**Elementos en HTML5**

|  |  |
| --- | --- |
| **Etiqueta** | **Uso y Ejemplo** |
| **Nuevos elementos semánticos en HTML5**  **Seccionamiento del contenido de una web** | **Usos principales para cada una de estas etiquetas semánticas:**   * **Header:** representa un grupo de contenidos introductorios o de navegación. Puede contener algunos elementos de encabezado, así como también un logo, un formulario de búsqueda, un nombre de autor y otros componentes. * **Main:** representa el contenido principal del <body> de un documento o aplicación. El área principal del contenido consiste en el contenido que está directamente relacionado, o se expande sobre el tema central de un documento o la funcionalidad central de una aplicación. Este contenido debe ser único al documento, excluyendo cualquier contenido que se repita a través de un conjunto de documentos como barras laterales, enlaces de navegación, información de derechos de autor, logos del sitio y formularios de búsqueda (a menos, claro, que la función principal del documento sea un formulario de búsqueda). * **Footer:** representa un pie de página. Un pie de página típicamente contiene información acerca del autor de la sección, datos de derechos de autor o enlaces a documentos relacionados. * **Article:** podría ser un mensaje en un foro, un artículo de una revista o un periódico, una entrada de blog, un comentario de un usuario, un widget interactivo o gadget, o cualquier otro elemento independiente del contenido. * **Section:** representa una sección genérica de un documento. Sirve para determinar qué contenido corresponde a qué parte de un esquema. Piensa en el esquema como en el índice de contenido de un libro; un tema común y subsecciones relacionadas. * **Aside:** suele utilizarse para barras laterales en las que se puede colocar publicidad, por ejemplo.   Aquí tenemos mucha más información asociada a estos elementos que permiten seccionar el contenido de una página web: [aquí](https://www.sitepoint.com/defining-the-sample-sites-page-structure/) y [aquí.](https://developer.mozilla.org/es/docs/Web/HTML/Elemento#Seccionamiento_del_contenido)  **Importante: diferencias entre article y section**  [Extraído de StackOverflow](https://es.stackoverflow.com/questions/42441/cual-es-la-diferencia-entre-section-y-article-en-html5)  La etiqueta <section>:  La etiqueta sección define las secciones de un documento, como capítulos, encabezados, pies de página o cualquier otra sección del documento.  De acuerdo con la documentación HTML5 del W3C:  "Una sección es un agrupamiento temático de contenido, normalmente con un encabezado".  Ejemplo:  <section>  <h1>WWF</h1>  <p>The World Wide Fund for Nature (WWF) is....</p>  </section>  La etiqueta <article>:  La etiqueta del article especifica contenido independiente y autónomo.  Un artículo debe tener sentido por sí mismo y debería ser posible Distribuirlo independientemente del resto del sitio.  Fuentes potenciales para el elemento artículo:   * Posts de foro * Blog post * Noticias * Comentarios   Ejemplo:  <article>  <h1>What Does WWF Do?</h1>  <p>WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.</p>  </article>  ¿Anidamiento <article> en <section> o viceversa?  El elemento <article> especifica un contenido independiente y autónomo.  El elemento <section> define la sección en un documento.  ¿Podemos utilizar las definiciones para decidir cómo anidar esos elementos? ¡No podemos!  Así, en Internet, encontrará páginas HTML con elementos <section> que contienen elementos <article> y <article> que contienen elementos <section>.  También encontrará páginas con elementos que contienen elementos <section> y elementos <article> que contienen elementos <article>.  Ejemplo para un periódico: Los artículos deportivos en la sección de deporte, pueden tener una sección técnica en cada artículo.  Básicamente estas etiquetas son *elementos semánticos* y la única diferencia real entre ellos es la legibilidad y la preferencia de diseño. |
