>

Grado en  
Ingeniería  
Informática  
del Software



# Lenguajes y estándares en la Web

## Introducción. Ejemplo: Consumo de Servicios Web

Grado en  
Ingeniería  
Informática  
del Software

The screenshot shows a web browser window with the address bar displaying `di002.edv.uniovi.es/~cueva/JavaScript/87-jQuery-AJAX-XML-meteo.html`. The page title is "jQuery: Usa AJAX para obtener un XML con los datos de la estación meteorológica de Oviedo usando [OpenWeatherMap](#)". Below the title is a button labeled "Obtener XML". The main content area displays the following text:

Datos en XML desde [OpenWeatherMap](#)  
;Todo correcto! XML recibido de [OpenWeatherMap](#)

XML

```
<current><city id="3114711" name="Oviedo"><coord lat="43.36" lon="-5.84" /><country>ES</country><sun set="2018-11-05T17:09:07" rise="2018-11-05T07:04:12" /></city><temperature unit="metric" max="10" min="10" value="10" /><humidity unit="%" value="87" /><pressure unit="hPa" value="1000" /><wind><speed name="Fresh Breeze" value="8.2" /><direction name="Southwest" value="220" code="SW" /></wind><clouds name="broken clouds" value="75" /><visibility value="6000" /><precipitation mode="no" /><weather value="lluvia ligera" icon="10d" number="500" /></lastupdate value="2018-11-05T13:30:00" /></current>
```

Datos

- Número de elementos del XML: 17
- Ciudad: Oviedo
- Longitud: -5.84 grados
- Latitud: 43.36 grados
- País: ES
- Amanece a las: 8:04:12
- Oscurece a las: 18:09:07
- Temperatura: 10 grados Celsius
- Temperatura mínima: 10 grados Celsius
- Temperatura máxima: 10 grados Celsius
- Temperatura (unidades): metric
- Humedad: 87 %
- Presión: 1000 hPa
- Velocidad del viento: 8.2 metros/segundo
- Nombre del viento: Fresh Breeze
- Dirección del viento: 220 grados
- Código del viento: SW
- Nombre del viento: Southwest
- Nubosidad: 75
- Nombre nubosidad: broken clouds
- Visibilidad: 6000 metros
- Precipitación valor: undefined
- Precipitación modo: no
- Descripción: lluvia ligera
- Hora de la medida: 14:30:00
- Fecha de la medida: 05/11/2018

<http://di002.edv.uniovi.es/~cueva/JavaScript/87-jQuery-AJAX-XML-meteo.html>

# Lenguajes y estándares en la Web

## Introducción. ¿Cómo funciona?

Grado en  
Ingeniería  
Informática  
del Software

- ¿Cómo funcionan los lenguajes de Script?
  - El navegador o agente de usuario es la herramienta clave
  - El navegador interpreta y ejecuta el código fuente incrustado en el HTML
    - Cuando el navegador solicita al servidor Web una página recibe tanto el código HTML como los scripts incluidos en ese código
    - El navegador **lee la página de inicio a final** mostrando los contenidos HTML y **ejecutando los scripts en el orden en que aparecen** (salvo que existan llamadas a funciones)
    - Aunque se ejecute códigos separados en distintas partes del HTML, el navegador los entiende como un código único que comparte la memoria en tiempo de ejecución..
- Siempre trataremos de **separar al máximo la computación (JS) de la presentación (HTML y CSS)**
  - Los archivos .js estarán separados y enlazados desde de los .html

<http://di002.edv.uniovi.es/~cueva/JavaScript/73-Asignatura.html>

<http://di002.edv.uniovi.es/~cueva/JavaScript/74-InfoNavegador.html>

# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- **Evolución de los lenguajes de Script**
- El estándar ECMAScript
- Implementaciones ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Evolución de los lenguajes de Script (I)

Grado en  
Ingeniería  
Informática  
del Software

- En 1995 nace *JavaScript*
  - Fue desarrollado originalmente por *Brendan Eich* de **Netscape** con el nombre de *Mocha*, el cuál fue renombrado posteriormente a *LiveScript*, para finalmente quedar como **JavaScript**.
  - El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador Web *Netscape Navigator* en la versión 2.0B3
  - La denominación produjo **confusión**, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una **estrategia de mercadotecnia de Netscape**.
- Hoy *JavaScript* es una **marca registrada de Oracle Corporation**. Es usada con licencia por los productos creados por Netscape Communications y entidades actuales como la **Fundación Mozilla**
  - <https://developer.mozilla.org/es/docs/Web/JavaScript>

# Lenguajes y estándares en la Web

## Evolución de los lenguajes de Script (II)

Grado en  
Ingeniería  
Informática  
del Software

- En **1996** se incorporan al navegador **Internet Explorer (IE)** versión 3.0 los lenguajes de Script: **VBScript** y **JScript**
- **VBScript**
  - Es una adaptación del lenguaje de programación **Visual Basic**
  - Cuando se utiliza en el entorno IE tiene una funcionalidad similar a **JScript**, pero su sintaxis es similar al lenguaje de programación Visual Basic
  - Es parte fundamental de la ejecución de aplicaciones de servidor Web programadas en **ASP** (*Active Server Pages*).
  - Microsoft ha intentado competir mediante esta tecnología también en entornos de cliente Web pero con poco éxito.
  - **VBScript solamente está disponible en el navegador IE**
- **JScript**
  - Basado en **JavaScript** de *Netscape* pero cambiando el nombre y algunas características para evitar violar los derechos de autor.
  - Actualmente el navegador **Edge no soporta JScript**.
  - Se inició con la versión 1.0 con Internet Explorer 3.0 en Agosto 1996
  - La última **versión 11.0** con **IE 11.0** en **Noviembre 2013**.
  - La última versión está basada en **ECMA-262 “5<sup>th</sup> edition”**.
  - La última versión se corresponde con la versión 1.8.1 de **JavaScript**
  - Hay un dialecto basado en **ECMA-262 “3<sup>th</sup> edition”** denominado **JScript 10.0** también conocido como JScript.NET para su uso con ASP.NET. Puede compilarse con **.NET Framework 4.0**



