

# Introducción al curso y al programa

# Curso: Diseño de aplicaciones Web en HTML5 y para móviles FirefoxOS

Curso de **iniciación**

al **diseño** y a la **programación** en **HTML5**

(**HTML**, **CSS** y **JavaScript**)

de **aplicaciones Web multi-dispositivo**

(PC, tableta, teléfono, ...)

y a su empaquetamiento para

**telefonos FirefoxOS** o para publicarlas

en **Internet** o en su **AppStore**



# Diseño de servicios en la nube, utilizando JavaScript, para acceso movil y multi-dispositivo con HTML5\*



## ◆ Programa de especialización con 5 cursos MOOC

- Desarrollo de Aplicaciones en HTML5, CSS y Javascript, incluyendo Dispositivos Móviles Firefox O.S.
- Desarrollo avanzado de Aplicaciones HTML5 y Firefox O.S.\*, incluyendo tecnicas de ingeniería de software
- Desarrollo de aplicaciones HTML5 multi-terminal (Android, iOS, ...)\*
- Desarrollo de servicios en la nube con HTML5, Javascript y node.js\*
- Desarrollo avanzado de servicios en la nube con Javascript y node.js\*



\***Nota:** Los otros 4 cursos empezarán en MiriadaX en 2014 y 2015.



- ◆ **Modulo 1:** Introducción a HTML, CSS, URLs y a las herramientas del curso.
- ◆ **Modulo 2:** Estructura de la navegación y el texto en páginas Web HTML5, prioridad en CSS3 y publicación en la nube de Google Drive de los ejercicios del curso.
- ◆ **Modulo 3:** Estructuración, visualización y depuración con Firefox de HTML5 y CSS3 adaptados a un entorno responsivo y multi-pantalla, para adaptación a móviles y tabletas.
- ◆ **Modulo 4:** Introducción a JavaScript, sus sentencias, variables, expresiones, scripts de cliente, así como a la utilización de la consola JavaScript del navegador Firefox.
- ◆ **Modulo 5:** Primeros programas Javascript con booleanos, igualdad y comparaciones, sentencia condicional, números, funciones, strings e internacionalización.
- ◆ **Modulo 6:** Objetos, propiedades, métodos, eventos, formularios, sentencia for/in y características avanzadas de objetos, incluyendo acceso a la página Web cargada en el navegador a través del árbol DOM (Document Object Model)
- ◆ **Modulo 7:** Interacción con el usuario, eventos y sus manejadores, formularios, jQuery, Zepto, Arrays y bucles.
- ◆ **Modulo 8:** Ejemplo de un cronómetro, eventos táctiles, localStorage y JSON, así como su utilización en aplicaciones Web.
- ◆ **Modulo 9:** Introducción a los gráficos y animaciones con SVG (Scalable Vector Graphics) y HTML5 CANVAS, así como al empaquetamiento de web apps para FirefoxOS.
- ◆ **Modulo 10:** API de geolocalización HTML5 y Mash-up con Google Maps, así como realización del proyecto final del curso.



## Módulos

# Desarrollo y evaluación del curso

- ◆ El curso consta de 5 tramos de 2 módulos
  - Diseñado para realizarse en 5 semanas (1 semana por tramo)
    - ◆ Pero se deja una semana adicional por tramo (~10 semanas, 2 semanas por tramo)
- ◆ Ejercicios P2P de **entrega obligatoria**
  - Al final de cada módulo (2 ejercicios por tramo)
    - además hay tests obligatorios y más ejercicios P2P opcionales
- ◆ Plazos y límites
  - Tramo 1: comienzo lunes **01-10** (o antes), final **19-10**
  - Tramo 2: comienzo lunes **12-10** (o antes), final **02-11**
  - Tramo 3: comienzo lunes **19-10** (o antes), final **16-11**
  - Tramo 4: comienzo lunes **26-10** (o antes), final **30-11**
  - Tramo 5: comienzo lunes **02-11** (o antes), final **14-12**
  - Evaluaciones y curso cierran el **22-12**

