



Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

PECT2 Arq. web

Viernes 10:00 – 12:00

Grado en Ingeniería Informática – Curso 2019/2020

Marcos Barranquero Fernández – 51129104N

JAVASCRIPT

JavaScript es un lenguaje de programación **interpretado** para navegador, **embebido** en páginas HTML. Está basado en **objetos**. No es **fuertemente tipado**, resolviendo las referencias en tiempo de ejecución. Creado por Netscape.

Ventajas:

- Desarrollo rápido de apps.
- Fácil aprendizaje
- Independencia de la plataforma
- Consumo mínimo de recursos

Inconvenientes:

- Métodos limitados solucionados por **librerías**
- Pocas herramientas de desarrollo y depuración.

Usado para:

- Formularios
- Cálculos sencillos
- Hacer la web **interactiva** mediante **eventos**
- Manejo de **gráficos**

Los eventos se pueden asignar a:

- Documentos HTML
- Formularios individuales
- Elementos del formulario

Se añade mediante el gestor: onClick, onChange, etc.

Jerarquía de objetos:

- Window es el nivel más alto de la jerarquía.
- El objeto document contiene todos los elementos del HTML.

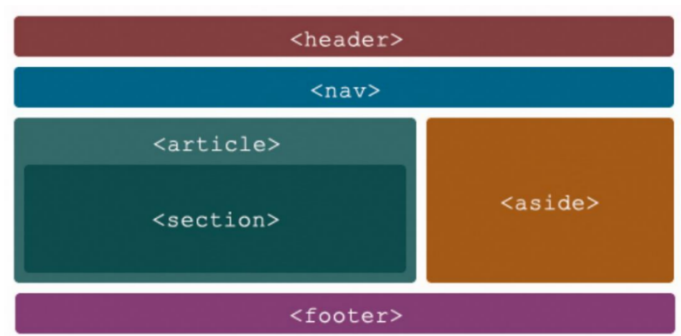
JQUERY

jQuery es una librería Javascript, que ofrece métodos que amplían la funcionalidad de Javascript o reducen considerablemente el código empleado.

La librería viene en un archivo js que no es necesario instalarse, solo importarlo.

HTML5

Surge como colaboración de W3C y WHATWG. Los navegadores aún no soportan toda la implementación.



- El header o encabezado puede tener un logo, un h1-h6, etc.
- El nav contiene un índice que enlaza con otras secciones de la web.
- El footer es un pie de página con info. de autores y copyright.
- El article encapsula contenidos con significado propio.
- El section representa un trozo de documento.
- admite etiquetas para mostrar contenido multimedia directamente.
- Contiene etiquetas para manejar grandes conjuntos de datos y generar tablas dinámicas.
- Mejora sobre formularios para correo, url, fechas, etc.
- Validación de contenido sin necesidad de Javascript. (Regex).
- Visores, Drag&Drop, Almacenamiento local, ubicación geográfica.
- Metainformación.

Soporte para API's:

- Drag&Drop por eventos.
- Trabajar offline descargando en local.
- API Storage: almacenamiento persistente local.
- Websockets: comunicación bidireccional entre páginas.
- WebWorkers: hilos de ejecución en paralelo.
- System information: acceso al HW a bajo nivel, aunque puede comprometer la seguridad.
- Algunas etiquetas eliminadas, otras añadidas, otras cambiadas.

CSS3

CSS es un lenguaje usado para definir la apariencia de un documento HTML. Permite separar la **estructura** del documento de su representación. CSS define estilos para las etiquetas HTML. Las etiquetas tienen un estilo por defecto que CSS sobrescribe.

```
<style type="text/css">
  body
  {
    color: purple;
    background-color: #d8da3d
  }
</style>
```

JAVA ENTERPRISE EDITION

Java EE (ahora Jakarta) es un estándar para desarrollo de aplicaciones empresariales basado en Java. Es un **conjunto de especificaciones** que permiten crear un producto, no el producto en sí.

