Diseño y Programación Web

Tema 3

HTML (IV)

Enrique Moguel
enrique@unex.es

HTML5

- HTML 5 incluye muchas novedades:
  - Elementos semánticos.
    - **<header>**, **<footer>**, **<section>**,…
  - Elementos de formularios.
    - **Number**, **date**, **time**, **calendar**, **range**, …
  - Elementos gráficos.
    - **<svg>**, **<canvas>**
  - Elementos multimedia.
    - **<audio>**, **<video>**

HTML5

- También APIs:
  - HTML Geolocation.
  - HTML Drag and Drop.
  - HTML Local Storage.
  - HTML Application Cache.
  - HTML Web workers.
  - HTML Server-Sent Events.

HTML5

Year	Version
1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1

HTML5

- Elementos semánticos:

Tag	Description
<article>	Defines an article in the document
<aside>	Defines content aside from the page content
<bdi>	Defines a part of text that might be formatted in a different direction from other text
<details>	Defines additional details that the user can view or hide
<dialog>	Defines a dialog box or window
<figcaption>	Defines a caption for a <figure> element
<figure>	Defines self-contained content, like illustrations, diagrams, photos, code listings, etc.
<footer>	Defines a footer for the document or a section
<header>	Defines a header for the document or a section
<main>	Defines the main content of a document
<mark>	Defines marked or highlighted text

HTML5

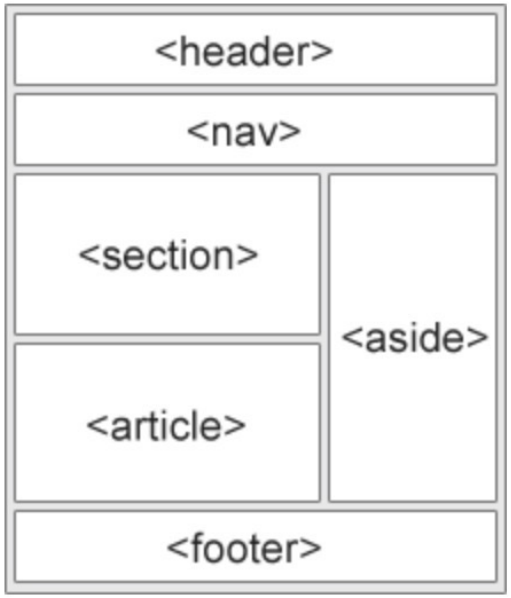
- Elementos semánticos:

<menuitem>	Defines a command/menu item that the user can invoke from a popup menu
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links in the document
<progress>	Defines the progress of a task
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<section>	Defines a section in the document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time
<wbr>	Defines a possible line-break

HTML5

- Elementos semánticos:
  - Tienen un significado claro para el navegador y el desarrollador.
    - ** ** VS **<section>**
    - **** VS **<article>**

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



HTML5

- Elementos de formulario:

Tag	Description
<datalist>	Defines pre-defined options for input controls
<keygen>	Defines a key-pair generator field (for forms)
<output>	Defines the result of a calculation

HTML5

- Tipos de input:

New Input Types	New Input Attributes
• color • date • datetime • datetime-local • email • month • number • range • search • tel • time • url • week	• autocomplete • autofocus • form • formaction • formenctype • formmethod • formnovalidate • formtarget • height and width • list • min and max • multiple • pattern (regexp) • placeholder • required • step

HTML5

- Nueva sintaxis para los atributos:

Type	Example
Empty	<input type="text" value="John" **disabled**>
Unquoted	<input type="text" **value=John**>
Double-quoted	<input type="text" **value="John Doe"**>
Single-quoted	<input type="text" **value='John Doe'**>

HTML5

- Gráficos:

Tag	Description
<canvas>	Draw graphics, on the fly, via scripting (usually JavaScript)
<svg>	Draw scalable vector graphics

- Multimedia:

Tag	Description
<audio>	Defines sound content
<embed>	Defines containers for external applications (like plug-ins)
<source>	Defines sources for <video> and <audio>
<track>	Defines tracks for <video> and <audio>
<video>	Defines video or movie content

HTML5

- Guía de estilo:
  - Más importante de lo que parece.
  - http://www.w3schools.com/html/html5_syntax.asp

HTML5

- Para aprovechar algunas de las características avanzadas de HTML 5 necesitamos usar **javascript**.
  - Veremos JavaScript en profundidad más adelante.
  - Vamos a usar algunos script bastante predefinidos.

HTML5

- Google maps:
  - Podemos incluir mapas de *Google Maps* en nuestras aplicaciones usando su API.
  - Mas info:
    - http://www.w3schools.com/graphics/google_maps_intro.asp

HTML5

- Google maps:
  - Definimos el contenedor del mapa:

```
<div id="map" style="width:400px;height:400px">
```

  - Establecemos las propiedades básicas del mapa:

```
function myMap() {
    var mapOptions = {
        center: new google.maps.LatLng(51.5, -0.12),
        zoom: 10,
        mapTypeId: google.maps.MapTypeId.HYBRID
    }
var map = new google.maps.Map(document.getElementById("map"), mapOptions);
}
```

HTML5

- Google Maps:
  - Añadimos la API de *Google Maps*.

```
<script src="https://maps.googleapis.com/maps/api/js?callback=myMap"></script>
```

  - Tenemos un mapa en nuestra página.