# Lenguajes y estándares en la Web

## Evolución de los lenguajes de Script (III)

Grado en  
Ingeniería  
Informática  
del Software

- JScript

Version	Date	Introduced with <sup>[7]</sup>	Based on <sup>[note 1]</sup>	Similar JavaScript version
1.0	Aug 1996	Internet Explorer 3.0	Netscape JavaScript	1.0
2.0	Jan 1997	Windows IIS 3.0	Netscape JavaScript	1.1
3.0	Oct 1997	Internet Explorer 4.0	ECMA-262 1st edition <sup>[note 2]</sup>	1.3
4.0		Visual Studio 6.0 (as part of Visual InterDev)	ECMA-262 1st edition	1.3
5.0	Mar 1999	Internet Explorer 5.0	ECMA-262 2nd edition	1.4
5.1		Internet Explorer 5.01	ECMA-262 2nd edition	1.4
5.5	Jul 2000	Internet Explorer 5.5 & Windows CE 4.2	ECMA-262 3rd edition	1.5
5.6	Oct 2001	Internet Explorer 6.0 & Windows CE 5.0	ECMA-262 3rd edition	1.5
5.7	Nov 2006	Internet Explorer 7.0	ECMA-262 3rd edition + ECMA-327 (ES-CP) <sup>[note 3]</sup>	1.5
5.8	Mar 2009	Internet Explorer 8.0 & Internet Explorer Mobile 6.0	ECMA-262 3rd edition + ECMA-327 (ES-CP) + JSON (RFC 4627) <sup>3</sup>	1.5
9.0	Mar 2011	Internet Explorer 9.0	ECMA-262 5th edition	1.8.1

- JScript.NET

Version	Platform	Date	Introduced with	Based on
7.0	Desktop CLR 1.0	2002-01-05	Microsoft .NET Framework 1.0	ECMA-262 3rd edition <sup>[note 4]</sup>
7.1	Desktop CLR 1.1	2003-04-01	Microsoft .NET Framework 1.1	ECMA-262 3rd edition <sup>[note 4]</sup>
8.0	Desktop CLR 2.0	2005-11-07	Microsoft .NET Framework 2.0	ECMA-262 3rd edition <sup>[note 4]</sup>
10.0	Desktop CLR 4.0	2010-08-03	Microsoft .NET Framework 4.0	ECMA-262 3rd edition <sup>[note 4]</sup>

<https://en.wikipedia.org/wiki/JScript>



# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- **El estándar ECMAScript**
- Implementaciones ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## El estándar ECMAScript (I)

Grado en  
Ingeniería  
Informática  
del Software

- **ECMAScript** es el nombre del lenguaje de Script definido en el estándar ECMA-262 por “ECMA International” y adoptado como norma **ISO/IEC 16262**.
- Esta norma se comenzó a definir en noviembre de 1996, provocado por las **diferencias en el manejo de fechas** entre *JavaScript* y *JScript*.
- La **primera versión** fue aprobada en junio de **1997**
- **El nombre es un acuerdo** entre las principales empresas implicadas (Netscape y Microsoft).
- La documentación del estándar puede encontrarse en
  - ❑ [ECMA-262 - Ecma International \(ecma-international.org\)](http://ecma-international.org)

# ECMA-262

ECMAScript® 2024 language specification

15th edition, June 2024

- Última especificación del lenguaje:
  - **15ª edición**
  - **Junio de 2024**

<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>

# Lenguajes y estándares en la Web

## El estándar ECMAScript (III)

Grado en  
Ingeniería  
Informática  
del Software

- *JavaScript* y *JScript* se definen a si mismos como compatibles con el **estándar ECMAScript**
- Sin embargo *JavaScript* y *JScript* tienen **características adicionales** no descritas en el estándar ECMA-262.
- Hay navegadores como “Opera” que incluyen extensiones de ambos lenguajes (*JavaScript* y *JScript*)
- Se aconseja **usar las características del estándar** ECMA-262 y no lo específico de cada navegador para que **los sitios Web diseñados puedan ser visualizados con cualquier navegador**.

# Lenguajes y estándares en la Web

## El estándar ECMAScript (IV)

Grado en  
Ingeniería  
Informática  
del Software

- Versiones de ECMAScript

- Versión 1.0 - Junio 1997. Primera edición
- Versión 2.0 - Junio 1998. Pequeños cambios para adaptarse al estándar ISO/IEC 16262.
- Versión 3.0 - Diciembre 1999. Se añaden: expresiones regulares, mejora del manejo de “string”, nuevas instrucciones de control de flujo, manejo de excepciones con try/catch, se incluyen definiciones de errores más estrictas, se añaden capacidades de formateo de presentación de números y algunas extensiones más.
- Versión 4.0 - Desechada por diferencias entre los ponentes sobre el diseño de las nuevas tendencias del lenguaje.
- Versión 5.0 - Diciembre 2009. Clarifica muchas ambigüedades de la versión 3 y se acomoda a algunas implementaciones usadas comúnmente por los navegadores y que eran diferentes de la especificación de la versión 3. Añade algunas nuevas características como: “getters” y “setters”, biblioteca de soporte de JSON y una definición más completa de las propiedades de reflexión de los objetos.
- Versión 5.1. Junio 2011. Se añade el “modo estricto” para mejorar el control de errores. Véase [http://msdn.microsoft.com/en-us/library/br230269\(v=vs.94\).aspx](http://msdn.microsoft.com/en-us/library/br230269(v=vs.94).aspx)
- Versión 6. Junio 2015. La sexta edición, conocida como ES6 o ECMAScript 2015, añade una nueva sintaxis para escribir aplicaciones complejas, incluyendo las clases y módulos. Otras nuevas características incluyen iteradores y las expresiones generadoras, las funciones de dirección, datos binarios, matrices con tipo, nuevas colecciones ...
- Versión 7. Junio 2016. Continúa las reformas de ES6, con el control de los efectos laterales. Incluye dos nuevas características: el operador de exponenciación (\*\*) y *Array.prototype.includes*.

# Lenguajes y estándares en la Web

## El estándar ECMAScript (V)

Grado en  
Ingeniería  
Informática  
del Software

- **Versiones de ECMAScript**
  - **Versión 8.0** - Junio 2017.
    - Las nuevas funcionalidades propuestas incluyen concurrencia
    - Nuevos elementos
    - Transferencia de datos binarios
    - Nuevos tipos numéricos y más funciones matemáticas
    - Nueva especificación sintáctica en procesos concurrentes (estados de espera / estados asíncronos)
    - Mejora la Meta-programación con meta-clases, clases y propiedades de instancia, sobrecarga de operadores, etc.
  - **Versión 9.0** – Junio 2018
    - Propiedades rest/spread
    - Iteración asíncrona
    - Nuevas expresiones regulares
    - `prototype.finally()`
  - **Versión 10.0** – Junio 2019
    - `Array.flat()`
    - `Array.flatMap()`
    - `String.trimStart()` & `String.trimEnd()`
    - Se permite usar `try/catch` sin el parámetro `error` dentro del bloque `catch`.
    - `Object.fromEntries()`
  - **Versión 11.0** – Junio 2020
    - Trabaja con módulos
    - Permite importar y exportar
    - Nuevos tipos de datos y objetos (por ejemplo: enteros de precisión arbitraria)
    - El método `matchAll()` para expresiones regulares
    - El objeto `globalThis`
    - Nuevos operadores

