

# Web Programming Foundations

## Contenido

Glosario .....	2
Links útiles.....	2
Resumen .....	3
Capítulo 1. ....	3
Capítulo 2. ....	3
Capítulo 3. ....	4
Capítulo 4. ....	4

## Glosario

- **UX User Experience:** la forma como un sitio se ve y se “siente”
- **UI User Interface:** La funcionalidad de un sitio
- **Servers:** Machines that serve up information
- **NoSQL:** permite archivos sin una estructura específica.
- **Forking:** Crear versión propia de un código (como en GitHub)
- **IDE:** Integrated Development Environment
- **Drop Cap:** A drop cap (dropped capital) is a large capital letter used as a decorative element at the beginning of a paragraph or section

## Links útiles

[www.slack.com](http://www.slack.com)

[www.browsersync.io](http://www.browsersync.io) (Para probar en distintos equipos un sitio web creado)

# Resumen

## Capítulo 1.

- URL = *Universal Resource Locator*  
Apunta a un recurso universal en el internet.
  1. Cuando se digita la URL en la casilla de dirección, el navegador envía una petición GET a esta URL para que envíe todos los datos (o el recurso) que tiene.
  2. El recurso que recibe es el HTML junto a las referencias de imágenes, videos, audios y los links a los CSS y JavaScript.
  3. El navegador junta el HTML, CSS y JavaScript y muestra la página como se debe mostrar.
- Cuando un navegador exhibe un documento de HTML, crea el DOM que es la representación estructural de los elementos en el documento y sus relaciones.
- Se puede visualizar como el Árbol de DOM

## Capítulo 2.

Metadata: Es datos agregados a un documento web para proveer mayor información sobre el propósito y función.

- Links, title, scripts
- Schemas y microformats
- ARIA Accesible Rich Internet Attributes: Ayuda a los lectores de pantalla.

## Accesibilidad

Es una de las partes más importantes comparándose con la estética (aesthetic) y la interactividad. El contenido debe ser accesible sin importar el equipo o herramientas que usen, ya sean con un mouse, teclado, con una pantalla táctil, etc.

Cualquier página que uno cree debe seguir estos principios:

- Primero hacerlo accesible
- Después hacerlo *Fancy* (fantástico)
- Verificar que lo que se hizo para hacerlo *Fancy* no haya dañado la accesibilidad.

## Capítulo 3.

### Herramientas

- Navegador:
  - La aplicación o intérprete que lee los documentos web.
  - Estos intérpretes no muestran el contenido de la misma manera, por eso se debe testear en distintos navegadores.
  - El navegador debe ser considerado como solo una forma de mostrar el contenido, ya que es posible que en el futuro se acceda a este contenido de otra forma.
- Editor de código
  - Resalta la sintaxis del lenguaje
  - Code Hinting.
  - Gestión del proyecto para acceder a todos los archivos de una carpeta.
- Herramientas de desarrollador

## Capítulo 4.

### Las partes que componen la Web

- HTML para contenido
- CSS para estilo
- JavaScript para interactividad

### HTML

Es el documento Web. Hoy en día se crean plantillas para mostrar la estructura HTML y el servidor o el navegador combina las plantillas con datos de la base de datos u otro recurso para crear el documento web final.

- *¿Quién escribe el HTML?*

#### Who Writes the HTML?

- Human + code editor
- Human + server-side app
- Human + JS framework

## CSS

- Las reglas se aplican de arriba hacia abajo de la hoja de estilos, haciendo que el mayor peso dado a las reglas lo componen los elementos ubicados más abajo.
- Los descendientes heredan las propiedades de estilo

## JavaScript

Un lenguaje de scripting que corre en el navegador que interactúa con el HTML y el CSS y los eventos dentro del viewport del navegador.

La evolución de Javascript ha sido desde la manipulación única de HTML y CSS para añadir interactividad, a aplicaciones que lograr correr en el navegador o en un servidor para generar documentos y vistas.

## Eventos

Los eventos son manipulados desde un JavaScript. Son todas las interacciones que se tienen con el navegador.

La técnica es llamada *Event Handling*

- Identificar el nodo DOM a monitorear
- Identificar el evento al que se quiera responder
- Crear una función que se ejecute cuanto el evento ocurra.