HTML5

- Geolocation API:
  - Se utiliza para obtener la posición de un usuario.
  - Tiene que aceptar.

```
<script>
var x = document.getElementById("demo");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
    } else {
        x.innerHTML = "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}
</script>
```

HTML5

- Geolocation API:
  - Podemos gestionar los errores.

```
function showError(error) {
    switch(error.code) {
        case error.PERMISSION_DENIED:
            x.innerHTML = "User denied the request for Geolocation."
            break;
        case error.POSITION_UNAVAILABLE:
            x.innerHTML = "Location information is unavailable."
            break;
        case error.TIMEOUT:
            x.innerHTML = "The request to get user location timed out."
            break;
        case error.UNKNOWN_ERROR:
            x.innerHTML = "An unknown error occurred."
            break;
    }
}
```

HTML5

- Geolocation API:
  - Mostrar el resultado en un mapa de Google.
    - Ejemplo con imagen estática.
    - Podemos usar la API de Google para hacerlo mejor.

```
function showPosition(position) {
    var latlon = position.coords.latitude + "," + position.coords.longitude;

    var img_url = "http://maps.googleapis.com/maps/api/staticmap?center=
    "+latlon+"&zoom=14&size=400x300&sensor=false";

    document.getElementById("mapholder").innerHTML = "<img src='"+img_url+"'>";
}
```

HTML5

- Geolocation API:
  - Información adicional.

Property	Returns
coords.latitude	The latitude as a decimal number (always returned)
coords.longitude	The longitude as a decimal number (always returned)
coords.accuracy	The accuracy of position (always returned)
coords.altitude	The altitude in meters above the mean sea level (returned if available)
coords.altitudeAccuracy	The altitude accuracy of position (returned if available)
coords.heading	The heading as degrees clockwise from North (returned if available)
coords.speed	The speed in meters per second (returned if available)
timestamp	The date/time of the response (returned if available)

HTML5

- Geolocation API:
  - Podemos monitorizar la posición mientras el usuario se mueve.
    - Hay que probarlo en un móvil.

```
<script>
var x = document.getElementById("demo");
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.watchPosition(showPosition);
    } else {
        x.innerHTML = "Geolocation is not supported by this browser.";
    }
}
function showPosition(position) {
    x.innerHTML = "Latitude: " + position.coords.latitude +
    "<br>Longitude: " + position.coords.longitude;
}
</script>
```

HTML5

- Drag and Drop API:
  - Permite arrastrar y soltar objetos en HTML.

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev) {
    ev.preventDefault();
}

function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<img id="drag1" src="img_logo.gif" draggable="true"
ondragstart="drag(event)" width="336" height="69">

</body>
</html>
```

HTML5

- Local Storage API:
  - Alternativa a las cookies tradicionales.
  - Permite almacenar información en la máquina de los usuarios.

```
// Store
localStorage.setItem("lastname", "Smith");
// Retrieve
document.getElementById("result").innerHTML = localStorage.getItem("lastname");
```

```
if (localStorage.clickcount) {
    localStorage.clickcount = Number(localStorage.clickcount) + 1;
} else {
    localStorage.clickcount = 1;
}
document.getElementById("result").innerHTML = "You have clicked the button " +
localStorage.clickcount + " time(s).";
```

HTML5

- Application Cache API:
  - Permite almacenar aplicaciones web en una cache.
    - Accesible sin conexión.

```
<!DOCTYPE HTML>
<html manifest="demo.appcache">

<body>
The content of the document......
</body>

</html>
```

```
CACHE MANIFEST
# 2012-02-21 v1.0.0
/theme.css
/logo.gif
/main.js

NETWORK:
login.asp

FALLBACK:
/html/ /offline.html
```

https://www.html5rocks.com/es/tutorials/appcache/beginner/

HTML5

- Web Workers API:
  - Permite definir tareas javascript que se ejecutan en segundo plano (asíncronas).

```
var i = 0;

function timedCount() {
    i = i + 1;
    postMessage(i);
    setTimeout("timedCount()",500);
}

timedCount();
```

```
if (typeof(w) == "undefined") {
    w = new Worker("demo_workers.js");
}
```

```
w.onmessage = function(event){
    document.getElementById("result").innerHTML = event.data;
};
```

https://www.w3schools.com/html/tryit.asp?filename=tryhtml5_webworker

HTML5

- Server-Sent Events API:
  - Permite que una página web sea modificada por iniciativa del servidor.

```
var source = new EventSource("demo_sse.php");
source.onmessage = function(event) {
    document.getElementById("result").innerHTML += event.data + "<br>";
};
```

```
<?php
header('Content-Type: text/event-stream');
header('Cache-Control: no-cache');

$time = date('r');
echo "data: The server time is: {$time}\n\n";
flush();
?>
```

Diseño y Programación Web

Tema 3

HTML

Enrique Moguel
enrique@unex.es