# Lenguajes y estándares en la Web

## El estándar ECMAScript (VI)

Grado en  
Ingeniería  
Informática  
del Software

- Versiones de ECMAScript

- Versión 12.0 - Junio 2021

- String.replaceAll()
      - Nuevo método de los objetos *String*
      - Reemplaza un sub-String en todas las instancias de una constante String
    - Métodos y atributos privados
      - Se utiliza el **carácter #** delante del método o del atributo para indicar que es privado
    - Promise.any()
      - El método acepta una lista de objetos Promise como un objeto iterable. Tan pronto como se cumple una Promesa de la lista, devuelve la Promesa cumplida.
      - Si no se cumplen las promesas en el iterable, se produce un error.
    - Separadores numéricos
      - Ejemplo

```
let presupuesto = 1_000_000_000_000;  
// ¿Cuánto es el Presupuesto? ¡Es un trillón!  
//  
// Vamos a confirmarlo  
console.log(presupuesto === 10 ** 12); // true
```
    - Operadores de asignación lógica con &&, || o ??
      - `x &&= y;`
      - `x ||= y;`
      - `x ??= y;`

- Versiones de ECMAScript

- Versión 13.0 - Junio 2022

- <https://dev.to/brayanarrieta/new-javascript-features-ecmascript-2022-with-examples-4nhg>
    - Espera (await) de nivel superior
    - Campos, métodos y accesorios de instancia privada
    - Campos y métodos de clase estática
    - Bloques de inicialización de clases estáticas
    - Error: .cause
    - Array, String y TypedArray: Método .at()
    - Objeto: .hasOwn()
    - RegExp: match .indices (bandera "d")



# Lenguajes y estándares en la Web

## El estándar ECMAScript (VIII)

Grado en  
Ingeniería  
Informática  
del Software

- Versiones de ECMAScript (ES14)
  - Versión 14.0 - Junio 2023- principales novedades
    - <https://www.infoworld.com/article/3703571/all-the-new-features-in-ecmascript-2023-es14.html>
    - **Array.prototype.toSorted**  

```
let arr = [5,4,2,3,1]  
arr === arr.sort(); // true - [1, 2, 3, 4, 5]  
arr === arr.toSorted(); // false - [1, 2, 3, 4, 5]
```
    - **Array.prototype.toReversed**
    - **Array.prototype.with**
    - **Array.prototype.findLast**
    - **Array.prototype.findLastIndex**
    - **Array.prototype.toSpliced**
    - **Los símbolos como claves en colecciones débiles (WeakRef).**
      - Las colecciones débiles son un poco esotéricas en comparación con el uso diario de JavaScript. En programación, una referencia débil es aquella que se eliminará si entra en funcionamiento el recolector de basura. Dicho de otra manera, una referencia débil por sí sola no es suficiente para evitar que el objeto referenciado sea eliminado por el algoritmo de recolección de basura (por eso es una referencia débil).
      - [WeakRef - JavaScript | MDN \(mozilla.org\)](#)

# Lenguajes y estándares en la Web

## El estándar ECMAScript (IX)

Grado en  
Ingeniería  
Informática  
del Software

- Versiones de ECMAScript (ES15)
  - Versión 15.0 - Junio 2024- principales novedades



<https://medium.com/@yourfuse/javascript-whats-new-with-ecmascript-2024-es15-ef056d2f4bf1>

# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- **Implementaciones de ECMAScript**
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Implementaciones de ECMAScript (I)

Grado en  
Ingeniería  
Informática  
del Software

- ECMAScript es compatible con la mayoría de los navegadores Web
- Las implementaciones de los navegadores incluyen **extensiones** al lenguaje no incluidas en el estándar ECMAScript
- Esto significa que las aplicaciones que usan extensiones específicas de un navegador pueden ser **incompatibles** cuando se ejecutan en otro navegador
- Se aconseja el uso de las **características comunes** del estándar ECMAScript.
- Actualmente Google y su navegador Chrome lidera la mayor parte de las iniciativas del estándar ECMAScript

# Lenguajes y estándares en la Web

## Implementaciones de ECMAScript (II)

Grado en  
Ingeniería  
Informática  
del Software

Implementation	Applications	ECMAScript edition
SpiderMonkey	Firefox, the Gecko layout engine, Adobe Acrobat <sup>[d 1]</sup>	2017 <sup>[37][d 2]</sup>
V8	Google Chrome, Node.js, Opera, MarkLogic. <sup>[38]</sup>	2017 <sup>[37][d 3]</sup>
JavaScriptCore (Nitro)	WebKit, Safari, Qt 5	2017 <sup>[39]</sup>
Chakra	Microsoft Edge	5.1 and features from 2015, <sup>[d 4]</sup> 2016 and 2017 <sup>[37]</sup>
JerryScript	Resource constrained IoT devices, Pebble	5.1 <sup>[40]</sup>
JScrip 9.0	Internet Explorer, the Trident layout engine	5.1 <sup>[41]</sup>
Nashorn	Java 9	6 <sup>[42]</sup>
Nashorn	Java	5.1 <sup>[43]</sup>
Rhino	Java Platform, Standard Edition	3
Carakan (deprecated)	Opera 12	5.1 <sup>[44][d 5]</sup>
KJS	KHTML	5.1 and features from 2015 <sup>[citation needed]</sup>
Ejscrip	Appweb Web Server, Samba 4	2015 <sup>[45]</sup>
JScrip .NET	Microsoft .NET Framework	3 <sup>[d 6]</sup>
ActionScript	Adobe Flash, Adobe Flex, Adobe AIR	4 <sup>[46]</sup>
ExtendScript	Adobe Creative Suite products: InDesign, Illustrator, Photoshop, Bridge, After Effects, Premiere Pro	3 <sup>[citation needed]</sup>
InScript	iCab	3
Max/MSP engine	Max	3
QtScript (deprecated)	KDE SC 4	3
Caja		5 <sup>[47]</sup>