| **Comentarios en HTML** | Uso: los comentarios se utilizan para dejar anotaciones en el código y permitir que sea entendible. Esto facilita el mantenimiento por parte de personas que no han realizado tareas en el desarrollo del código.  Ejemplo:  <!--Esto es un comentario en HTML--> |
| **<h1>**  **.**  **.**  **.**  **<h6>** | Uso: los encabezados sirven para titular secciones de la página. Van ordenados de mayor a menor importancia (y de mayor a menor tamaño), donde el encabezado h1 es el principal en la página. Su contenido debe ir en consonancia al tema de la web.  Ejemplo:  <h1>Este es el encabezado principal de la página</h1> |
| **<a>** | Uso: Sirven para enlazar con otros documentos o recursos  Ejemplo:  <span>Haz clic <a href=”http://iesmariamoliner.centros.educa.jcyl.es/sitio/”>aquí</a> para ir a la web del centro</span>  **Ampliación de enlaces:**  **Concepto de ancla (Permite saltar a una sección de la misma página, es decir, proporciona una manera de realizar enlaces internos).**  <p><a href="#aqui">Saltar hasta el ancla que hemos fijado</a></p>  <h2 id="aqui">Destino del enlace</h2>  <p>En enlace nos llevará hasta aquí</p>  **Lista de enlaces**  <ul>  <li><a href="http://www.google.com/ ">Google</a></li>  <li><a href="http://www.bing.com/ ">Bing</a></li>  <li><a href="http://www.search.yahoo.com/">Yahoo</a></li>  <li><a href="http://www.yandex.ru/ ">Yandex</a></li>  </ul> |
| **<p>** | Uso: Es un bloque de texto básico. El párrafo ignora los saltos de línea que apliquemos dentro de él y se trata de un elemento de bloque que ocupará automáticamente todo el ancho disponible. También se aplica un margen de manera automática al texto.  Ejemplo:  <p>Esto es un párrafo en el que aún no vamos a aplicar un estilo css</p> |
| **<span>** | Uso: se trata de un elemento en línea (in line). Se suele utilizar para dar estilos a palabras o conjunto de palabras dentro de un párrafo, aunque tiene muchas más aplicaciones.  Ejemplo:  <span>El elemento span es un elemento en línea</span> |
| **Elementos en línea y elementos de bloque** | Este concepto es fundamental para entender el posicionamiento de los elementos HTML.  Los elementos en línea sólo ocupan el espacio que necesite su contenido.  Los elementos de bloque ocupan toda la línea, del extremo izquierdo al extremo derecho del navegador, es decir, ocupan todo el ancho disponible.  Cuando empecemos con los estilos CSS profundizaremos más en estos conceptos.  **Algunos ejemplos de elementos en línea (inline)**  <a>  <abbr>  <acronym>  <b>  <bdo>  <big>  <br>  <button>  <cite>  <code>  <dfn>  <em>  <i>  <img>  <input>  <kbd>  <label>  <map>  <object>  <output>  <q>  <samp>  <script>  <select>  <small>  <span>  <strong>  <sub>  <sup>  <textarea>  <time>  <tt>  <var>  **Algunos ejemplos de elementos de bloque (block)**  <address>  <article>  <aside>  <blockquote>  <canvas>  <dd>  <div>  <dl>  <dt>  <fieldset>  <figcaption>  <figure>  <footer>  <form>  <h1>  <h6>  <header>  <hr>  <li>  <main>  <nav>  <noscript>  <ol>  <p>  <pre>  <section>  <table>  <tfoot>  <ul>  <video> |
| **<div>** | Uso: se trata del elemento de bloque por excelencia. Se utiliza para realizar distribuciones de los elementos (layouts) en una página web, normalmente como contenedor de otros elementos. Este elemento cobrará sentido cuando empecemos con el temario dedicado a estilos.  Ejemplo:  <div>Esto es una capa</div> |