# Actividades de un Módulo

## ◆ **Tarea 0:** Descargar transparencias y ejemplos del módulo

- Fichero ZIP para descargar con
  - ◆ Transparencias en formato PDF
  - ◆ Directorio con ejemplos presentados en el módulo
    - para realizar ejercicios modificando los descargados
- Directorio con ejemplos publicados en Internet (Google Drive)
  - ◆ **ejecutables en Firefox**

## ◆ **Tareas de Aprendizaje (varias):**

- un **video** o **screencast** del tema (3 y 14 minutos)
  - ◆ evaluado (no siempre) con un **test** o un **ejercicio P2P opcional**

## ◆ **Tarea final:** Ejercicio P2P de entrega obligatoria

- Cada modulo acaba con un ejercicio P2P obligatorio

Tarea 0: Transparencias y Ejemplos de un Módulo

Desarrollo de Aplicaciones en HTML5 y para Dispositivos Móviles Firefox O.S.

Desactivar edición

Modulo 1: Introducción a Internet, el Web, la nube, HTML5 y CS

Lista de Módulos

+ Nuevo

Transparencias y ejemplos del modulo

Las transparencias y los ejemplos están en el fichero ZIP adjunto, que se pueden...

Además, tal y como se explica en el Tema 0, haciendo click aquí se accede a un c... en Google Drive, donde pueden ver los ejemplos funcionando al hacer clic...

modulo01.zip

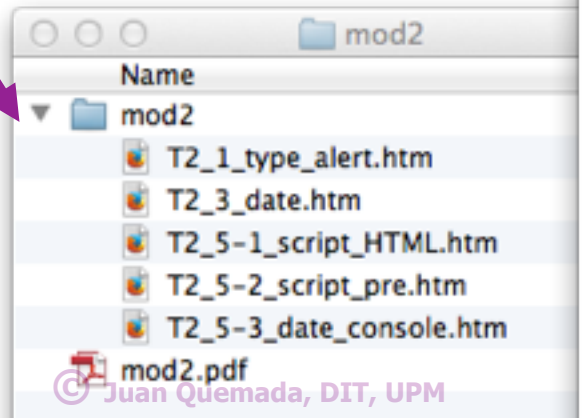
haciendo click aquí

Actividades del Módulo

● Módulo 0. Introducción al curso y al programa "Diseño de servicios en la nube para acceso movil con HTML5"

● Modulo 1: Introducción a Internet, el Web, la nube, HTML5 y CSS

Descarga



mod2

TITLE

← Back to parent

Archive.zip

T2\_1\_type\_alert.htm

T2\_3\_date.htm

T2\_5-1\_script\_HTML.htm

T2\_5-2\_script\_pre.htm

T2\_5-3\_date\_console.htm

mod2.pdf

# Equipos y servicios a utilizar

- ◆ Un PC o portatil de trabajo (necesario)
  - con S.O. Windows de Microsoft
  - con S.O. UNIX (Ubuntu, ..) de software libre
  - Ordenador MAC con sistema operativo con OS X
- ◆ Móvil o tableta (conveniente para probar, pero no necesario)
  - Android, iPhone, iPad, FirefoxOS, .. pero no es necesario
    - ◆ Se puede utilizar un simulador de FirefoxOS para probar Apps
- ◆ Cuenta en Google para **Google Drive** o **Google Docs**
  - para publicar Apps y Páginas Web en la nube
    - ◆ Es la cuenta unificada de acceso a Gmail, Youtube, Google+, Docs, Drive, ...



