

#### [Duración](http://hackspace.la/frontend.html)

4 Semanas

#### [Entrega Final](http://hackspace.la/frontend.html)

14 Feb 2016

#### Sobre los Entrenadores

* 

Alvaro Concha

Alvaro es Desarrollador Frontend con varios años de experiencia. Además, es uno de los fundadores de HackSpace.

* 

Martin Vuelta

Martin es un Desarrollador Fullstack con experiencia en diversos lenguajes y frameworks. Está demás decir que es un excelente desarrollador Frontend con mucho enfoque en los detalles.

#### Programas Necesarios

[Google Chrome](https://www.google.com/chrome/browser/desktop/index.html), [Sublime Text 2](http://www.sublimetext.com/2)

#### Semana 1 : Fundamentos, HTML y CSS

**Ejemplos**Introducción práctica para entender los fundamentos de Frontend.

[F-Ejemplos-1.2.zip ( 2 KB )](http://hackspace.la/assets/ejemplos/F-Ejemplos-1.2.zip)

Configurando el entorno

Antes de empezar, debemos tener las herramientas necesarias para visualizar correctamente y entender mejor qué es lo que sucede en una página web.

Estas son las herramientas que necesitaremos:

* En Windows/Linux/MAC
  1. [Google Chrome](https://www.google.com/chrome/browser/desktop/index.html): Descárgate este navegador (si aún no lo tienes), más adelante lo utilizaremos con mayor profundidad.
  2. [Sublime Text 2](http://www.sublimetext.com/2): Este es uno de los mejores editores de texto gratuitos disponibles. Les servirá para poder abrir los archivos que les brindaremos, para revisarlos y editarlos en su computadora.

Ejemplo 1

Ahora que ya tenemos los programas necesarios descargados, empecemos con un primer ejemplo simple de un archivo HTML sin CSS ni Javascript.

Abre un nuevo archivo en Sublime Text 2 (el cual también descargaste anteriormente). Y escribe el siguiente código en ese documento:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Página 1</title>

</head>

<body>

<h1>Título</h1>

<p>Párrafo: Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laborum sequi optio ullam ad nihil dolores eum, officiis assumenda delectus ea.</p>

<img src="http://hackspace.la/img/ejemplos/logo-hs.png" alt="HackSpace Perú">

<button type="button">¡Hazme clic!</button>

</body>

</html>

Si guardas el archivo con la extención \*.html y abres este archivo con Google Chrome verás lo siguiente:



Como verás, este es un archivo de HTML simplificado. Ahora seguiremos con el segundo ejemplo.

Ejemplo 2

Continuaremos con un archivo HTML con CSS y Javascript incluido dentro de él.

En este ejemplo añadimos dentro del <body> el estilo (CSS) y el script (Javascript), el orden en el que ha sido colocado influye en la funcionalidad que posee, pero ahora lo que nos interesa es solo ver que funciona.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Página 2</title>

<style type="text/css">

.clase1 {

color: blue;

}

</style>

</head>

<body>

<h1 class="clase1">Título</h1>

<p id="demo">Párrafo: Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laborum sequi optio ullam ad nihil dolores eum, officiis assumenda delectus ea.</p>

<img src="http://hackspace.la/img/ejemplos/logo-hs.png" alt="HackSpace Perú">

<script type="text/javascript">

function miFuncion() {

var x = document.getElementById("demo");

x.style.fontSize = "25px";

x.style.color = "red";

}

</script>

<button type="button" onclick="miFuncion()">¡Hazme clic!</button>

</body>

</html>

Si abres este archivo con Google Chrome verás lo siguiente:



El mejor lugar para colocar la etiqueta <style> es el <head> ya que la página empezaría a renderizar el<body> ya con los estilos definidos.

Por otro lado, en este ejemplo vemos cómo con el CSS estamos cambiando el color del Título. Ya demás, con el Javascript estamos añadiendo la funcionalidad que si haces clic en el botón ¡Hazme Clic! agregamos atributos CSS al Párrafo y este crece y cambia de color.

Ya que entendemos mejor a qué nos referíamos cuando decíamos que el CSS cambia el estilo y el Javascript agrega funcionalidad, veremos un último ejemplo con estos archivos separados.

Ejemplo 3

Debido a que cuando trabajamos en una web tenemos bastante código para trabajar, es preferible separar y organizar el HTML, CSS y Javascript en archivos y hasta carpetas diferentes. De esta manera, en el ejemplo anterior lo podemos realizar también de esta otra manera:

1. Creamos los archivos ejemplo3.html, style.css y main.js, y los ponemos en una ubicación que conozcamos (en este caso los 3 están en una misma carpeta). Estos son los archivos HTML, CSS y Javascript respectivamente y pueden tener cualquier nombre que queramos pero con sus respectivas terminaciones.
2. Abrimos el archivo HTML y vemos que en vez del código CSS y Javascript que había anteriormente, ahora hay un links dentro de la etiqueta <head> y la etiqueta <script> de Javascript con código que teníamos se redujo a un src="main.js"
3. Abrimos el archivo CSS llamado style.css y vemos el mismo código que teníamos dentro de <style type="text/css">.
4. Abrimos el archivo Javascript llamado main.js y copiamos el mismo código que teníamos dentro de<script type="text/javascript" src="static/template/themes/admin/layout.js"></script>.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Página 3</title>

<link type="text/css" href="style.css" rel="stylesheet">

</head>

<body>

<h1 class="clase1">Título</h1>

<p id="demo">Párrafo: Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laborum sequi optio ullam ad nihil dolores eum, officiis assumenda delectus ea.</p>

<img src="http://hackspace.la/img/ejemplos/logo-hs.png" alt="HackSpace Perú">

<button type="button" onclick="miFuncion()">¡Hazme clic!</button>

<script type="text/javascript" src="main.js"></script>

</body>

</html>

Si abres el archivo ejemplo3.html con Google Chrome verás que se ve exactamente igual que en el Ejemplo 2 y tiene la misma funcionalidad, pero la diferencia es que lo hemos organizado todo de una mejor manera:



Vale la pena aclarar que la intención de estos ejemplos no es que entiendas el código que estamos utilizando, ese será tema de las siguientes secciones. Pero si es que haz entendido mejor cómo es que los archivos HTML, CSS y Javascript interactúan, entonces estás listo para empezar a aprender a desarrollar con estos lenguajes.

**Teoría**HTML: Empezando un documento HTML y conociendo las etiquetas.

Cómo comenzar un documento HTML

Lo primero que se declara siempre es el <!DOCTYPE>. Con esto definimos que versión de html usaremos.<!DOCTYPE html> define html5.

Todo html sigue una estructura similar, primero se escribe la etiqueta <html></html> y dentro de el contiene una etiqueta <head></head> el cual define información del documento y otra <body></body> el cuál define información del cuerpo del documento:

<!DOCTYPE html>

<html>

<head>

</head>

<body>

<!– Así escribimos comentarios –>

</body>

</html>

Generalmente las etiquetas de abren <etiqueta> y cierran </etiqueta>, teniendo contenido por dentro, habiendo excepciones que no se cierran porque no tienen contenido, como <img>, <br> o <base>.