| **<table>** | Uso: Es un elemento que sólo se debe utilizar para mostrar información, no para maquetar una página.  Tienen varios elementos:   * <tr> Que representan las filas * <td> Representan las celdas de cada fila * Opcional: <thead>Agrupa las filas de la cabecera. En este caso se utiliza <th> en vez de <td> * Opcional: <tbody> Agrupan las filas de datos. En este caso se utilizan <td> * Opcional: <tfoot> Agrupa las filas del pie de tabla. Se pueden utilizar tanto etiquetas <th> como <td>. * Atributo colspan en las celdas: sirve para combinar columnas. * Atributo rowspan en las celdas: sirve para combinar filas.   El elemento caption es el encargado de darle un título descriptivo a las tablas.  Los estilos sobre los bordes cobrarán sentido cuando empecemos a trabajar con CSS.  Vamos a ver un pequeño adelanto:  td, th {  border: 1px solid #999;  padding: 0.5rem;  }  Ejemplo: es fundamental estructurar el contenido de la tabla con las etiquetas thead, tbody y tfooter.  <table style=”border:1px solid black;”>  <caption>Tabla de ejemplo</caption>  <thead>  <tr>  <th>Nombre:</th>  <th>Descripción:</th>  </tr>  </thead>  <tbody>  <tr>  <td>HTML</td>  <td>Lenguaje de marcas para la elaboración de páginas web</td>  </tr>  <tr>  <td>CSS</td>  <td>Lenguaje de diseño gráfico para establecer estilos en páginas web</td>  </tr>  </tbody>  <tfoot>  <tr>  <td colspan=”2”>Pie de la tabla</td>  </tr>  </tfoot>  En el siguiente enlace tenemos una fantástica guía sobre las tablas en HTML5: [aquí.](https://css-tricks.com/complete-guide-table-element/) |
| **<ol>** | Uso: Representa una lista ordenada.  Ejemplo:  <h2>Días de la semana</h2>  <ol>  <li>Lunes</li>  <li>Martes</li>  <li>Miércoles</li>  </ol> |
| **<ul>** | Uso: Representa una lista desordenada.  Ejemplo:  <h3>Juegos remarcables:</h3>  <ol>  <li>The Witcher 3</li>  <li>XCOM 2</li>  <li>Cyberpunk 2077</li>  </ol> |
| **<form>** | Uso: Un formulario engloba un conjunto de campos que son enviados para ser procesados por un servidor web.  <fieldset> Sirve para agrupar, mediante un recuadro, una serie de campos.  <legend> Sirve para dar un título al formulario.  Atributos importantes:   * Id -> Identifica de manera única al formulario. * Name -> Especifica el nombre del formulario. * Existen muchos más atributos que veremos la próxima semana.   Ejemplo:  <form name=”formulario1” id=”form1” method=”” action=””>  <fieldset>  <legend>Esto es un formulario</legend>  <input type=”text” name=”usuario” value=””>  <input type=”submit” name=”enviardatos” value=”Enviar”>  </fieldset>  </form> |
| **<label>** | Es un elemento de un formulario.  Uso: permite etiquetar el campo de un formulario. Mediante el atributo for se relaciona con el id de otro elemento, por ejemplo, un input. Es fundamental para proporcionar usabilidad y accesibilidad. Si se hace clic en la etiqueta, el cursor automáticamente se situará en el campo asociado.  Ejemplo:  <form name=”formulario2” id=”form1” method=”” action=””>  <fieldset>  <label id=”etiquetausuario” for=”usuario”>Usuario: </label>  <input type=”text” name=”nombreusuario” id=”usuario” value=””>  </fieldset>  </form> |
| **<datalist>** | Es un elemento de un formulario.  Uso: Genera una lista de datos que es reutilizable por otro tipo de inputs. Está formada por otros subelementos llamados <option>  Atributos importantes:   * Value – Indica el valor que va a tomar el input asociado.   Ejemplo:  <form name=”formulario2” id=”form2” method=”” action=””>  <input list=”listaSSOO” id=”SSOO”>  <datalist id=”listaSSOO”>  <option value=”Windows 10 October Update”>Windows 10</option>  <option value=”Linux Mint 19 Tara”>Linux Mint</option>  <option value=”Mac OS X Mojave”>Mac OS X</option>  </datalist>  </form> |
| **<input>** | Es un elemento de formulario.  Uso: Se utiliza para crear los campos de un formulario. Normalmente suelen permitir introducir una serie de datos.  Atributos importantes:   * Type -> Indica el tipo del campo. Existen multitud de tipos. Hablaremos de esto la próxima semana.   <form name=”formulario3” id=”form3” method=”” action=””>  <label for=”nombreusu”>Nombre de usuario:</label>  <input type=”text” id=”nombreusu” name=”acceso” placeholder=”Escribe tu nombre de usuario”>  </form> |