<http://en.wikipedia.org/wiki/ECMAScript>

# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

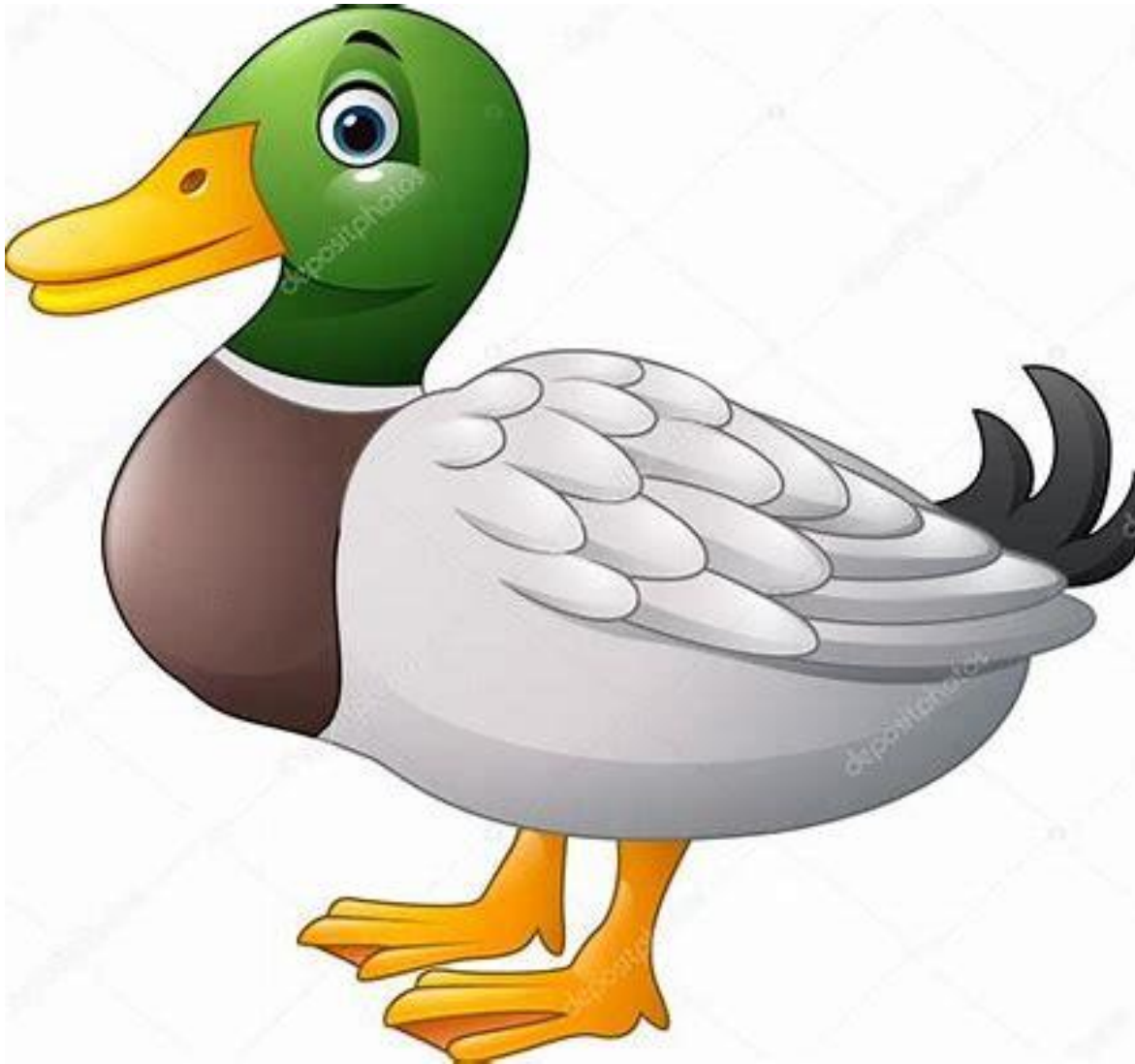
Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones de ECMAScript
- **El lenguaje ECMAScript**
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

El lenguaje ECMAScript: “un pato creado por un comité”

Grado en  
Ingeniería  
Informática  
del Software



# Lenguajes y estándares en la Web

## El lenguaje ECMAScript: Paradigma procedimental

Grado en  
Ingeniería  
Informática  
del Software

- **Lenguaje imperativo y estructurado. Paradigma procedimental**
  - Es un lenguaje imperativo y estructurado con sintaxis similar al lenguaje C
    - Sin embargo, no soporta el ámbito (“scope”) entre bloques de llaves (“{” y “}”) de las variables que soporta el lenguaje C
    - El ámbito de las variables y objetos es local en el interior de las funciones siempre que se declaren con la palabra reservada “**var**”.
    - El dialecto JavaScript 1.7 y siguientes versiones soporta el ámbito entre bloques usando la palabra reservada “**let**”.
    - Como el lenguaje C diferencia entre *expresiones* y *sentencias*.
    - A diferencia del lenguaje C no es obligatoria la colocación del punto y coma (“;”) para terminar las sentencias. Es optativo: se puede colocar el punto y coma o no.
    - Otra diferencia con el lenguaje C es el uso de variables locales y globales con la palabra reservada “**var**”
      - » En el interior de una función hace que las variables sean locales cuando llevan “**var**” o “**let**” en su declaración.
      - » Fuera de una función las variables son globales (lleven o no “var”)
      - » **En el interior de una función las variables que no llevan “var” son globales cuando se termina de ejecutar la función.**



# Lenguajes y estándares en la Web

## El lenguaje ECMAScript: Paradigma procedimental

Grado en  
Ingeniería  
Informática  
del Software

- **Lenguaje imperativo y estructurado. Paradigma procedimental**
  - Es un lenguaje imperativo y estructurado con sintaxis similar al lenguaje C
    - Sin embargo no soporta el ámbito (“scope”) entre bloques de llaves (“{” y “}”) de las variables que soporta el lenguaje C
    - El ámbito de las variables y objetos es local en el interior de las funciones siempre que se declaren con la palabra reservada **var**
    - El dialecto JavaScript 1.7 y siguientes versiones soporta el ámbito entre bloques usando la palabra reservada **let**
    - Como el lenguaje C permite expresiones y sentencias.
    - A diferencia del lenguaje C es obligatoria la colocación del punto y coma (“;”) para terminar las sentencias. Es optativo: se puede colocar el punto y coma o no.
    - Una diferencia con el lenguaje C es el uso de variables locales y globales con la palabra reservada **var**
      - » En el interior de una función hace que las variables sean locales cuando llevan **“var”** o **“let”** en su declaración.
      - » Fuera de una función las variables son globales (lleven o no “var”)
      - » **En el interior de una función las variables que no llevan “var” son globales cuando se termina de ejecutar la función.**

# Cronómetro

00:00:00:0

Arrancar

Stop

Reset

Elige la acción sobre el cronómetro



Prohibido

# Lenguajes y estándares en la Web

## El lenguaje ECMAScript: Paradigma procedimental

<http://di002.edv.uniovi.es/~cueva/JavaScript/93-Cronometro.html>

Grado en  
Ingeniería  
Informática  
del Software

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Cronómetro</title>
</head>
<body>
  <h1>Cronómetro</h1>
<h2 id="pantalla">00:00:00:0</h2>
<main id="contenedorBotones">
  <button id="arrancar" onclick="arrancar();">Arrancar</button>
  <button id="stop" onclick="parar();">Stop</button>
  <button id="reset" onclick="reset();">Reset</button>
  <p>Elige la acción sobre el cronómetro</p>
</main>
```

Prohibido

# Lenguajes y estándares en la Web

## El lenguaje ECMAScript: Paradigma procedimental

Grado en  
Ingeniería  
Informática  
del Software