Otro concepto fundamental es entender que cada etiqueta tiene atributos y cada atributo asume un valor:

<etiqueta atributo1=valor1 atributo2=valor2 .... atributon=valorn></etiqueta>

El **atributo id** por ejemplo define un identificador único en todo el documento que servirá para asignarle estilos o scripts al componente de la etiqueta en que se encuentre gracias a los llamados **selectores**. Otro atributo que permite también acceder a elementos específicos es el **atributo class**. Para mayor detalle de los atributos se recomienda este [enlace](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes).

Dentro del **head** y del **body** también encontramos otras etiquetas que ayudan a definir la información del documento y de su cuerpo, respectivamente.

Etiquetas principales del <head>

Las etiquetas más comunes que encontramos son:

| **Etiqueta** | **Descripción** | **Atributos** |
| --- | --- | --- |
| <title> | Define el título del documento. |  |
| <meta> | Especifica una pareja propiedad/valor del documento. | name: Identifica un nombre de propiedad. |
|  |  | http-equiv: Los servidores HTTP utilizan este atributo para obtener información sobre los encabezados. |
|  |  | content: Este atributo especifica el valor de una propiedad. |
|  |  | scheme: Proporciona a los agentes de usuario más contexto para la interpretación correcta de los datos. |
| <link> | Define un vínculo. Solo puede aparecer en la sección head de un documento. |  |
|  |  | charset: Especifica la codificación de caracteres. |
|  |  | href: Especifica la localización de un recurso de la web. |
|  |  | hreflang: Especifica el lenguaje de la URL |
|  |  | media: El medio para la información del estilo. |
|  |  | target: Indica donde abrir el URL. |
| <style> | Define el estilo de la información del documento html. |  |
| <base> | Especifica la URL base que será usada para todos los vínculos de la página.. |  |

Revisar este [link](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/head#See_also) para mayor profundidad.

Definamos un archivo **head.html**:

<!DOCTYPE html>

<html>

<head>

<title>Mi primer titulo</title>

<meta charset="UTF-8">

<meta name="description" content="Core Upgrade 2015">

<meta name="keywords" content="HTML,CSS,XML,JavaScript">

<meta name="author" content="Hackspace">

<link rel="stylesheet" type="text/css" href="styles.css">

<style>

h1 {color:red;}

p {color:blue;}

</style>

</head>

<body>

<h1>Style rojo.</h1>

<p>Style azul.</p>

</body>

</html>

Este será el archivo **styles.css**:

h1 {

color: green;

border: '1pt solid black'

}

p {

color: red;

background-color:#EFE7D6;

border: '1pt solid black'

}

div {

color: #FFFFFF;

background-color:#000000;

}

span {

color: #000000;

background-color:#FFFFFF;

}

Etiquetas de texto y contenido

| **Etiqueta** | **Descripción** | **Atributos** |
| --- | --- | --- |
| <b> | Define un texto en negrita. |  |
| <h1>-<h6> | Definen cabeceras de texto. Ordenadas de más a menos importante. |  |
| <p> | Define un párrafo. |  |
| <div> | Define una división o sección. |  |
| <span> | Permite agrupar varios elementos en línea seguidos dentro de un mismo bloque (por ejemplo, varias palabras seguidas en un párrafo), para después darles formato con la hoja de estilo. |  |
| <iframe> | Inserta de manera embebida otro documento dentro del documento html actual. |  |

Veamos el ejemplo con estas etiquetas:

<!DOCTYPE html>

<html>

<body>

<div style="color:#0000FF">

<p><b>Semana 1:</b></p>

<h1>HTML</h1>

<h2>CSS</h2>

<h3>Javascript</h3>

</div>

<div style="color:#00FFFF">

<p><b>Semana 2:</b></p>

<h4>Bootstrap/Foundations</h4>

<h5>Firebase</h5>

<h6>Nodejs</h6>

</div>

<p>La Semana 1 <span style="color:#0000FF;font-weight:bold">sera buena</span> y la Semana 2 <span style="color:00FFFF;font-weight:bold">aun mejor </span> :).</p>

<br>

<iframe src="http://hackspace.la/" />

</body>

</html>

Introducción a JavaScript con la etiqueta <script></script>

| **Etiqueta** | **Descripción** | **Atributos** |
| --- | --- | --- |
| <script> | Coloca un script dentro del documento cuando está en head. |  |
|  |  | src: Especifica la localización de un script externo. |
|  |  | type: Especifica el lenguaje de scripts de los contenidos del elemento. |

Vamos a escribir JavaScript dentro de un elemento HTML con atributo id=”hackspace”:

<!DOCTYPE html>

<html>

<body>

<script>

function myFunction() {

document.getElementById("hackspace").innerHTML = "Hola JavaScript!";

}

</script>

<p id="hackspace">Antes de JavaScript</p>

<button type="button" onclick="myFunction()">Probar script</button>

</body>

</html>

Tablas, Listas, Links e Imágenes

| **Etiqueta** | **Descripción** | **Atributos** |
| --- | --- | --- |
| <table> | Define una tabla. |  |
| <td> | Define una columna. |  |
| <th> | Define las cabeceras. |  |
| <tr> | Define una fila. |  |
| <li> | Define un ítem de lista. |  |
| <ol> | Define una lista ordenada. |  |
| <ul> | Define una lista desordenada. |  |
| <a> | Define un link. |  |
|  |  | href: Url del link externo, id del link interno o email. |
|  |  | target: Define si el link se abrirá en la misma o en una nueva ventana. |
| <img> | Inserta una imagen. |  |
|  |  | src: Origen de la imagen. |
|  |  | align: Alinea la imagen dependiendo del valor que tome. |

Veamos el ejemplo con estas etiquetas:

<!DOCTYPE html>

<html>

<body>

<p><a href="hackspace.la" target="\_blank">Lista ordenada:</a></p>

<img src="http://hackspace.la/img/ejemplos/logos.jpeg">

<ol>

<li>Frontend</li>

<li>Backend</li>

<li>Electronica</li>

</ol>

<p><a href="mailito:contacto@hackspace.pe">Lista no ordenada:</a></p>

<ul>

<li>Frontend</li>

<li>Backend</li>

<li>Electronica</li>

</ul>

<br>

Asi se veria en una tabla:<br>

<table>

<tr>

<th>Nro</th>

<th>Curso</th>

</tr>

<tr>

<td>1</td>

<td>Frontend</td>

</tr>

<tr>

<td>2</td>

<td>Backend</td>

</tr>

<tr>

<td>3</td>

<td>Electronica</td>

</tr>

</table>

</body>

</html>

Así se debería ver:

Formularios

| **Etiqueta** | **Descripción** | **Atributos** |
| --- | --- | --- |
| <form> | Esta etiqueta indica que se empezara a recolectar información del usuario. |  |
| <input> | Genera un campo en el cual el usuario puede introducir información o contenido. |  |
|  |  | required: Define que el campo es obligatorio. |
|  |  | type: Tipo de elemento input. |
| <select> y<option> | Genera un input de lista desplegable. |  |
| <textarea> | Genera un input tipo texto multilínea. |  |
| **Atributo** | **Valor** | **Descripcion** |
| type | text | Define un input tipo texto. |
|  | password | Define un input tipo password. |
|  | radio button | Define un input tipo radio button. |
|  | checkbox | Define un input tipo checkbox. |
|  | file | Define un input tipo file. |
|  | submit button | Define un input tipo submit button. |
|  | image button | Define un input tipo image button. |
|  | label | Define un input tipo etiqueta. |
|  | date | Define un input tipo fecha. |
|  | email | Define un input tipo email. |
|  | url | Define un input tipo url. |

Veamos el ejemplo con estas etiquetas:

<!DOCTYPE html>

<html>

<body>

<form action="/pagina">

Elija un username: <input type="text" name="Username" value="HackSpace"><br>

Elija un password: <input type="password" id="Password" value="123456789"><br>

Nombres: <input type="text" name="FirstName" value="Juan"><br>

Apellidos: <input type="text" name="LastName" value="Perez"><br>

Foto de perfil: <input type="file" id="myFile"><br>

Pais:

<select>

<option value="peru">Peru</option>

<option value="arge">Argentina</option>

<option value="chil">Chile</option>

<option value="bras">Brasil</option>

</select><br>

Fecha de nacimiento: <input type="date" id="myDate" value="2015-01-05"><br>

E-mail: <input type="email" id="myEmail" value="contacto@hackspace.com"><br>

Pagina web:<input type="url" id="myURL" value="http://www.tuweb.com"><br>

Sobre mi:<br>

<textarea rows="4" cols="50">

Esta es mi descripcion.

</textarea><br>

<input type="checkbox" id="myCheck"> He leido el acuerdo de seguridad

<br>

<input type="submit" value="Enviar">

<input type="button" id="Cancel" onclick="cancelar()" value="Cancel">

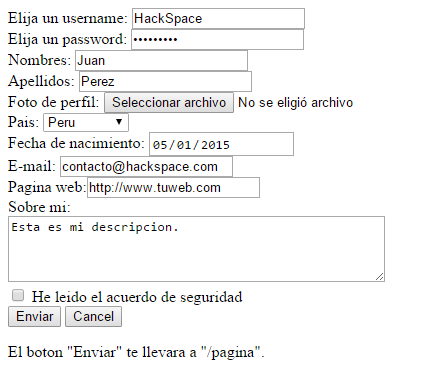
</form>

<p>El boton "Enviar" te llevara a "/pagina".</p>

</body>

</html>

Así se debería ver:

Otras etiquetas

| **Etiqueta** | **Descripción** | **Atributos** |
| --- | --- | --- |
| <audio> | Define un elemento audio. |  |
| <sourcet> | Define el origen del audio. |  |
|  |  | src: Es el url del audio. |
|  |  | type: Formato de audio. |

<!DOCTYPE html>

<html>

<body>

<audio controls>

<source src="http://www.w3schools.com/tags/horse.ogg" type="audio/ogg">

<source src="http://www.w3schools.com/tags/horse.mp3" type="audio/mpeg">

Tu browser no soporta el elemento audio.

</audio>

</body>

</html>

**Ejercicios**Hora de practicas HTML.

Antes de continuar, reforcemos lo que hemos aprendido en esta primera sección con unos cuántos ejercicios muy simples.

Ejercicio 1

* Escribe un documento HTML que tenga como título “Programación” y redacta brevemente acerca de programación. Incluye cabeceras, párrafos, divisiones, una tabla que enumere los cursos que desearías aprender.

Ejercicio 2

* Escribe un documento HTML con un formulario que pregunte por datos como nombre, apellido, cursos, notas, fecha de ingreso, un botón para subir una foto y un botón final para “enviar”.

### ****Teoría****HTML: Empezando un documento HTML y conociendo las etiquetas.

### Como agregar CSS al HTML

1. Agregar un link a un archivo de CSS externo:

<link rel="stylesheet" type="text/css" href="estilos.css" />

1. Agregar CSS directamente en la etiqueta <style> del HTML:

<style type="text/css">

p {

color:green

}

</style>

1. Agregar CSS en las etiquetas del HTML:

<p style="color:white;background:green;">Texto blanco fondo verde</p>

1. Importando un archivo CSS:

@import "estilos.css"

### Estructura básica

selector {

propiedad1:valor;

propiedad2:valor;

propiedad3:valor;

}

* El selector identifica los elementos del HTML a los cuales se aplicara la regla.
* Los pares propiedad-valor dentro de las llaves están separados por puntos y coma; y las propiedades están separadas de su respectivo valor por dos puntos.
* Las propiedades definen lo que quieres modificar a los elementos seleccionados, por ejemplo, cambiar la posición, el color, los márgenes, el color de fondo, entre otros.
* Los valores son los valores que quieres cambiar a cada propiedad de los elementos seleccionados. Los valores dependen de la propiedad, por ejemplo las propiedades que afectan al color pueden tomar nombres de color (en inglés) como red, gray, yellow, código del color en hexadecimal, como009900# o valores RGB, como rgb(34,142,28). Las propiedades que afectan el ancho, la altura, la posición, el margen, etc pueden tomar valores medidos en pixel, ems, porcentaje, centímetros u otras unidades.

##### **Ejemplo**

h1 {

color: red;

font-family: Arial;

font-size: 30px;

}

El selector h1 selecciona a todos los elementos <h1< en el documento HTML que usa este CSS para aplicar esta regla, a menos que existan selectores más específicos cuyas reglas sobrescribirán a esta.

Las propiedades en esta regla son el color del texto, la fuente del texto, y el tamaño de la fuente.

Los valores de estas propiedades son rojo, Arial, y 30 pixeles respectivamente.

### Selectores

Existen muchos tipos de selectores los tres más usados que vas a encontrar son:

1. Tag name: Selecciona los elementos del HTML de la respectiva etiqueta:

p {

color: red;

}

1. Selector por clase: Selecciona los elementos del HTML que tienen un atributo class con el valor especificado:

<h1 class="ejemplo">Texto de ejemplo</p>

<div class="ejemplo">Contenido</div>

.ejemplo{

color: blue;

}

1. Selector por id: Selecciona los elementos del HTML que tienen un atributo id con el valor especificado:

<h1 id="myid">Texto de ejemplo</p>

<div id="myid">Contenido</div>

#myid{

color: blue;

}

Puedes unir varios selectores para definir reglas más específicas, por ejemplo:

p.importante : {

color: red;

font-size: 30px;

}

En el ejemplo anterior selecciona todos los elementos del HTML dentro de la etiqueta <p< cuyo valor **class**es importante

### Agrupando selectores

Agrupando selectores

h1 {

color: blue;

font-size: 20px;

}

h2 {

color: blue;

font-size: 20px;

}

Donde los selectores **h1** y **h2** tienen el mismo estilo es posible agrupar los selectores separados por una coma:

h1,h2 {

color: blue;

font-size: 20px;

}

### Selectores universales

En CSS existe el selector universal \* el cual selecciona todos los elementos de un tipo el cual no es muy usado ya que raras veces se desea que todos los elementos tengan el mismo estilo.