# Herramientas a utilizar

- ◆ Simulador/Editor interactivo sencillo: <http://vishub.org/categories/43>
  - Editor y visualizador de HTML5, CSS y JS con ejemplos cargados
- ◆ Sublime Text 2
  - Editor de HTML, CCS y JavaScript con ayudas visuales
    - ◆ <http://www.sublimetext.com>
- ◆ Creadores (wysiwyg) de páginas Web (opcional)
  - Dreamweaver: Creador de páginas Web muy potente e intuitivo.
    - ◆ <http://www.adobe.com/es/products/dreamweaver.html>
  - Bluegriffon: editor wysiwyg gratuito y de código abierto
    - ◆ <http://www.bluegriffon.org>
- ◆ Navegador Firefox
  - con su entorno de desarrollo de aplicaciones Web y FirefoxOS

<http://vishub.org/categories/43>

Seleccionar ejemplo

Hacer clic para actualizar pag.

Hacer clic en el Modulo 1

# Editor Interactivo

Editor interactivo de los ejemplos JavaScript del módulo 1 de @HTML5MOOC

Los ejemplos se pueden cambiar y ejecutar (visualizar) con los cambios introducidos pulsando el botón play, que está justo encima.

html css

Ejemplo:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Titulo del documento</title>

<style type="text/css">
body {
  font-family: Arial, Helvetica, sans-serif;
  color: red;
  background-color: #E9E9E9;
}
</style>
</head>
<body>
Mi primera página editada con el editor interactivo
</body>
</html>
```



Resultado:

Mi primera página editada con el editor interactivo

- ◆ Herramientas de desarrollo web de Firefox.
  - <https://developer.mozilla.org/en-US/docs/Tools>
- ◆ Inspector HTML de Firefox.
  - [https://developer.mozilla.org/en-US/docs/Tools/Page\\_Inspector](https://developer.mozilla.org/en-US/docs/Tools/Page_Inspector)
- ◆ Editor de Estilos de Firefox.
  - [https://developer.mozilla.org/en-US/docs/Tools/Style\\_Editor](https://developer.mozilla.org/en-US/docs/Tools/Style_Editor)
- ◆ Herramientas autónomas de Firefox.
  - Visor de repintado: [https://developer.mozilla.org/en-US/docs/Tools/Paint\\_Flashing\\_Tool](https://developer.mozilla.org/en-US/docs/Tools/Paint_Flashing_Tool)
  - Visor 3D: [https://developer.mozilla.org/en-US/docs/Tools/3D\\_View](https://developer.mozilla.org/en-US/docs/Tools/3D_View)
  - Borrador Javascript: <https://developer.mozilla.org/en-US/docs/Tools/Scratchpad>
  - Visor de diseño adaptativo: [https://developer.mozilla.org/en-US/docs/Tools/Responsive\\_Design\\_View](https://developer.mozilla.org/en-US/docs/Tools/Responsive_Design_View)
- ◆ Consola Web de Firefox.
  - [https://developer.mozilla.org/en-US/docs/Tools/Web\\_Console](https://developer.mozilla.org/en-US/docs/Tools/Web_Console)
- ◆ Depurador Javascript de Firefox.
  - <https://developer.mozilla.org/en-US/docs/Tools/Debugger>
- ◆ Perfilador de Firefox.
  - <https://developer.mozilla.org/en-US/docs/Tools/Profiler>
- ◆ Simulador de Firefox OS
  - Instalación: <https://addons.mozilla.org/en-US/firefox/addon/firefox-os-simulator>
  - Documentación: [https://developer.mozilla.org/en-US/docs/Tools/Firefox\\_OS\\_Simulator](https://developer.mozilla.org/en-US/docs/Tools/Firefox_OS_Simulator)

# Documentación de Firefox y sus herramientas

# Introducción a Internet y a la Web

# Cientes, servidores y la nube



## ◆ **Cientes:**

- Dan acceso a información y servicios en Internet

## ◆ **Servidores:**

- Alojan la información y los servicios

## ◆ **La nube:** conjunto de terminales y servidores

- interconectados con aplicaciones y protocolos de Internet

## ◆ **TCP/IP:** protocolos de interconexión de redes de fibra, cable, WIFI, ...

- sobre los que se implementan las aplicaciones de Internet y sus protocolos