```
<script>
//Version 1.1 23/10/2021
var hora = 0, minuto = 0, segundo = 0, decima = 0;
var pantalla = "<p>";
var stop = true;
function arrancar() {
    if(stop == true) {
        stop = false;
        cronometro();
    }
}
function cronometro() {
    if(stop == false) {
        decima++;
        if(decima > 9) {
            decima = 0;
            segundo++;
        }
        if(segundo > 59) {
            segundo = 0;
            minuto++;
        }
        if(minuto > 59) {
            minuto = 0;
            hora++;
        }
        mostrar();
        window.setTimeout("cronometro()", 100);
    }
}
```

Prohibido

# Lenguajes y estándares en la Web

## El lenguaje ECMAScript: Paradigma procedimental

```
function mostrar() {
    if(hora < 10) pantalla = "0" + hora + ":";
    else pantalla = hora + ":";
    if(minuto < 10) pantalla += "0";
    pantalla += minuto + ":";
    if(segundo < 10) pantalla += "0";
    pantalla += segundo + ":" + decima;
    document.getElementById("pantalla").innerHTML = pantalla;
}

function parar() {
    stop = true;
}

function reset() {
    if(stop == false) {
        stop = true;
    }
    hora = minuto = segundo = decima = 0;
    pantalla = "";
    mostrar();
}

</script>
<footer>
    <a href="http://validator.w3.org/check/referer"> </a>
</footer>
</body>
</html>
```

Prohibido

# Lenguajes y estándares en la Web

## El estándar ECMAScript: Lenguaje dinámico

Grado en  
Ingeniería  
Informática  
del Software

- **Lenguaje dinámico**
  - Tipos dinámicos (*“dynamic typing”*)
    - Los tipos de las variables no es necesario declararlos
    - El tipo se infiere dinámicamente por el valor que toma la variable
      - » Ejemplo: `var nombre = “Juan Manuel”`
      - » Se infiere que la variable `nombre` es de tipo `string`
    - Los tipos dinámicos son posibles debido a que la evaluación se realiza en tiempo de ejecución (*“run-time evaluation”*).
    - Las versiones 6 y 7 incorporan *“clases”* como Java o C++ cuya definición se realiza en tiempo de compilación.
    - Si tiene *objetos* que se crean en tiempo de ejecución o ya están predefinidos.
    - La versión 8 mejora la *meta-programación*
    - La versión 9 mejora los aspectos *asíncronos*
    - La versión 12 incorpora atributos y métodos *privados* en las clases

- **Lenguaje funcional**

- Las funciones son “*first-class*” permiten
  - Paso de funciones como argumentos a otras funciones
  - Ser devueltas por otras funciones
  - Ser manipuladas como cualquier objeto
- Funciones anidadas
  - Se permite construir funciones dentro de otras funciones
- Operador cierre (“*closure*”)
  - Incorpora el operador cierre “( )”

<http://di002.edv.uniovi.es/~cueva/JavaScript/95-jQuery-Cronometro.html>

### jQuery: Ejemplo de cronómetro

00:00:00:00

Arrancar





# Lenguajes y estándares en la Web

## El estándar ECMAScript: Paradigma funcional

```
// 95-jQuery-Cronometro.js
// Versión 1.0 28/11/2017 Juan Manuel Cueva Lovelle. Universidad de Oviedo
// Version 1.1 23/10/2021
$(document).ready(function(){
    var tiempo = {
        hora: 0,
        minuto: 0,
        segundo: 0,
        centesima:0,
        corriendo : 0
    };
    $("button").click(function(){
        if ( $(this).text() == 'Arrancar' ){
            $(this).text('Parar');
            tiempo.corriendo = setInterval(function(){
                // centesimas
                tiempo.centesima++;
                if(tiempo.centesima >= 100){
                    tiempo.centesima =0;
                    tiempo.segundo++;
                }
                // Segundos
                if(tiempo.segundo >= 60){
                    tiempo.segundo = 0;
                    tiempo.minuto++;
                }
                // Minutos
                if(tiempo.minuto >= 60){
                    tiempo.minuto = 0;
                    tiempo.hora++;
                }
                $("#hora").text(tiempo.hora < 10 ? '0' + tiempo.hora : tiempo.hora);
                $("#minuto").text(tiempo.minuto < 10 ? '0' + tiempo.minuto : tiempo.minuto);
                $("#segundo").text(tiempo.segundo < 10 ? '0' + tiempo.segundo : tiempo.segundo);
                $("#centesima").text(tiempo.centesima < 10 ? '0' + tiempo.centesima : tiempo.centesima);
            }, 10);
        }
        else
        {
            $(this).text('Stop');
            clearInterval(tiempo.corriendo);
        }
    });
});
```

Prohibido

**El paradigma funcional  
solamente se permite encapsulado  
en el interior de los métodos  
de las clases u objetos**

# Lenguajes y estándares en la Web

## El estándar ECMAScript: Paradigma orientado a objetos

Grado en  
Ingeniería  
Informática  
del Software

- **Lenguaje orientado a objetos basado en prototipos**
  - Es un lenguaje orientado a objetos
  - Los objetos nacen a partir de **clonar** otros objetos que son usados como prototipos.
  - Los objetos se clonan de otros objetos y pueden ampliarse sus atributos y métodos ampliando su funcionalidad.
  - Hay un objeto base denominado "**Object**"
  - Los constructores se construyen con funciones
  - Los métodos se construyen con funciones
  - El acceso a los campos y métodos miembros de un objeto puede hacerse con el operador punto (".")
  - En el acceso a los campos de un objeto también puede utilizarse el operador corchete ("["").
  - A partir de las **versiones 6 y 7** los objetos también pueden ser **instancias de clases**.
  - En la **versión 12** las clases pueden tener atributos y métodos privados antrponirndo el carácter #

### ECMAScript 6.0: Ejemplo de una clase Cronómetro y uso de bind

**Cronómetro : 00 : 00 : 00 : 0**

Arrancar

Parar

Reset

Elige la acción sobre el cronómetro



<http://di002.edv.uniovi.es/~cueva/JavaScript/94-EC6-Ejemplo-clase-Cronometro.html>

# Lenguajes y estándares en la Web

## El estándar ECMAScript: Paradigma orientado a objetos

Grado en  
Ingeniería  
Informática  
del Software

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8"/>
  <title>Clase Cronómetro</title>
</head>
<body>
  <h1>ECMAScript 6.0: Ejemplo de una clase Cronómetro y uso de bind</h1>
  <h2 id="pantalla">Pantalla del cronómetro</h2>

  <script>
    //Version 1.1 23/10/2021
    "use strict";
    class Cronómetro {
      constructor (nombre){
        this.nombre = nombre;
        this.decimas = 0;
        this.segundos = 0;
        this.minutos = 0;
        this.horas = 0;
        this.stop = true;
        this.mostrar();
      }
      reset(){
        this.stop = true;
        this.decimas = 0;
        this.segundos = 0;
        this.minutos = 0;
        this.horas = 0;
        this.mostrar();
        window.clearTimeout(this.timeoutID);
      }
      parar(){
        this.stop = true;
        this.mostrar();
      }
    }
  </script>
</body>
</html>
```