| **Botones** | Podemos implementarlos mediante:   * <input type=”button”> Es un botón genérico. Hay que asociarle Javascript para que haga algo. * <input type=”reset”> Vacía los campos de un formulario o establece sus valores por sus defecto. * <input type=”submit”> Envía los datos de un formulario. * <button id=”boton4”> Es una etiqueta que representa a un botón.   Ejemplos:  <input type=”button” id=”boton1” onclick=”alert(‘Hola’)” value=”Saludar”>  <input type=”reset” value=”Vaciar todos los campos” id=”boton2”>  <input type=”submit” value=”Enviar datos” id=”boton3”>  <button id=”boton4” onclick=”alert(‘Adiós’)”>Despedirse</button> |
| **<pre>** | Uso: Permite mostrar un trozo de texto con los mismos espacios y saltos de línea con los que aparece escrito en el código html.  Ejemplo:  <pre>  Este  Texto  Aparece tal cual lo he  Escrito  </pre> |
| **<img>** | Uso: Se utiliza para insertar una imagen en la página web.  Ejemplo:  <img src=”TBS.jpg” title=”The Banner Saga” alt=”The Banner Saga” id=”imagen1”> |
| **<video>** | Uso: Sirve para insertar un vídeo en una página web. El reproductor de vídeo dependerá del navegador.  Ejemplo:  <video id=”video1” width=”300” height=”200” autoplay=”autoplay” controls loop=”loop”>  <source src=”videos/Ashen.mp4” type=”video/mp4”>  <source src=”videos/Ashen.ogv” type=”video/ogv”>  <source src=”videos/Ashen.webm” type=”video/webm”>  Tu navegador no soporta esta etiqueta HTML5  </video> |
| **<audio>** | Uso: Se utiliza para insertar audio, de manera estándar, en una página web.  Ejemplo:  <audio id=”audio1” controls>  <source src=”Ghost.mp3” type=”audio/mpeg”>  <source src=”Ghost.ogg” type=”audio/ogg”>  <source src=”Ghost.wav” type=”audio/wav”>  Tu navegador no soporta esta etiqueta HTML5  </audio> |
| **<progress>** | Uso: muestra una barra de progreso.  Ejemplo:  <progress max="100" value="80"></progress> |
| **<canvas>** | Uso: es un lienzo donde se pueden pintar gráficos. Para poder pintar debemos apoyarnos en Javascript.  Ejemplo:  <canvas id="canvas1" width="300" height="150"  style="border:1px solid #000000;">  </canvas>  Ejemplo con pequeño script:  <canvas id="canvas2" width="200" height="100"  style="border:1px solid #c3c3c3;">  Tu navegador no soporta el elemento canvas.  </canvas>  <script>  var canvas = document.getElementById("canvas2");  var contexto = canvas.getContext("2d");  contexto.fillStyle = "#FF0000";  contexto.fillRect(0,0,150,75);  </script> |
| **Campos de formulario en HTML5** | **Recordatorio: Para realizar la validación automática de los campos del formulario en HTML5 hay que añadir una etiqueta input de tipo submit, ya que cuando presionamos la tecla de enviar es cuando se realiza el proceso de validación.**  **Input de tipo text**  <input type=”text”>  Ejemplo y uso: lo hemos visto anteriormente.  **Textarea**  Uso: permite conseguir una caja de texto multilínea.  Ejemplo:  <textarea id=”textarea1” name=”Descripcion” placeholder=”Introduce aquí la descripción de tu puesto de trabajo requerido” cols=”60” rows=”10”>  **Input de tipo search**  Uso: Se utiliza para especificar al navegador que éste será el campo oficial de búsquedas.  Ejemplo:  <input type=”search” id=”busqueda1” name=”busquedaweb” placeholder=”Introduce los términos de búsqueda”>  **Input de tipo password**  Uso: Se utiliza para insertar un campo que contenga datos de tipo contraseña.  Ejemplo:  <input type=”password” id=”pass1” name=”contrasena” placelholder=”Introduce una contraseña”>  **Input de tipo number**  Uso:Se utiliza para insertar un campo que contenga datos de tipo número.  