# Cientes, navegadores y tiendas



## ◆ **Cientes** de acceso a Internet más importantes

- PCs, portátiles, tabletas, teléfonos inteligentes

## ◆ **Navegador (browser)** cliente Web de acceso a servidores

- Utilizando: **URL, HTTP, HTML, CSS y JS**
  - ◆ p.e. Chrome, Firefox, Internet Explorer, Opera, Safari, ...



## ◆ **Tiendas de aplicaciones**

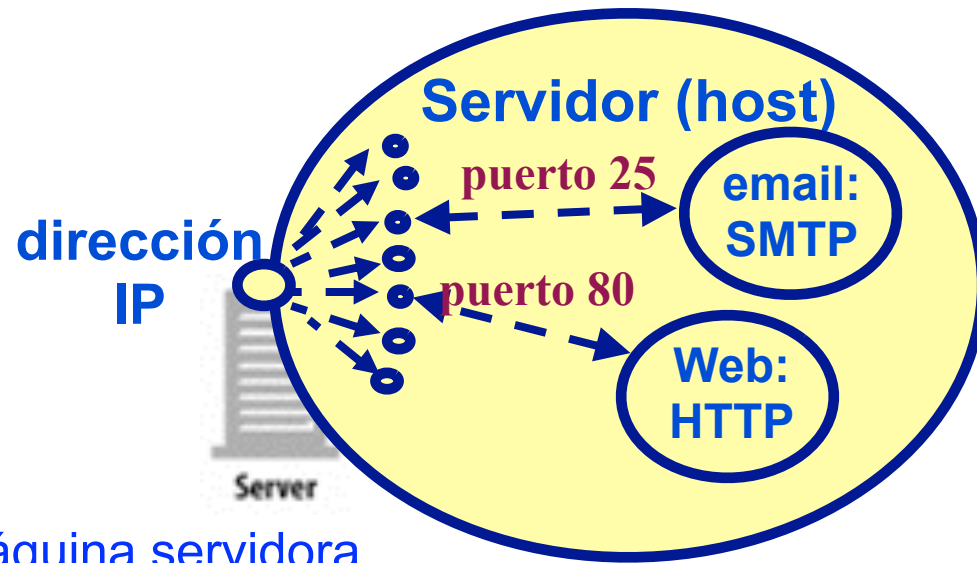
- Instalan aplicaciones en móviles y tabletas
  - ◆ Las aplicaciones usan las normas de la Web (URL, HTTP, ....)

# Máquina servidora (host)



- ◆ Contiene información y servicios
- ◆ Una máquina servidora tiene una dirección “**conocida**” en Internet
  - Dirección simbólica (de dominio o **DNS**): **upm.es**, **google.com**, ...
    - ◆ Cada dirección de dominio tiene una dirección IP (binaria) asociada
- ◆ Hay 2 tipos de direcciones IP: IPv4 e IPv6
  - **IPv4**: versión 4 del protocolo IP con dirección de **32 bits** o **4 octetos**
    - ◆ ejemplo: **192.9.0.144**, **127.0.0.1** (localhost - mi máquina), ...
  - **IPv6**: versión 6 del protocolo IP (última) con dirección de **128 bits**
    - ◆ ejemplo: **2001:db8:85a3::8a2e:370:7334**, ....

# Servidores y puertos



## ◆ Puerto

- Dirección de 16 bits dentro de la máquina servidora
  - ◆ Es donde se instala el **programa servidor**
- El **programa servidor** es lo que normalmente denominamos **servidor**
  - ◆ Cliente y servidor se comunican a través de un protocolo: HTTP, SMTP, .....
  - Utilizando el interfaz de sockets TCP/IP para comunicar entre ambos

## ◆ Los servicios tienen un protocolo y un puerto por defecto

- **Web:** protocolo HTTP (puerto 80), HTTPS (443)
- **Email:** protocolo SMTP (puerto 25), POP3 (110), IMAP143)
- **Shell segura:** protocolo SSH (puerto 22)

## ◆ Si un servidor no está en el puerto por defecto

- Su dirección debe incluir el puerto, p.e. [dit.upm.es:8080](http://dit.upm.es:8080), [192.9.0.144:8080](http://192.9.0.144:8080)



# URL



## ◆ URL (Uniform Resource Locator)

- Dirección de un recurso en un servidor en Internet
- ◆ Internet soporta muchos tipos de servicios diferentes
  - Cada tipo de servicio utiliza un URL y protocolo diferentes
- ◆ Algunos ejemplos de tipos de URLs
  - **URL Web:** utiliza HTTP para acceder a recursos, incluye
    - ◆ Protocolo, servidor y recurso (camino): <http://google.com/picture.png>
  - **URL de correo (email):** identifica el buzón de usuario, incluye
    - ◆ protocolo, buzón de usuario y servidor: [mailto:pepe\\_garcia@gmail.com](mailto:pepe_garcia@gmail.com)

# HTTP (HiperText Transfer Protocol)

## ◆ Protocolo del Web

- Procesa recursos identificados por un URL en un servidor remoto

## ◆ Métodos o comandos principales de HTTP

- **GET:** trae al cliente (lee) un recurso identificado por un **URL**
- **POST:** crea un recurso identificado por un URL
- **PUT:** actualiza un recurso identificado por un URL
- **DELETE:** borra un recurso identificado por un URL
- ..... (hay mas comandos)



# Aplicación Web

- ◆ Aplicaciones ejecutables en un navegador creadas con
  - **HTML, CSS y JavaScript**



- ◆ **HTML**

- Lenguaje de marcado de páginas Web
  - ◆ define la estructura del contenido de una página Web
- En WebApps define la interfaz de la aplicación con el usuario

- ◆ **CSS**

- Define el estilo visual de un una página o aplicación Web (HTML)

- ◆ **JavaScript**

- Lenguaje de programación de aplicaciones de cliente

# Aplicación Web: HTML, CSS y JavaScript

## ◆ HTML

- Lenguaje de marcado

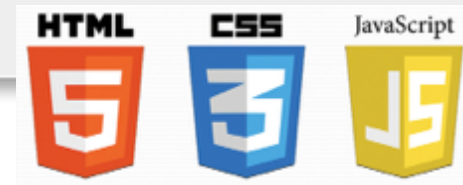
## ◆ CSS

- Estilo la visualización

## ◆ JavaScript

- Lenguaje de programación