# Lenguajes y estándares en la Web

## El estándar ECMAScript: Paradigma orientado a objetos

Grado en  
Ingeniería  
Informática  
del Software

```
mostrar(){
    if (this.horas < 10) var stringHoras = "0" + this.horas;
    else stringHoras = this.horas;
    if (this.minutos < 10) var stringMinutos = "0" + this.minutos;
    else stringMinutos = this.minutos;
    if (this.segundos < 10) var stringSegundos = "0" + this.segundos;
    else stringSegundos = this.segundos;
    var stringCronometro = this.nombre + " : "
        + stringHoras + " : "
        + stringMinutos + " : "
        + stringSegundos + " : "
        + this.decimas;
    document.getElementById("pantalla").innerHTML = stringCronometro;
}
arrancar(){
    this.stop = false ;
    this.mostrar();
    this.reloj();
}
reloj(){
    if(this.stop == false) {
        this.decimas++;
        if(this.decimas > 9) {
            this.decimas = 0;
            this.segundos++;
        }
        if(this.segundos > 59) {
            this.segundos = 0;
            this.minutos++;
        }
        if(this.minutos > 59) {
            this.minutos = 0;
            this.horas++;
        }
    }
    this.mostrar();
    //bind() permite acceder a this de los objetos de la clase Cronómetro
    //Si no se usase bind() this es el objeto window
    this.timeoutID = window.setTimeout(this.reloj.bind(this),100);
}
```

# Lenguajes y estándares en la Web

## El estándar ECMAScript: Paradigma orientado a objetos

Grado en  
Ingeniería  
Informática  
del Software

```
var cronómetro = new Cronómetro("Cronómetro");
</script>

<main>
  <button onclick="cronómetro.arrancar()">Arrancar</button>
  <button onclick="cronómetro.parar()">Parar</button>
  <button onclick="cronómetro.reset()">Reset</button>
  <p>Elige la acción sobre el cronómetro</p>
</main>

<footer>
  <a href="http://validator.w3.org/check/referer"> </a>
</footer>
</body>
</html>
```

# Lenguajes y estándares en la Web

## El estándar ECMAScript: Objetos predefinidos

Grado en  
Ingeniería  
Informática  
del Software

- Otras características

- Tiene **objetos predefinidos** para manipular los elementos de HTML y del navegador
  - **Ejemplo:** Usa un objeto predefinido “*document*” que se corresponde con el documento HTML, e inserta un encabezado de nivel 1 usando el operador “+” de concatenación de “*string*”.

```
var nombre = "Juan Manuel Cueva Lovelle"  
document.write(nombre)  
document.write("<h1>" + nombre + "</h1>")
```

- Soporta “**expresiones regulares**” para la manipulación de “*string*” de una forma similar al lenguaje de programación **Perl**.



# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones de ECMAScript
- El lenguaje ECMAScript
- **Dialectos de ECMAScript**
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Dialectos de ECMAScript (I)

- Están ligados a cada navegador y a su “**motor de maquetado**”
  - El “*motor de maquetado*” es el componente software que toma el contenido marcado (como HTML, XML, archivos de imágenes, código fuente en un lenguaje de *script*, etc.) e información de formateo (como CSS, XSL, etc.) y luego muestra el contenido ya formateado en la pantalla.
  - Los “*motores de maquetado*” también denominados en lengua inglesa
    - “*Web browser engine*”, “*layout engine*”, “*rendering engine*”
  - El mismo “motor de maquetado” puede ser utilizado por distintos navegadores.
  - Listado de algunos “motores de maquetado” y algunos navegadores que los usan

Scripting engine ↕	Reference application(s) ↕	Conformance <sup>[35]</sup>			
		ES5 <sup>[36]</sup> ↕	ES6 <sup>[37]</sup> ↕	ES7 <sup>[38]</sup> ↕	Newer (2016+) <sup>[38][39]</sup> ↕
Chakra	Microsoft Edge	100%	96%	100%	54%
SpiderMonkey	Firefox	100%	98%	100%	77%
Chrome V8	Google Chrome, Opera	100%	98%	100%	93%
JavaScriptCore (Nitro)	Safari	97%	99%	100%	83%

# Lenguajes y estándares en la Web

## Dialectos de ECMAScript (II)

Grado en  
Ingeniería  
Informática  
del Software

Engine ↕	Status ↕	Steward ↕	License ↕
<b>Blink</b>	Active	Google	GNU LGPL, BSD-style
<b>EdgeHTML</b>	Active	Microsoft	Proprietary
<b>Gecko</b>	Active	Mozilla	Mozilla Public
<b>Goanna</b>	Active	M.C. Straver <sup>[1]</sup>	Mozilla Public
<b>Servo</b>	Active	Mozilla	Mozilla Public
<b>WebKit</b>	Active	Apple	GNU LGPL, BSD-style
<b>KHTML</b>	Discontinued	KDE	GNU LGPL
<b>Presto</b>	Discontinued	Opera Software	Proprietary
<b>Tasman</b>	Discontinued	Microsoft	Proprietary
<b>Trident</b>	Discontinued	Microsoft	Proprietary

[https://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_browser\\_engines](https://en.wikipedia.org/wiki/Comparison_of_web_browser_engines)

# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones de ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- **Extensiones de ECMAScript**
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Extensiones de ECMAScript (**OBSOLETAS**)

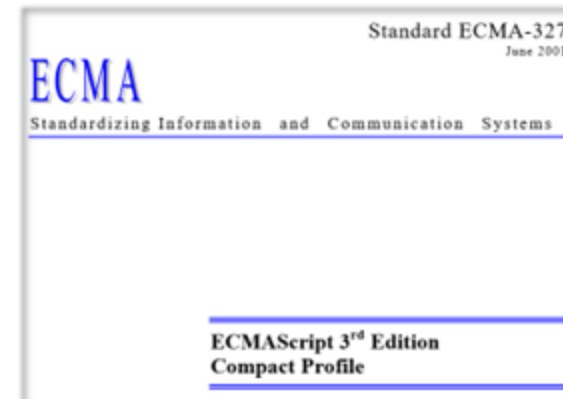
Grado en  
Ingeniería  
Informática  
del Software

- Extensiones y variantes de ECMAScript
  - **ECMAScript para XML**, es una extensión del estándar para añadir soporte a XML nativo
    - Estándar denominado ECMA-357
    - También se le denomina **E4X** (*ECMAScript for XML*)
    - **2ª Edición, Diciembre 2005**



<http://www.ecma-international.org/publications/files/ECMA-ST-WITHDRAWN/Ecma-357.pdf>

- **"Compact Profile"** para ECMAScript
  - Es una versión reducida para dispositivos con recursos limitados
  - Estándar denominado ECMA-327
  - También se le denomina **ES-CP**
  - **3ª Edición, Junio de 2001**



<http://www.ecma-international.org/publications/standards/Ecma-327.htm>

# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones de ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- **Preprocesadores de ECMAScript**
- WebAssembly
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Preprocesadores de ECMAScript

Grado en  
Ingeniería  
Informática  
del Software

- Son lenguajes de programación que una vez compilados generan código ECMAScript
- Ejemplos:
  - **TypeScript**
  - CoffeeScript
  - Dart (<https://dart.dev>)
  - Elm
  - Opa
  - Amber Smalltalk
  - Nim
  - Axe

# Lenguajes y estándares en la Web

## Preprocesadores de ECMAScript: **TypeScript**

Grado en  
Ingeniería  
Informática  
del Software

- Es un lenguaje de programación **libre** y de **código abierto** mantenido por Microsoft
- Creado por *Anders Hejlsberg*
  - diseñador de C#
  - creador de Delphi y Turbo Pascal
- TypeScript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del **cliente** o del **servidor** (Node.js).
- El **sistema de tipos** de TypeScript realiza una formalización de los tipos de Javascript, mediante una **representación estática** de sus tipos dinámicos.
- Se pueden definir **variables y funciones con tipos** sin perder la esencia de JavaScript.
- Poder definir los tipos durante el tiempo de diseño nos ayuda a **evitar errores en tiempo de ejecución**
- <https://www.typescriptlang.org/>



# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones de ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- **WebAssembly**
- Anexos
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## WebAssembly (Wasm): Introducción

Grado en  
Ingeniería  
Informática  
del Software

- Es un estándar web definido en 2017
- Es un **formato binario basado en una máquina de pila**
- Tiene su correspondencia con un código ensamblador
- Permite la **ejecución rápida de aplicaciones** en la web
- Está pensado para ser generado por lenguajes de programación muy eficientes en cuanto al rendimiento
  - Ejemplo: C/C++
- No reemplaza a ECMAScript, pero lo complementa
- Se está desarrollando por el W3C y las principales empresas:
  - <https://www.w3.org/community/webassembly/>
- Sitio oficial:
  - <https://webassembly.org/>

WebAssembly 1.0 has shipped in 4 major browser engines.



# Lenguajes y estándares en la Web

## WebAssembly (Wasm): Ejemplo

Grado en  
Ingeniería  
Informática  
del Software

C input source	Linear assembly bytecode (intermediate representation)	Wasm binary encoding (hexadecimal bytes)
<pre>int factorial(int n) {     if (n == 0)         return 1;     else         return n * factorial(n-1); }</pre>	<pre>get_local 0 i64.eqz if (result i64)     i64.const 1 else     get_local 0     get_local 0     i64.const 1     i64.sub     call 0     i64.mul end</pre>	<pre>20 00 50 04 7E 42 01 05 20 00 20 00 42 01 7D 10 00 7E 0B</pre>

# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

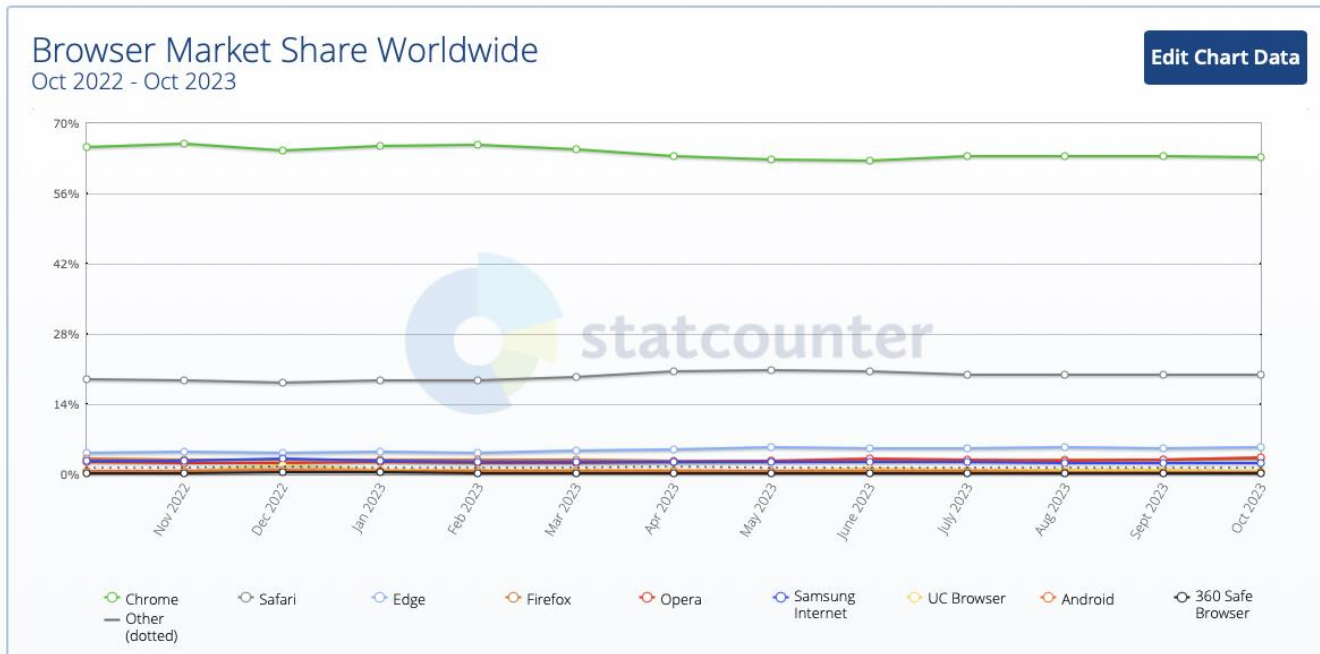
Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones de ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- **Anexos**
- Referencias
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Anexo A-Estadística de uso de los navegadores en el mundo