\* {

margin: 0;

padding: 0;

}

### Selectores descendentes

En algunas ocasiones se requiere dar estilos diferentes a elementos de un solo tipo pero que se diferencia por el elemento que los contiene.

Un elemento A es descendiente de otro elemento B si el elemento A esta incluido en el elemento B-

<h1>

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

<p>Praesent ac risus sed magna ornare pulvinar. </p>

</h1>

<p>Integer ex tortor, eleifend non porttitor eget, dictum eu elit.</p>

h1 p {

color: red;

}

En el ejemplo anterior existen dos etiquetas <p< pero se aplica solamente a la primera ya que esta está contenida dentro de la etiqueta h1 como se especifica en el selector del CSS.

Hay que resaltar que no necesariamente el elemento descendiente tiene que estar contenido inmediatamente dentro del elemento externo, es decir el elemento descendiente puede estar contenido dentro de otro elemento que a su vez este contenido en el elemento más externo.

<h1>

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

<p>Praesent ac risus sed <strong>magna</strong> ornare pulvinar. </p>

</h1>

<strong>Integer ex tortor, eleifend non porttitor eget, dictum eu elit.</strong>

h1 span {

color: red;

}

En el ejemplo anterior la etiqueta <strong<> en la que deseamos aplicar el estilo está contenida dentro de una etiqueta <p< la cual a su vez está contenida en una etiqueta <h1< por lo cual cumple la regla del selector del CSS.

En general la estructura para seleccionar elementos descendientes es los selectores de cada elemento individual comenzando del más externo hasta el más interno separadas por un espacio:

selector1 selector2 selector3 {

...

}

**Ejercicios**Hora de practicas CSS.

Ejercicio 1

* Agrega una hoja de estilos externa al documento HTLM en el cual apliques los siguientes estilos a todos los párrafos <p< del documento:
  + Color: Azul
  + Tamaño de fuente: 12px
  + Tipo de fuente: Times New Roman

<!doctype html>

<html lang="es">

<head>

<meta charset="UTF-8"/>

<title>Document</title>

</head>

<body>

<h1>Bienvenido!</h1>

<a href="http://www.hackspace.la">Visita la pagina</a>

<p> Dolor sit amet, consectetur adipiscing elit. Praesent et tortor id turpis vestibulum vestibulum et quis magna. Praesent ac risus sed magna ornare pulvinar. Duis et aliquet nunc. In elit diam, elementum eu dictum quis, pharetra nec justo. Curabitur hendrerit fermentum nisl. Cras convallis gravida mollis. Nunc et feugiat sapien, eu ullamcorper lorem. </p>

<p> Mauris faucibus tristique posuere. Nam vehicula quam at venenatis imperdiet. Vivamus mattis tortor eget est mollis tempor. Quisque nec eros libero. Suspendisse ultrices facilisis mi, eget bibendum libero auctor in. </p>

</html>

Ejercicio 2

* Modificar el CSS para agregar los siguientes estilos al documento HTML:
  + El color del texto de todos los elementos del documento es verde.
  + Los elementos con id HS tienen un color de fondo negro.
  + Los elementos con clase CoreUp tienen borde solido de 2px y color de fondo con código rgbrgb(224,224,224).

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8"/>

<title>Document</title>

</head>

<body>

<h1>Bienvenido!</h1>

<a href="http://www.hackspace.la">Visita la pagina</a>

<p> Dolor sit amet, consectetur adipiscing elit. Praesent et tortor id turpis vestibulum vestibulum et quis magna. Praesent ac risus sed magna ornare pulvinar. Duis et aliquet nunc. In elit diam, elementum eu dictum quis, pharetra nec justo. Curabitur hendrerit fermentum nisl. Cras convallis gravida mollis. Nunc et feugiat sapien, eu ullamcorper lorem. </p>

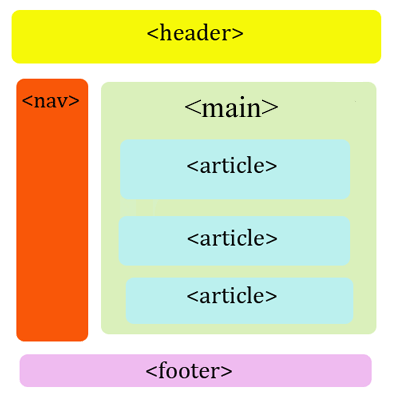
<p> Mauris faucibus tristique posuere. Nam vehicula quam at venenatis imperdiet. Vivamus mattis tortor eget est mollis tempor. Quisque nec eros libero. Suspendisse ultrices facilisis mi, eget bibendum libero auctor in. </p>

</html>

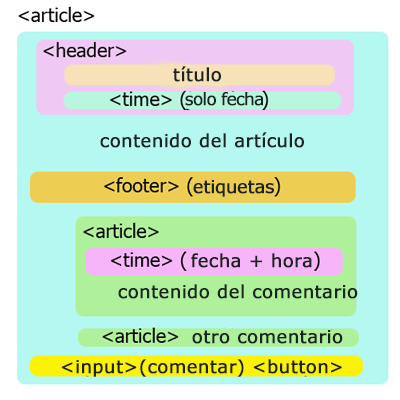
**Reto 1**Primer reto del entrenamiento de Frontend.

Reto 1 de Frontend

Este 1er reto de Frontend se trata de maquetación. No queremos que te compliques mucho con el diseño en CSS, pues para eso tendrías que tener una línea gráfica definida. Utiliza el CSS y el HTML para poder crear una estructura como la que te presentamos a continuación (las etiquetas son de HTML5):



Esta maquetación es de una sola página de un blog. Además dentro de cada <article>, debes de incluir la siguiente estructura:



No te confundas, no son dos páginas diferentes. La segunda imagen es solo una descripción más a fondo de lo que tiene que tener cada <article> dentro de <main>

Te recomendamos, para no empezar desde cero, descargarte cualquiera de las 3 opciones que encuentras en el siguiente link: <http://www.initializr.com/>

No te hemos mencionado aún sobre esta herramienta, así que cuando ingreses a la web te recomendamos descargarte la opción Classic H5BP para que no te confundan los CSS predefinidos más complejos que pueden tener las otras dos opciones. Aunque si ya tienes alguna noción sobre Twitter Bootstrap, eres libre de utilizarlo ya que te facilitará mucho más el trabajar con la maquetación y el CSS.

Finalmente, no te preocupes por el diseño que puedas ponerle a la página, por ahora solo nos interesa que cumpla con la estructura que te mencionamos.

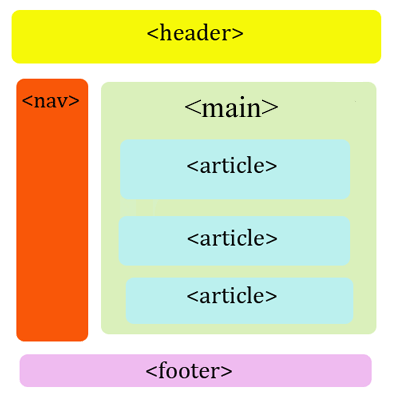
* PD1: Es recomendable que los anchos los trabajes con porcentajes y no con medidas fijas de pixeles.
* PD2: Si se te hace muy complicado el descargar la plantilla en Initializer, puedes descargarte [este repositorio](https://github.com/HackSpacePeru/reto-1-frontend) pues ya lo tenemos descargado para ti.
* PD3: Empieza a editar el index.html que te brinda Initializer, justamente una de sus funciones es brindarte una plantilla del HTML y CSS predefinido, para que te concentres en empezar a maquetear únicamente.