Ejemplo:  <input type=”number” id=”numero1” name=”camponumero1” placeholder=”Introduce un número”>  **Input de tipo file**  Uso: Mediante este tipo de campo se permite al usuario adjuntar ficheros para su posterior envío.  Ejemplo:  <input type=”file” id=”fichero1” name=”fichero1”>  **Input de tipo url**  Uso: Mediante este tipo de campo podemos especificar una url. La autovalidación HTML5 permite obligar a cumplir el formato adecuado.  Ejemplo:  <input type=”url” id=”url1” name=”url1” placedholder=”Introduce una url válida”>  **Input de tipo email**  Uso: Representa un campo de tipo correo electrónico. La validación es no muy exigente y permite dominios no válidos. Con este campo vamos a probar el atributo size.  Ejemplo:  <laber for=”correo1” >Correo electrónico:</label>  <br>  <input type=”email” id=”correo1” name=”correo1” placeholder=”Introduce una dirección de correo electrónico” size=”50”>  **Input de tipo date**  Uso: Representa un campo de tipo fecha. Cuidado, etiqueta no soportada en Safari.  Ejemplo:  <laber for=”fecha1”>Fecha de nacimiento:</label>  <br>  <input type=”date” id=”fecha1” name=”fecha1”>  **Input de tipo time**  Uso: Sirve para representar un campo de tipo hora. Safari no soporta este tipo de etiqueta.  Ejemplo:  <laber for=”hora1”>Hora de salida:</label>  <br>  <input type=”time” id=”hora1” name=”hora1”>  **Input de tipo color**  Uso: Permite la selección de un color.  Ejemplo:  <laber for=”color1”>Selecciona un color:</label>  <br>  <input type=”color” id=”color1” title=”Color a introducir”>  **Input de tipo range o rango de valores**  Uso: Permite seleccionar un valor dentro de un rango  Ejemplo:  <laber for=”rango1”>Selecciona un valor:</label>  <br>  <input type=”range” id=”rango1” title=”Selecciona un valor del rango” min=”1” max=”20” value=”10” step=”2”> |
| **Atributos comunes de los campos de formularios en HTML** | **Imagen que contiene texto  Descripción generada con confianza muy alta**  Ejemplos importantes (pattern es fundamental):  <input type=”text” name=”dni” id=”inputdni” placeholder=”Introduce tu DNI” maxlength=”9” autofocus required=”true” pattern=”[0-9]{8}[A-Z]{1}” title=”Introduce un DNI: 8 números y una letra de la A a la Z”>  Gracias al atributo title, podemos ayudar al usuario y darle consejos.  Pattern para un DNI con letra mayúscula o minúscula:  pattern="[0-9]{8}[A-Za-z]{1}"  Al menos 1 caracter con minúsculas, mayúsculas o números:  [a-zA-Z0-9]+  0 o más caracteres con minúsculas, mayúsculas o números:  [a-zA-Z0-9]\* |
| **meta** | **Viewport**  Uso: Se trata del área de la página web visible por el usuario. Da al navegador instrucciones de cómo controlar las dimensiones de la página y el escalado. Es fundamental para que las webs se vean correctamente en todo tipo de dispositivos.  Más adelante veremos más consejos para realizar webs que se adapten a los dispositivos, pero ahora adelantaremos que nos ayudaremos del uso de media queries y que debemos tener cuidado al usar un ancho particular del viewport.  width=device-width  Uso: Establece el ancho de la página para que se adapte a las dimensiones del dispositivo.  initial-scale=1.0  Uso: Establece el zoom inicial cuando la página se carga por primera vez.  <meta name="viewport" content="width=device-width, initial-scale=1.0">  **http-equiv="X-UA-Compatible" content="IE=edge"**  Uso: Se utiliza esta etiqueta para solucionar posibles problemas de compatibilidad con Internet Explorer, que es un navegador en desuso pero que todavía tiene un porcentaje de mercado. En este caso, se usa el modo de compatibilidad más alto, que utiliza el modo Edge (nuevo navegador de Microsoft). |

**Breve iniciación al entorno de trabajo Bootstrap 4**

|  |  |
| --- | --- |
| **¿Qué es Bootstrap?** | Es un conjunto de herramientas de código abierto que permiten diseñar páginas web.  Contiene plantillas de diseño basadas en HTML y CSS. |
| **Instalación** | <http://getbootstrap.com/> |
| **Código esencial** | <!DOCTYPE html> <html lang="en">   <head>     <meta charset="utf-8">    </head> </html> |
| **Bootstrap está diseñado para ser responsivo (Adaptable a dispositivos móviles)** | Debemos añadir este código a <head>  <meta name="viewport" content="width=device-width, initial-scale=1"> |
| **Librerías básicas necesarias** | Podemos hacerlo de dos maneras para incluirlas:   * Realizar la instalación de Bootstrap descargando los ficheros necesarios: <http://getbootstrap.com/> * Incluyendo Bootstrap 4 CDN   **En <head> añadimos la hoja de estilos necesaria:**  <!doctype html>  <html lang="en">  <head>  <!-- Required meta tags -->  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  <!-- Bootstrap CSS -->  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">  <title>Hola!</title>  </head>  <body>  <h1>Hola!</h1>  <!-- Optional JavaScript -->  <!-- jQuery first, then Popper.js, then Bootstrap JS -->  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>  </body>  </html>  **Es posible que necesitemos también el JavaScript personalizado para aplicar comportamientos a los elementos que añadamos en Bootstrap.** |
| **Requiere unos contenedores para adaptar el contenido adecuadamente** | **Contenedor responsivo con tamaño fijo**  <div class="container">   <h1>My First Bootstrap Page</h1>   <p>This is some text.</p>  </div>  **Contenedor responsivo con ancho a tamaño completo**  <div class="container-fluid">   <h1>Mi primera web en Bootstrap 4</h1>   <p>Un poco de texto.</p>  </div> |
| **Sistema de rejilla en Bootstrap** | Para organizar el contenido, Bootstrap permite una división de hasta 12 columnas. Si no se quiere utilizar las 12 columnas, se pueden realizar subdivisiones.    Tenemos varias clases para hacer esto:   * col- (para dispositivos extra pequeños – pantallas de menos de 576px de ancho) **– Diseño responsivo** * col-sm- (para dispositivos pequeños – pantallas con un ancho 576px o más) **– Diseño responsivo** * col-md- (para dispositivos medianos – pantallas con 768px o más de ancho) – Diseño responsivo * col-lg- (para dispositivos grandes – pantallas con ancho mayor o igual a 992px) **– Diseño responsivo** * col-xl- (para pantallas de alta resolución 1200px o más de ancho)   Estos diseños apilarán unas columnas sobre otras si el ancho es menor que el especificado.  **Ejemplo:**  Lo primero que debemos hacer es crear una fila con row. Luego hacemos la división con las clases anteriores.  En este ejemplo, si lo probamos sobre un dispositivo que tenga menos de 768px de ancho, las columnas automáticamente se apilan.  <div class="row">   <div class="col-sm-4">.col-sm-4</div>   <div class="col-sm-4">.col-sm-4</div>   <div class="col-sm-4">.col-sm-4</div> </div> |
| **Ampliación del sistema de rejilla - Combinación de las diferentes clases para que se adapten a cualquier dispositivo** | Aquí tenemos un ejemplo que se adapta a dispositivos con pantallas de todo tipo, dependiendo del tamaño.  Observamos que las clases del mismo tipo suman 12 columnas en total (9 y 3, 7 y 12, 6 y 6, 10 y 2…) para realizar correctamente la distribución de la rejilla en Bootstrap.  <!-- 50%/50% en dispositivos extra pequeños(col) 75%/25% en dispositivos pequeños(sm) -->  <div class="row">  <div class="col-6 col-sm-9">col-6 col-sm-9</div>  <div class="col-6 col-sm-3">col-6 col-sm-3</div>  </div>  <!-- 25%/75% en dispositivos pequeños, a 50%/50% en dispositivos de tamaño medio, and a 33%/66% en dispositivos grandes. En dispositivos extra pequeños (100%) -->  <div class="row">  <div class="col-sm-3 col-md-6 col-lg-4">col-sm-3 col-md-6 col-lg-4</div>  <div class="col-sm-9 col-md-6 col-lg-8">col-sm-9 col-md-6 col-lg-8</div>  </div> |
| **Formateo de la rejilla** | **Colores de fondo predefinidos en Bootstrap**  bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark y .bg-light.  **Ejemplo:**  <div class="col-sm-3 col-md-6 bg-success">Hola</div>  <div class="col-sm-3 col-md-6 bg-warning">Hola</div>  <div class="col-sm-3 col-md-6 bg-success">Hola</div> |
| **Texto-Tipografía** | **<h1>-<h6> para los encabezados**  Podemos usar las clases para enriquecer los encabezados:   * .display-1, .display-2, .display-3, .display-4   **<small> para crear texto secundario en el encabezado**  Ejemplo: <h1> Encabezado 1 <small>Texto secundario</small></h1>  **<mark>** **para realizar resaltados**  **<abbr> para especificar abreviaturas**  **<blockquote> para especificar citas de texto**  **<code> para citar código en algún lenguaje de programación**  **<kbd> para citar combinaciones de teclado**  **<pre> para aplicar un texto preformateado con espacios y tabulados**  Tenemos muchas más clases tipográficas, pero sólo vamos a ver estas a modo de iniciación a Bootstrap. |
| **Colores** | **Texto**  .text-muted, .text-primary, .text-success, .text-info, .text-warning, .text-danger, .text-secondary, .text-white, .text-dark, .text-body .text-light  **Fondo**  .bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bg-secondary, .bg-dark, .bg-light |
| **Tablas** | **Una tabla básica:**  **Clase .table**  **<**table class="table">  <thead>  <tr>  <th>Nombre</th>  <th>Apellidos</th>  <th>Email</th>  </tr>  </thead>  <tbody>  <tr>  <td>Gonzalo</td>  <td>Ramírez</td>  <td>correo1@dominio.com</td>  </tr>  <tr>  <td>Marta</td>  <td>Torres</td>  <td>correo2@dominio.com</td>  </tr>  <tr>  <td>Tobías</td>  <td>González</td>  <td>correo3@dominio.com</td>  </tr>  </tbody>  </table>  **Tenemos otras clases para las tablas:**   * .table-striped * .table-bordered * .table-hover * .table-dark * .table-striped * .table-borderless   **Clases contextuales para aplicar a cada <tr>**   * .table-primary * .table-success * .table-danger * .table-info * .table-warning * .table-active * .table-secondary * .table-light * .table-dark   **Colores para las cabeceras:**   * .thead-dark * .thead-light   **Tablas responsivas (automáticamente incluyen un scroll horizontal únicamente para la tabla):**   * .table-responsive  |  |  | | --- | --- | | .table-responsive-sm | < 576px | | .table-responsive-md | < 768px | | .table-responsive-lg | < 992px | | .table-responsive-xl | < 1200px | | |
| **Imágenes** | **Con bordes redondeados:** .rounded  **Forma circular:** .rounded-circle  **Formato miniatura:** .img-thumbnail  **Alienación de imágenes:**   * **Flotante a la izquierda:** .float-left * **Flotante a la derecha:** .float-right * **Imagen centrada (con margin auto y display:block):** .mx-auto y .d-block **Ejemplo:** <img src="paris.jpg" class="mx-auto d-block">   **Imágenes responsivas:**   * **Ancho máximo 100% y height:auto:** .img-fluid |
| **Jumbotron – Caja con texto llamativo** | **Clase** .jumbotron  **Con ancho a tamaño completo:** .jumbotron-fluid |
| **Botones** | <button type="button" class="btn">Botón básico</button>  **Tenemos muchas clases adicionales para darle formato a este botón básico:**   * btn-primary * btn-secondary * btn-success * **…**   **Tamaños de botón:**   * btn-lg * btn-primary * btn-sm * **Ancho completo:** btn-block   **Activar y desactivar botones:**   * .active * .disabled   **Grupos de botones:**   * .btn-group * .btn-group-vertical * **Tenemos más clases: menús con botones desplegables…** |
| **Barras de progreso** | **Ejemplo básico:**  <div class="progress">   <div class="progress-bar" style="width:70%"></div> </div> |