Grado en  
Ingeniería  
Informática  
del Software

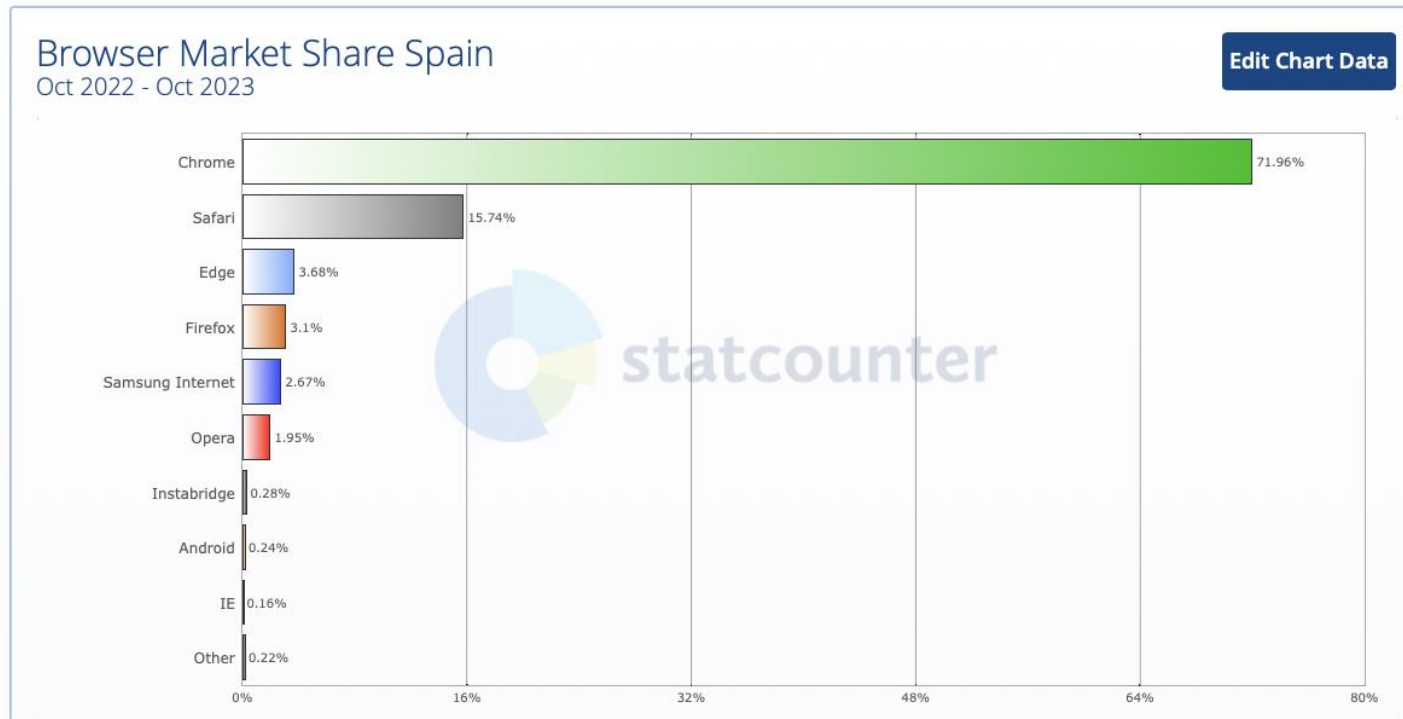


<https://gs.statcounter.com/>

# Lenguajes y estándares en la Web

## Anexo B-Uso de los navegadores en España

Grado en  
Ingeniería  
Informática  
del Software



# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones de ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- **Referencias**
- Ejercicios propuestos

# Lenguajes y estándares en la Web

## Referencias (I)

**Grado en  
Ingeniería  
Informática  
del Software**

- ECMAScript
  - <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- JavaScript
  - <https://developer.mozilla.org/es/docs/Web/JavaScript>
- V8
  - <https://developers.google.com/v8/>
  - [http://en.wikipedia.org/wiki/V8\\_\(JavaScript\\_engine\)](http://en.wikipedia.org/wiki/V8_(JavaScript_engine))
  - [http://en.wikipedia.org/wiki/Google\\_Chrome](http://en.wikipedia.org/wiki/Google_Chrome)
- Gecko
  - [http://en.wikipedia.org/wiki/Gecko\\_\(software\)](http://en.wikipedia.org/wiki/Gecko_(software))
  - <http://en.wikipedia.org/wiki/Firefox>
- Trident
  - [http://en.wikipedia.org/wiki/Trident\\_\(layout\\_engine\)](http://en.wikipedia.org/wiki/Trident_(layout_engine))
  - [http://en.wikipedia.org/wiki/Internet\\_Explorer](http://en.wikipedia.org/wiki/Internet_Explorer)



# Lenguajes y estándares en la Web

## Referencias (II)

Grado en  
Ingeniería  
Informática  
del Software

- Blink
  - <http://www.chromium.org/blink>
  - [http://en.wikipedia.org/wiki/Blink \(layout engine\)](http://en.wikipedia.org/wiki/Blink_(layout_engine))
  - [http://en.wikipedia.org/wiki/Opera \(web browser\)](http://en.wikipedia.org/wiki/Opera_(web_browser))
- Webkit
  - <http://www.webkit.org/>
  - <http://en.wikipedia.org/wiki/WebKit>
  - [http://en.wikipedia.org/wiki/Safari \(web browser\)](http://en.wikipedia.org/wiki/Safari_(web_browser))
- UC Browser (navegador para Android)
  - [https://es.wikipedia.org/wiki/UC Browser](https://es.wikipedia.org/wiki/UC_Browser)
- Estadísticas de uso de navegadores
  - <http://gs.statcounter.com/>

# Lenguajes y estándares en la Web

## Esquema – Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

- Introducción
- Evolución de los lenguajes de Script
- El estándar ECMAScript
- Implementaciones de ECMAScript
- El lenguaje ECMAScript
- Dialectos de ECMAScript
- Extensiones de ECMAScript
- Preprocesadores de ECMAScript
- WebAssembly
- Anexos
- Referencias
- **Ejercicios propuestos**

# Lenguajes y estándares en la Web

## Ejercicios propuestos: OPTATIVO

Grado en  
Ingeniería  
Informática  
del Software

- Hacer un estudio sobre las diferencias de los distintos dialectos de ECMAScript
- Ilustrar el estudio con ejemplos con código fuente ilustrando las diferencias de comportamiento en diferentes navegadores
- Comprobar todo con el mayor número de navegadores posible.

# Lenguajes y estándares en la Web

## Ejercicios propuestos: OPTATIVO

Grado en  
Ingeniería  
Informática  
del Software

- Hacer unos ejemplos con TypeScript
- Verificar su funcionamiento en todos los navegadores

# Lenguajes y estándares en la Web

## Tema 2: Lenguajes de Script

Grado en  
Ingeniería  
Informática  
del Software