#### Semana 2 : Jquery

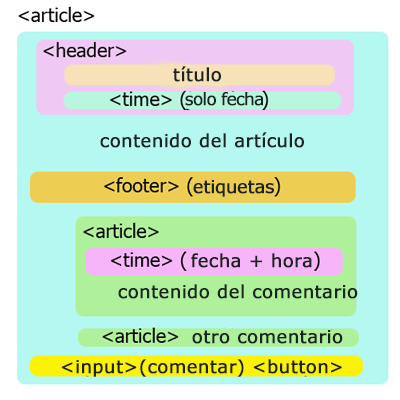
**Ejemplo**¿No entendiste el reto 1? Acá te ayudamos.

Recordando

En el 1er reto teníamos que estructurar en HTML la siguiente página:



Junto con la siguiente estructura dentro de la sección <article>.



Descárgate el siguiente ejemplo y verás lo mínimo que esperamos que presentes (ya que acá está la respuesta, deberías agregarle ahora un poco más de personalidad).

[F-Ejemplo-1.1.zip ( 68 KB )](http://hackspace.la/assets/ejemplos/F-Ejemplo-2.1.zip)

Ahora, a continuar con el material de la 2da semana.

### ****Teoría****Empezando a utilizar Jquery.

### Antes de empezar:

Jquert es una librería de JavaScript. No es un lenguaje de programación. Simplifica grandemente la programación con JavaScript y es fácil de aprender.

Para importar la librería incluir la siguiente ruta dentro de la etiqueta <script>:

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

Y es recomendable colocarla al final de <body>.

Durante esta semana, tendrás bastantes ejemplos en código que pondremos listo para que lo copies en Sublime Text y lo pruebes en tu navegados.

### Selectores

La función $() recibe como parámetro una etiqueta <html> y devuelve todos los elementos (nodos) que concuerden con la expresión, para luego poder ejecutar una acción (action) $(selector).action(). Esta expresión es denominada selector y usa la sintaxis CSS para seleccionar elementos.

$("p").hide() : El método hide() oculta elementos, en este caso elementos.

$("#test").hide() : Ocultamos elementos con el atributo id="test".

$(".test").hide() : Ocultamos elementos con el atributo class="test".

$(this).hide() : Ocultamos el actual elemento html.

#### Ejemplo

Empecemos con el primer ejemplo: Copia el siguiente código en Sublime Text, guardalo con la extensión.html y ábrelo en Google Chrome o cualquier navegador que tengas para probarlo.

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("span").hide();

$("#test").hide();

$(".test").hide();

$(this).hide();

});

});

</script>

</head>

<body>

<h2>Yo no desaparezco.</h2>

<span>Estoy dentro de la etiqueta span</span>

<p>Desapareceré por la etiqueta p. :(</p>

<p id="test">Desapareceré por el id "test". :(</p>

<p class="test">Desapareceré por la clase "test". :(</p>

<button>Hazme clic. Desapareceré por ser button. :(</button>

</body>

</html>

[Conoce más aquí.](http://api.jquery.com/category/selectors/)

### Eventos jQuery

click() : Define un evento al momento de hacer click.

dblclick() : Define evento al momento de hacer doble click

mouseenter() : Define evento al momento en que el puntero del mouse ingresa a la ubicacion del elemento.

mouseleave() : Define evento al momento en que el puntero del mouse deja la ubicacion del elemento.

mousedown() : Define evento al momento en que el puntero del hace click en la ubicacion del elemento sin soltar.

mouseup() : Define evento al momento en que el puntero suelta despues de hacer click en la ubicacion del elemento.

hover() : Permite definir los eventos mouseenter y mouseleave en una sola funcion.

focus() y blur() : El evento focus se activa cuando se ingresa con el puntero del mouse a un campo del formulario y el evento blur se activa cuando se deja el campo.

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("h1").click(function(){

$(this).hide();

});

$("p").dblclick(function(){

$(this).hide();

});

$("#p1").mouseenter(function(){

alert("Has entrado a p1!");

});

$("input").focus(function(){

$(this).css("background-color","#cccccc");

});

$("input").blur(function(){

$(this).css("background-color","#ffffff");

});

});

</script>

</head>

<body>

<h1>Si me haces clic, desaparezco.</h1>

<p>Si me hace doble clic, desaparezco.</p>

<p id="p1">Pasa el mouse por este párafo</p><br>

Nombre: <input type="text" name="fullname"><br>

Email: <input type="text" name="email">

</body>

</html>

[Conoce más aquí.](http://api.jquery.com/category/events/)

### ****Ejemplos****Empecemos a realizar algunas acciones con los elementos de nuestra web.

### Mostrar y Ocultar elementos

hide() : Método para ocultar un elemento.

show() : Método para mostrar un elemento.

toggle() : Alterna entre los eventos hide() y show().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#hide").click(function(){

$(".p1").hide();

});

$("#show").click(function(){

$(".p1").show();

});

});

</script>

</head>

<body>

<p class="p1">Si clickeas en el boton "Ocultar", Yo desaparecere.</p>

<button id="hide">Ocultar</button>

<button id="show">Mostrar</button>

</body>

</html>

### Control de opacidad

fadeIn() : Método para descolorar un elemento.

fadeOut() : Método para colorear un elemento.

fadeToggle() : Alterna entre los métodos fadeIn() y fadeOut().

fadeTo() : Método para controlar la opacidad de un elemento siguiendo un parametro de referencia.

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#div1").fadeIn();

$("#div2").fadeOut("slow");

$("#div3").fadeToggle(3000);

$("#div4").fadeTo("slow",0.7);

});

});

</script>

</head>

<body>

<p>Demostrar fadeIn,fadeOut,fadeToggle y fadeTo con diferentes parametros.</p>

<button>Click para probar fadeIn() en las cajas.</button>

<br><br>

<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>

<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>

<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>

<div id="div4" style="width:80px;height:80px;display:none;background-color:pink;"></div>

</body>

</html>

### Deslizar elementos

slideDown() : Método para desliza hacia abajo un elemento.

slideUp() : Método para deslizar hacia arriba un elemento.

slideToggle() : Alterna entre los métodos slideDown() y slideUp().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#flip").click(function(){

$("#panel").slideToggle("slow");

});

});

</script>

<style>

#panel,#flip

{

padding:5px;

text-align:center;

background-color:#e5eecc;

border:solid 1px #c3c3c3;

}

#panel

{

padding:50px;

display:none;

}

</style>

</head>

<body>

<div id="flip">Click para aplicar slide up y slide down al panel</div>

<div id="panel">Hola mundo</div>

</body>

</html>

### Animar elementos

Utilizamos animate().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

var div=$("div");

div.animate({height:'300px',opacity:'0.4'},"slow");

div.animate({width:'300px',opacity:'0.8'},"slow");

div.animate({height:'100px',opacity:'0.4'},"slow");

div.animate({width:'100px',opacity:'0.8'},"slow");

});

});

</script>

</head>

<body>

<button>Empezar Animacion</button>

<p>Por default, los elementos HTML tienen una posicion estatica, y no pueden ser movidos. Para manipularl la posicion, recuerda primero configurar la propiedad de posicion CSS de un elemento a relativo, fijo o absoluto.</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;">

</div>

</body>

</html>

### Detener Animaciones

stop() : detiene una animacion en curso.