| **Menús y botones dropdown** | **Necesitamos la siguiente librería (actualizar con enlaces a local):**  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script><script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous"></script><script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></script>  **Ejemplo básico con** .dropdown-toggle y data-toggle="dropdown"  <div class="dropdown">   <button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">     CD Projekt Red   </button>   <div class="dropdown-menu">     <a class="dropdown-item" href="#">The Witcher</a>     <a class="dropdown-item" href="#">The Witcher2</a>     <a class="dropdown-item" href="#">The Witcher 3</a>  <a class="dropdown-item" href="#">Cyberpunk 2077</a>   </div> </div> |
| **Menús horizontales** | **Ejemplo básico:**  <ul class="nav">   <li class="nav-item">     <a class="nav-link" href="#">Fable</a>   </li>   <li class="nav-item">     <a class="nav-link" href="#">Halo</a>   </li>   <li class="nav-item">     <a class="nav-link" href="#">Quantum Break</a>   </li>   <li class="nav-item">     <a class="nav-link disabled" href="#">Gears Of War</a>   </li> </ul> |
| **Formularios** | **Ejemplo básico:**  <form action="">   <div class="form-group">     <label for="email">Correo electrónico:</label>     <input type="email" class="form-control" id="email">   </div>   <div class="form-group">     <label for="pwd">Contraseña:</label>     <input type="password" class="form-control" id="pwd">   </div>   <div class="form-group form-check">     <label class="form-check-label">       <input class="form-check-input" type="checkbox"> Recordar mis datos     </label>   </div>   <button type="submit" class="btn btn-primary">Submit</button> </form> |
| **Carrusel de imágenes** | **Ejemplo básico:**  <div id="demo" class="carousel slide" data-ride="carousel">    <!-- Indicators -->   <ul class="carousel-indicators">     <li data-target="#demo" data-slide-to="0" class="active"></li>     <li data-target="#demo" data-slide-to="1"></li>     <li data-target="#demo" data-slide-to="2"></li>   </ul>    <!-- The slideshow -->   <div class="carousel-inner">     <div class="carousel-item active">       <img src="la.jpg" alt="Los Angeles">     </div>     <div class="carousel-item">       <img src="chicago.jpg" alt="Chicago">     </div>     <div class="carousel-item">       <img src="ny.jpg" alt="New York">     </div>   </div>    <!-- Left and right controls -->   <a class="carousel-control-prev" href="#demo" data-slide="prev">     <span class="carousel-control-prev-icon"></span>   </a>   <a class="carousel-control-next" href="#demo" data-slide="next">     <span class="carousel-control-next-icon"></span>   </a>  </div> |
| **Flex – Elemento muy importante para la distribución (layouts) en las páginas web** | **Esta tecnología no es soportada en IE9 y anteriores. En este caso debemos usar Bootstrap 3.**  **Para crear un layout mediante flex con elementos felixbles utilizamos: d-flex**  **Mediante flex-row obligamos a mostrar los elementos de izquierda a derecha:**  <div class="d-flex flex-row bg-secondary">   <div class="p-2 bg-info">Flex item 1</div>   <div class="p-2 bg-warning">Flex item 2</div>   <div class="p-2 bg-primary">Flex item 3</div> </div>  **Para mostrar los elementos de manera vertical:**  <div class="d-flex flex-column">   <div class="p-2 bg-info">Flex item 1</div>   <div class="p-2 bg-warning">Flex item 2</div>   <div class="p-2 bg-primary">Flex item 3</div> </div>  <div class="d-flex flex-column-reverse">   <div class="p-2 bg-info">Flex item 1</div>   <div class="p-2 bg-warning">Flex item 2</div>   <div class="p-2 bg-primary">Flex item 3</div> </div |
| **Más elementos en Bootstrap**  **(No entra en el examen)** | **Bootstrap tiene una cantidad enorme de elementos y clases que pueden ser consultados en:** [**https://www.w3schools.com/bootstrap4/default.asp**](https://www.w3schools.com/bootstrap4/default.asp) |