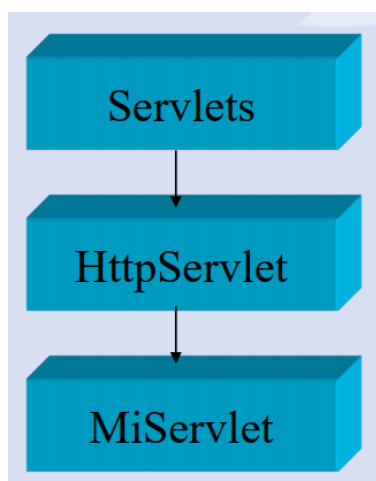
Respecto a desarrollo web, Java EE dispone de **Servlets** y **JSP**.

SERVLETS

- Un servlet es un programa java que se ejecuta como un servidor web, que recibe y responde peticiones por http.
- Los servlets son independientes del servidor y de la plataforma.
- Se puede utilizar como interfaz para comunicar formularios con bases de datos o para el manejo de sesiones.
- El servidor debe tener Java EE instalado. (duh...)

API SERVLET

Se tiene el paquete *javax.servlet* que proporciona una clase para Servlet Genérica.



También está el paquete *httpServlet* que ofrece una interfaz específica para el protocolo Http, que es la más usada.

- *ServletRequest/httpServletRequest* encapsula las peticiones del cliente.
- *ServletResponse/httpServletResponse* encapsula la respuesta del servidor.
- El servlet tiene un método **service** que recibe la petición y la respuesta y realiza las operaciones pertinentes.

El ciclo de vida es el siguiente:

1. El servidor carga el servlet invocando al método **init**.
2. Las peticiones son atendidas por el método **service**.
3. El servlet puede ser descargado con el método **destroy**.

ACCESO A DATOS

JDBC es una api para acceder a BBDD que proporciona independencia sobre el SGBD. Para conectarse:

```
Connection conexion = DriverManager.getConnection(url, user, password)
```

Para lanzar consultas:

```
Statement query = conexion.createStatement();  
ResultSet resultado = query.executeQuery("SELECT * FROM ...");
```

JSP

Permite mezclar código HTML que genera la parte estática con código Java incrustado que genera la parte dinámica. Además, permite acceso a BBDD y sesiones.

Su primera versión es de 1999.

El fichero JSP es convertido en un Servlet por el motor JSP la primera vez que se solicita al servidor web. Si se modifica el JSP, el servlet es regenerado.

CICLO DE VIDA

Es similar a los Servlets:

- `JspInit()` inicializa el servlet asociado.
- `JspService(petición, respuesta)` maneja peticiones.
- `JspDestroy()` invoca el motor para eliminar el servlet.

ELEMENTOS

Declaraciones	<code><% ! Connection c = new Connection %></code>
Código Java (Scriptlets)	<code><% for i ... %></code>
Inserciones	<code><%= 5 * 2 %> // 10</code>

También se tienen objetos implícitos, que son accesibles por el motor JSP y no es necesario que se declaren para su uso. Modelan funcionalidad básica de la web.

Algunos son:

- **Session:** mantiene info. sobre el cliente que ha establecido conexión. El objeto sesión es único para cada usuario. Se pueden escribir, obtener y eliminar atributos. Guarda los atributos como Objects, se debe realizar las conversiones adecuadas.

```
<% session.setAttribute("nombre", "Bachatas"); %>
<% String nombre = (String) session.getAttribute("nombre"); %>
<% session.removeAttribute("nombre"); %>
```

- **Config:** acceso a parámetros de inicialización del servlet
- **Application:** permite comunicación entre JSP y server sin objeto request.
- **Request:** permite obtener parámetros que vienen de formulario.
- **Response y out:** dan métodos para responder y generar la salida HTML.

MVC

Es un patrón de arquitectura que pretende aislar los datos de la aplicación (**modelo**) de la representación (**vista**), unidas por un **controlador** que modela las transiciones de datos y estados.

Para implementarlo, existen multitud de implementaciones. En la asignatura, usaremos:

- Vista: navegador + JSP's
- Controlador: gestión de sesiones, motor de JSP, controlador Servlet, etc.
- Modelo: Base de datos y lógica de la aplicación.