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#start").click(function(){

$("div").animate({left:'100px'},5000);

$("div").animate({fontSize:'3em'},5000);

});

$("#stop").click(function(){

$("div").stop();

});

$("#stop2").click(function(){

$("div").stop(true);

});

$("#stop3").click(function(){

$("div").stop(true,true);

});

});

</script>

</head>

<body>

<button id="start">Empezar</button>

<button id="stop">Detener</button>

<button id="stop2">Detener todo</button>

<button id="stop3">Detener y Terminar</button>

<p>El boton "Empezar" empieza la animacion.</p>

<p>El boton "Detener" detiene la animacion activa actual, pero permite que pueda continuar despues.</p>

<p>El boton "Detener todo" detiene la actual animacion activa y borra la cola de animacion; por lo que todas las animaciones en el elemento son detenida .</p>

<p>El boton "Detener y Terminar" apresura a través de la corriente de animación activa, luego se detiene.</p>

<div style="background:#98bf21;height:100px;width:200px;position:absolute;">HELLO</div>

</body>

</html>

### ****Ejemplos****Trabajemos con el contenido de nuestra web.

### Obtener contenido y atributos

text() y html() : Obtener contenidos con los métodos text() y html().

attr() : Obtener el valor de un atributo con el método attr().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#btn1").click(function(){

alert("Text: " + $("#test").text());

});

$("#btn2").click(function(){

alert("HTML: " + $("#test").html());

});

$("#button").click(function(){

alert($("#hsla").attr("href"));

alert("Valor: " + $("#test").val());

});

});

</script>

</head>

<body>

<p id="test">Esto es letra <b>negrita</b> en un parrafo.</p><br>

<input input type="button" id="btn1" value="Mostrar texto." /><br>

<input input type="button" id="btn2" value="Mostrar HTML." /><br>

<p>Name: <input type="text" id="test" value="Hackspace"></p><br>

<p><a href="http://www.hackspace.la" id="hsla">hackspace.la</a></p><br>

<input input type="button" id="button" value="Mostrar valor" />

</body>

</html>

### Control de opacidad

Configurar contenido con los métodos jQuery text(), html() y val().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("#btn1").click(function(){

$("#test1").text("Hola mundo texto!");

});

$("#btn2").click(function(){

$("#test2").html("<b>Hola mundo html!</b>");

});

$("#btn3").click(function(){

$("#test3").val("Hola mundo valores!");

});

});

</script>

</head>

<body>

<p id="test1">Este es un parrafo.</p>

<p id="test2">Este es otro parrafo.</p>

<p>Campo input: <input type="text" id="test3" value="Hackspace"></p>

<button id="btn1">Configurar texto</button>

<button id="btn2">Configurar HTML</button>

<button id="btn3">Configurar valores</button>

</body>

</html>

Configurar valores de atributos usando la funcion callback (funcion parametro de otra funcion) dentro del método attr().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("#hsla").attr("href", function(i,origValue){

return origValue + "/jquery";

});

});

});

</script>

</head>

<body>

<p><a href="http://www.hackspace.la" id="hsla">Hackspace Core Upgrade 2015</a></p>

<button>Cambiar el valor href</button>

<p>Mouse sobre el link (o click en el) para ver que el valor del atributo href ha cambiado.</p>

</body>

</html>

### Agregar o Eliminar un elemento HTML

Usaremos los métodos append() y remove() , para realizar esto.

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

function appendText() {

var txt1 = "<p>Text.</p>"; // Crear texto con HTML

var txt2 = $("<p></p>").text("Text."); // Crear texto con jQuery

var txt3 = document.createElement("p");

txt3.innerHTML = "Text."; // Crear texto con DOM

$("body").append(txt1, txt2, txt3); // Agregar txt1,txt2 y txt3

$("#div1").remove(); // Eliminar div1

}

</script>

</head>

<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

Texto dentro del div1.

<p>Dentro del div1.</p>

<p>Dentro del div1.</p>

</div>

<p>Fuera del div1.</p>

<button onclick="appendText()">Append text</button>

</body>

</html>

### Animar elementos

Utilizamos animate().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

var div=$("div");

div.animate({height:'300px',opacity:'0.4'},"slow");

div.animate({width:'300px',opacity:'0.8'},"slow");

div.animate({height:'100px',opacity:'0.4'},"slow");

div.animate({width:'100px',opacity:'0.8'},"slow");

});

});

</script>

</head>

<body>

<button>Empezar Animacion</button>

<p>Por default, los elementos HTML tienen una posicion estatica, y no pueden ser movidos. Para manipularl la posicion, recuerda primero configurar la propiedad de posicion CSS de un elemento a relativo, fijo o absoluto.</p>

<div style="background:#98bf21;height:100px;width:100px;position:absolute;">

</div>

</body>

</html>

### Configurar CSS

Para esto usaremos el método css().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").css({"background-color":"yellow","font-size":"200%"});

});

});

</script>

</head>

<body>

<h2>Esta es una cabecera</h2>

<p style="background-color:#ff0000">Este es un parrafo.</p>

<p style="background-color:#00ff00">Este es un parrafo.</p>

<p style="background-color:#0000ff">Este es un parrafo.</p>

<p>Este es un parrafo.</p>

<button>Configurar estilos</button>

</body>

</html>

### Configurar CSS de elemento padre e hijo

Para esto usaremos parent() y children().

#### Ejemplo

<!DOCTYPE html>

<html>

<head>

<style>

.descendants \*

{

display: block;

border: 2px solid lightgrey;

color: lightgrey;

padding: 5px;

margin: 15px;

}

</style>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

<script>

$(document).ready(function(){

$(".descendants").children().css({"color":"red","border":"2px solid red"});

$("p").parent().css({"color":"red","border":"2px solid pink"});

$("p").children().css({"color":"red","border":"2px solid green"});

});

</script>

</head>

<body>

<div class="grandparent" style="width:800px">div abuelo

<div class="descendants" style="width:500px;">div padre

<p>p hijo

<span>span nieto</span>

</p>

<p>p hijo

<span>span nieto</span>

</p>

</div>

</div>

</body>

</html>

**Ejercicios**Hora de practicar Jquery.

Pero antes de continuar, reforcemos lo que hemos aprendido en esta primera sección con unos cuántos ejercicios muy simples.

Ejercicio 1

En 5 segundos aumentar de tamaño a un texto **"TEST"** y luego desaparecerlo al pasar el mouse encima del texto. Esto usando animate() y modificando el css apartir de eso.