```
02-date_CSS.htm UNREGISTERED
<!DOCTYPE html><html>
<head>
  <title>Ejemplo</title>
  <meta charset="UTF-8">
  <style type="text/css">
    body {color: blue;}
  </style>
</head>
<body>
  <h3>Fecha y hora</h3>
  <script type="text/javascript">
    document.write(new Date());
  </script>
</body>
</html>
```



# WebApps o aplicaciones de cliente

## ◆ Aplicaciones que **residen en un servidor**

- pero se **ejecutan en un cliente**

- ◆ Se identifican con un **URL**: `http://upm.es/apps/webapp.html`

- El cliente trae la aplicación del servidor con el **protocolo HTTP (GET)**

## ◆ Las apps se construyen con las tecnologías de la Web

- URLs, HTTP, HTML, CSS y JavaScript

1) Cliente solicita WebApp identificada con URL

2) Script se ejecuta al cargar la página Web en el navegador:



Client

Solicitud HTTP GET asociada a un URL

Respuesta HTTP: página Web con script



Server

Servidor sirve fichero identificado por URL

```
02-date_CSS.htm UNREGISTERED
<!DOCTYPE html><html>
<head>
<title>Ejemplo</title>
<meta charset="UTF-8">
<style type="text/css">
  body {color: blue;}
</style>
</head>
<body>
<h3>Fecha y hora</h3>
<script type="text/javascript">
  document.write(new Date());
</script>
</body>
</html>
```