Ejercicio 2

Crear un programa con jQuery para obtener un menú, tal que cada vez que el usuario introduce el puntero el item cambia de color.

### ****Reto 2****Segundo reto del entrenamiento de Frontend.

### Reto 2 de Frontend

Este es un reto muy simple y dejamos el contenido a tu criterio (Además, seguro el reto de la semana pasada te debió consumir bastante).

A continuación tenemos un tutorial (en inglés) de codrops. Esta es una web que de todas maneras vas a tener que ir revisando para ver ejemplos y tutoriales de cosas muy interesant

Entra al siguiente link: <http://tympanus.net/codrops/2014/09/23/animated-background-headers/>

Ahí podrás ver el demo y descargarte el código fuente para utilizarlo donde desees. Queremos que lo utilices en una página web. Subas tu reto resuelto nuevamente a tu github, pero esta vez te pediremos una cosa más: Crea tu Github Page. Esto lo consigues clonando la rama master donde normalmente se sube toda tu página web y llamando gh-pages a esa nueva rama.

[Acá encuentras un pequeño tutorial para crear tu **Github Page**.](http://albertoromeu.com/como-publicar-tu-aplicacion-web-en-github-con-gh-pages/)

#### Semana 3 : Materialize

### ****Teoría****Conozcamos al Material Design.

### ¿Qué es el Material Design?

Material Design es un concepto, una filosofía, unas pautas enfocadas al diseño utilizado en Android, pero también en la web y en cualquier plataforma.

Creado y diseñado por Google, Material Design combina los principios clásicos del diseño exitoso junto con la innovación y la tecnología. El objetivo de Google consiste en desarrollar un sistema de diseño que permite una experiencia de usuario unificada a través de todos sus productos en cualquier plataforma.

Puedes encontrar más información sobre Material Design en el siguiente link:<http://www.elandroidelibre.com/2014/11/que-es-material-design.html>

A continuación un ejemplo gráfico de lo que es un ejemplo de lo que es Material Design

### ****Teoría****Empecemos a trabajar con Materialize.

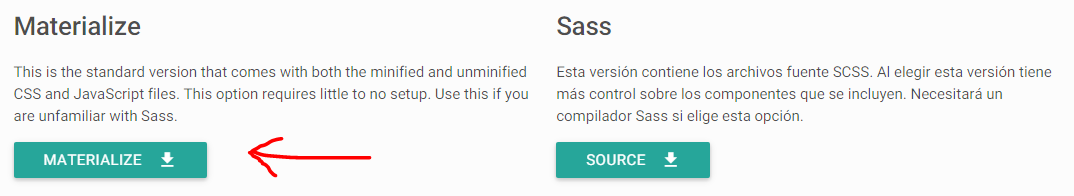
### ¿Qué es un Framework para Frontend?

Un framework frontend es un conjunto de recursos y lineamientos para maquetear y diseñar una web. Normalmente comprende una serie de estilos css, íconos, fuentes, colores, animaciones y funciones javascript para hacernos la vida más fácil en lo que se refiere a maquetear y diseñar una web rápidamente.

### Empezando con Materialize

Materialize es uno de los frameworks de frontend más conocidos que utiliza Material Design. Antes de empezar, debemos descargarnos los archivos necesarios. Estos los podemos encontrar en la misma página de Materialize, el link es el siguiente:

<http://materializecss.com/getting-started.html>



Al extraer el archivo encontrarás un directorio como el siguiente:

materialize/

|--css/

| |--materialize.css

| |--materialize.min.css

|

|--font/

| |--material-design-icons/

| |--roboto/

|

|--js/

| |--materialize.js

| |--materialize.min.js

|

|--index.html

#### Configurando el HTML

A continuación, hay que crear el archivo index.html dentro de la carpeta materialize y nos aseguramos de enlazar los archivos correctamente. Generalmente es aconsejable importar archivos de javascript al final de la etiqueta body para reducir tiempos de carga. Sigue el siguiente ejemplo sobre cómo importar Materialize en una página web.

¡Una última cosa a tener en cuenta es que tiene que importar jQuery antes de importar materialize.js!

<!DOCTYPE html>

<html>

<head>

<!--Importa la Fuente de Google Icons-->

<link href="http://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<!--Importa materialize.css-->

<link type="text/css" rel="stylesheet" href="css/materialize.min.css" media="screen,projection"/>

<!--Haz saberle al navegador que la página está optimizada para móviles-->

<meta name="viewport" content="width=device-width, initial-scale=1.0"/>

</head>

<body>

<!--Importa jQuery antes de materialize.js-->

<script type="text/javascript" src="https://code.jquery.com/jquery-2.1.1.min.js"></script>

<script type="text/javascript" src="js/materialize.min.js"></script>

</body>

</html>

### Documentación

Todo lo que vamos a necesitar para aprender Materialize lo encontraremos en su página web:<http://materializecss.com/>

Sin embargo, no todo está traducido, así que realizaremos algunos ejemplos con las secciones más importantes que necesitas aprender de ella.

### ****Ejemplo****Conociendo el Grid System de Materialize.

### Documentación

La documentación específica para esta sección la encuentras en: <http://materializecss.com/grid.html>

### Grid System

Su traducción literal sería Sistema de Grillas, pero mejor no tratemos de traducirlo. Específicamente en Materialize se utiliza el tradicional sistema de 12 columnas, en el que, sin importar el ancho de la pantalla, se considera que esta se divide en 12 columnas. De esta manera, podemos agrupar y maquetear toda la web considerando distribuciones y agrupaciones de estas columnas.

Este es el código básico con el que empezaríamos a trabajar una web en Materialize:

<body>

<div class="container">

<div class="row">

Acá van las columnas, dentro de row.

</div>

</div>

</body>

Si revisas el link de la documentación del grid system encontrarás que tenemos dos opciones para empezar a trabajar con este:

Este es una columna sin .container dentro del div inicial:



<body>

<div>

<div class="row">

<div class="col s12"></div>

</div>

</div>

</body>

Este es una columna con .container dentro del div inicial



<body>

<div class="container">

<div class="row">

<div class="col s12"></div>

</div>

</div>

</body>

#### Creando Columnas

Para crear una columna y definir su tamaño en un div, debemos utilizar uno de los siguiente **Prefijos de Clases** junto con el número de columnas que deseemos utilizar:

|  | **Dispositivos Móviles  <= 600px** | **Tablets  <= 992px** | **Dispositivos de Escritorio  > 992px** |
| --- | --- | --- | --- |
| **Prefijo de Clase** | .s | .m | .l |
| **Ancho de Contenedor** | 85% | 85% | 70% |
| **Número de Columnas** | 12 | 12 | 12 |

Descarga el ejemplo aquí:

[F-Ejemplo-3.1.zip ( 1,347 KB )](http://hackspace.la/assets/ejemplos/F-Ejemplo-3.1.zip)

Dentro de la **Sección Central** del index.htmk del ejemplo que te acabas de descargar encontrarás el siguiente código:

<!-- Sección Central -->

<div class="container">

<div class="row">

<div class="col s12 m4">

<div class="icon-block">

<h2 class="center light-blue-text"><i class="material-icons">flash\_on</i></h2>

<h5 class="center">Agiliza el Desarrollo</h5>

<p class="light">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse

cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

</div>

</div>

<div class="col s12 m4">

<div class="icon-block">

<h2 class="center light-blue-text"><i class="material-icons">group</i></h2>

<h5 class="center">Experiencia Enfocada en el Usuario</h5>

<p class="light">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse

cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

</div>

</div>

<div class="col s12 m4">

<div class="icon-block">

<h2 class="center light-blue-text"><i class="material-icons">settings</i></h2>

<h5 class="center">Fácil Para Trabajar</h5>

<p class="light">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,

quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse

cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

</div>

</div>

</div>

</div>

Verás que los <div class="col s12 m4"> las clases **s12** y **m4** indican que cuando la pantalla es pequeña o**s**mall (<= 600px) el div ocupará las 12 columnas de la pantalla. Pero, si la pantalla es **m**ediana (<= 992px), el div ocupará únicamente 4 columnas. Además, como no se está usando ninguna propiedad para pantallas grandes o **l**arge, para pantallas de un tamaño > 992px, se seguirán utilizando las propiedades para pantallas **m**edianas

.

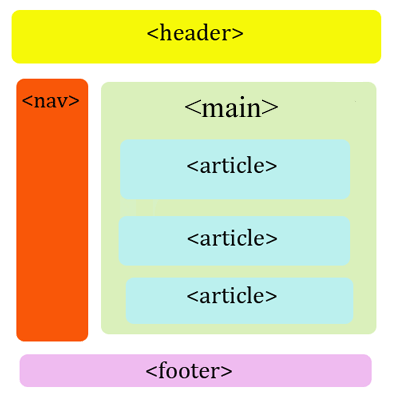
Este es un primer ejemplo bastante simple para que puedas ir entendiendo cómo se usa el Grid System de 12 columnas de Materialize.

A continuación un video para que pueda profundizar un poco más:

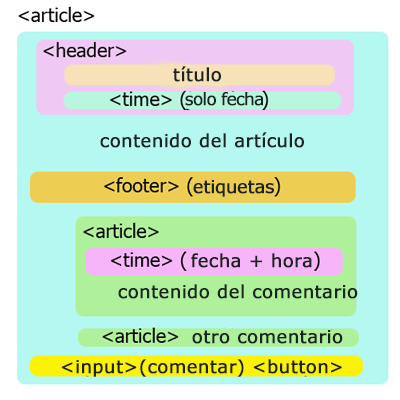
**Reto 3**Tercer reto del entrenamiento de Frontend.

Reto 3 de Frontend

¿Recuerdas el reto de la semana 1? Pues ahora tendrás que hacer la misma estructura de la web, pero ahora utilizando la mayor cantidad de componentes que puedas de Materialize. No es necesario que utilices las mismas etiquetas de HTML5 que hemos puesto, pero sí que te guíes de la distribución (no uses esos colores que están horribles).



Esta maquetación, como recordarás, es de una sola página de un blog. Además dentro de cada <article>, debes de incluir la siguiente estructura:



Ahora sí preocúpate un poco más por el diseño y que quede bonito.

A continuación te volvemos a poner el ejemplo que hicimos para el reto 1, pare que tengas una mejor noción sobre la distribución:

[F-Ejemplo-1.1.zip ( 68 KB )](http://hackspace.la/assets/ejemplos/F-Ejemplo-2.1.zip)

### ****Ejemplo****Aprendiendo a Utilizar los Compontentes de Javascript en Materialize.

### Continuamos con los Ejemplos de la Semana 3

De la misma manera que hicimos con la semana 3. A continuación tendrás un grupo de videos para que puedas profundizar cada elemento de Javascript de Materialize.

#### Modal

¿Recuerdas los popups? Un modal es una sección de contenido que aparece delante de la página. Es un recurso muy utilizado y es muy recomendable el aprender a utilizarlo.

Documentación y ejemplos: <http://materializecss.com/modals.html>

#### Collapsible

Esta función nos permite utilizar el efecto acordeón.

Documentación y ejemplos: <http://materializecss.com/collapsible.html>

#### Tabs

Este componente nos ayuda a crear pestañas dentro de nuestra página.

Documentación y ejemplos: <http://materializecss.com/tabs.html>

#### Media

Estas funciones nos permiten trabajar con imágenes grandes o presentaciones de estas en estilo carrusel.

Documentación y ejemplos: <http://materializecss.com/media.html>

#### Waves

Esta función le da un efecto de ondas cuando presionamos un botón.

Documentación y ejemplos: <http://materializecss.com/waves.html>

#### Scroll Spy

Esta función nos ayuda a saber en qué sección de la página estamos, además de poder ejecutar funciones cuando esté en alguna sección en especial.

Documentación y ejemplos: <http://materializecss.com/scrollspy.html>

**Reto Final**Reto Final del Entrenamiento de Frontend.

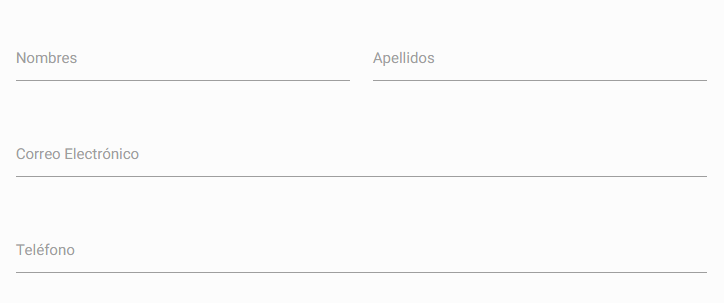
Reto Final de Frontend

Como verás, todos los retos tienen relación entre sí. Es así que este reto, se trata de agregarle unas cuántas funciones más al [reto de la semana 3](http://hackspace.la/semana4/semana3/reto3.html).

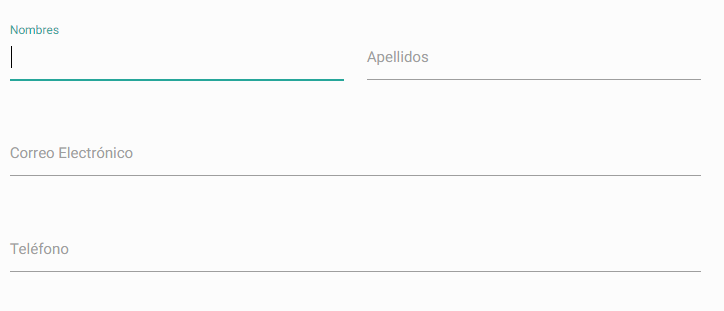
No será tan complicado, solo te pedimos que en cualquier sección de la página agregues un [botón de materialize](http://materializecss.com/buttons.html) que diga **Suscríbete** y que cuando hagas clic en ese botón aparezca un [modal](http://materializecss.com/modals.html).

Pero ahí no termina todo, dentro del [modal](http://materializecss.com/modals.html) tiene que haber un [formulario de materialize](http://materializecss.com/modals.html)

Así se debería de ver el formulario:



Así se debe ver el formulario cuando hacemos clic en el campo **Nombres**:



Y así se debería ver el **Modal**:

