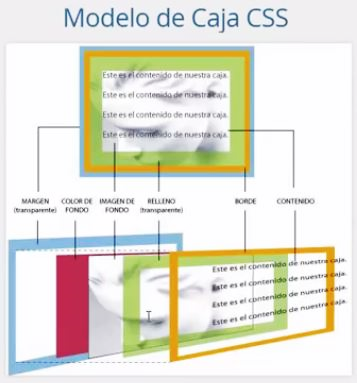
**GUIA DEL MODELO DE CAJA - FUNDAMENTOS**

Antes de comenzar, hay que diferenciar y comprender que una cosa es el "**Modelo de caja**", otra cosa es el "**Posicionamiento de caja**" y otra cosa el “**Modelo de Layout**”.

Esta guía trata el tema del **“Modelo de caja”** o también conocido como “**Box Model**”.

**ENSEÑANZA SOBRE EL “MODELO DE CAJA” … SEGÚN JONMIRCHA (BEXTLAN.COM)**

El “**Modelo De Caja**” hace que todos los elementos de un documento HTML se representen mediante **cajas rectangulares**. CSS permite controlar el aspecto de todas las “**cajas**” o mejor conocidos como “**contenedores**”.

****CSS permite definir la altura (**height**) y anchura (**width**) de cada “**caja**”, el margen (**margin**) existente entre cajas y el espacio de relleno interior (**padding**) que muestra cada caja. A demás, CSS permite controlar la forma en la que se visualizan, pero la mayoría de cajas no muestran ningún color de fondo ni ningún borde, no son visibles a simple vista.

Existen **6 propiedades CSS** que podemos utilizar para manipular/manejar este modelo de caja:

* **Margin**
* **Border**
* **Padding**
* **Content**
* **Background-color (opcional)**
* **Background-image (opcional)**
* **Margen o margin**: Es la distancia externa que hay entre los elementos y los márgenes del documento u otros elementos, pueden manejarse los márgenes de las cuatro coordenadas: **top**, **bottom**, **left**, **right**.

Sintaxis: “**margin-top: 1em;**”

Sintaxis: “**margin-bottom: 1em;**”

Sintaxis: “**margin-left: 1em;**”

Sintaxis: “**margin-right: 1em;**”

* Shorthand con un valor: “**margin: 1em;** Asignando un solo valor se indica que los 4 lados lo tendrán”
* Shorthand con dos valores: “**margin: 1em 2em;** Asignando dos valores se indica que el primero [**1em**] se aplicará a los verticales: **top** y **bottom**, y el segundo valor [**2em**] se aplicará a los horizontales: **left** y **right**”
* Shorthand con tres valores: “**margin: 1em 2 em 3em;** Asignando tres valores se indica que el primero [**1em**] se aplicará a **top**, el segundo valor [**2em**] se aplicará a los horizontales: **left** y **right**, y el tercero [**3em**] se aplicará a **bottom**”
* Shorthand con cuatro valores: “**margin: 1em 2em 3em 4em;** Asignando 4 valores se indica que el primero [**1em**] se aplicará a **top**, el segundo valor [**2em**] se aplicará a **right**, el tercer valor [**3em**] se aplicará a **bottom**, y el cuarto valor [**4em**] se aplicará a **left**”
* **Borde o border**: Es la línea que delimita los elementos y divide el “**margin**” del “**padding**”.

Sintaxis: “**border-width: thin | thick | medium | 10px | 1em;**”

Sintaxis: “**border-style: solid | dashed | dotted;**”

Sintaxis: “**border-color: #HEXADECIMAL | orange;**”

Shorthand: “**border: thin solid #HEXADECIMAL;**”

Sintaxis: “**border-right: thin solid green;**”

Sintaxis: “**border-left: thin solid green;**”

Sintaxis: “**border-top: thin solid green;**”

Sintaxis: “**border-bottom: thin solid green;**”

Shorthand: “**border: thin solid green;**”

* **Relleno o padding**: Es la distancia interna que hay entre los márgenes del elemento, pueden manejarse los rellenos de las cuatro coordenadas: **top**, **bottom**, **left**, **right**.

Sintaxis: “**padding-top: 1em;**”

Sintaxis: “**padding -bottom: 1em;**”

Sintaxis: “**padding -left: 1em;**”

Sintaxis: “**padding -right: 1em;**”

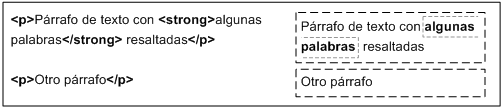
* Shorthand con un valor: “**padding: 1em;** Asignando un solo valor se indica que los 4 lados lo tendrán”
* Shorthand con dos valores: “**padding: 1em 2em;** Asignando dos valores se indica que el primero [**1em**] se aplicará a los verticales: **top** y **bottom**, y el segundo valor [**2em**] se aplicará a los horizontales: **left** y **right**”
* Shorthand con tres valores: “**padding: 1em 2 em 3em;** Asignando tres valores se indica que el primero [**1em**] se aplicará a **top**, el segundo valor [**2em**] se aplicará a los horizontales: **left** y **right**, y el tercero [**3em**] se aplicará a **bottom**”
* Shorthand con cuatro valores: “**padding: 1em 2em 3em 4em;** Asignando 4 valores se indica que el primero [**1em**] se aplicará a **top**, el segundo valor [**2em**] se aplicará a **right**, el tercer valor [**3em**] se aplicará a **bottom**, y el cuarto valor [**4em**] se aplicará a **left**”
* **Contenido o content**:
* **Imagen de fondo o background-image (opcionales)**:
* **Color de fondo o background-color (opcionales)**:

**NOTA: Hay que tomar en cuenta que los atributos del “Modelo De Caja” (border, margin y padding) por defecto aumentarán el ancho (width) y el alto (height) de la misma.**

**ENSEÑANZA SOBRE EL “MODELO DE CAJA” … SEGÚN INTRODUCCION A CSS (LIBROSWEB.ES)**

El “**Modelo de caja**” o "**Box Model**" es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web. El “**Modelo de caja**” **es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante cajas rectangulares**.

Las cajas de una página se crean automáticamente. **Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento**. La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:



**El contenido está dentro de una caja y las palabras en negrita están dentro de otra caja.**

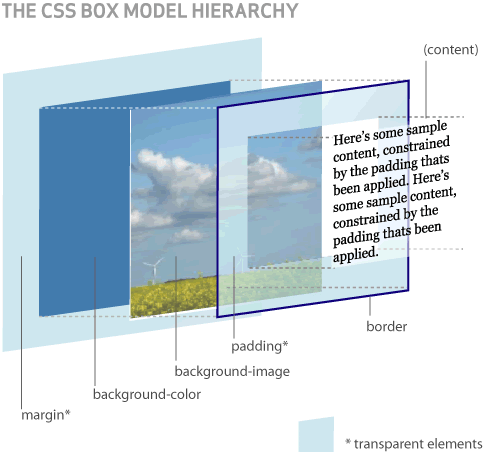
Como se dijo, las cajas se crean automáticamente al definir cada elemento HTML.

**Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde**. La siguiente imagen muestra las cajas que forman la página web de [www.alistapart.com](http://www.alistapart.com) después de forzar que todas las cajas muestren su borde:

Cada una de las cajas está formada por seis partes, tal y como muestra la siguiente imagen:

**Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes**:

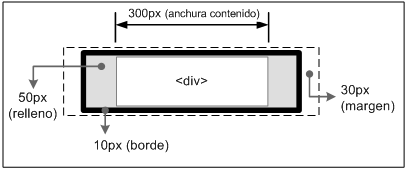
* **Contenido (content)**: se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
* **Relleno (padding)**: Es **el espacio libre** opcional existente entre el **contenido** y el **borde**.
* **Borde (border)**: Es **la línea que encierra** completamente el **contenido** y su **relleno**.
* **Imagen de fondo (background image)**: Es la imagen que **se muestra por detrás del contenido y el espacio de relleno**.
* **Color de fondo (background color)**: Es el color que **se muestra por detrás del contenido y el espacio de relleno**.
* **Margen (margin)**: Es **la separación** opcional **existente entre la** **caja** **y el resto de cajas adyacentes**.



El **relleno** y el **margen** son transparentes, por lo que en el espacio ocupado por el relleno se muestra el color o imagen de fondo (si están definidos) y en el espacio ocupado por el margen se muestra el color o imagen de fondo de su elemento padre (si están definidos). Si ningún elemento padre tiene definido un color o imagen de fondo, se muestra el color o imagen de fondo de la propia página (si están definidos).

**NOTA: Si una caja define tanto un color como una imagen de fondo, la imagen tiene más prioridad y es la que se visualiza. No obstante, si la imagen de fondo no cubre totalmente la caja del elemento o si la imagen tiene zonas transparentes, también se visualiza el color de fondo. Combinando imágenes transparentes y colores de fondo se pueden lograr efectos gráficos muy interesantes.**

La anchura y altura de un **elemento** o **contenedor** no solamente se calculan teniendo en cuenta sus propiedades “**Width**” y “**height**”. El **margen**, el **relleno** y los **bordes** establecidos a un elemento determinan la **anchura y altura final del elemento**. En el siguiente ejemplo se muestran los estilos CSS de un **elemento**:



*div {*

*width: 300px;*

*padding-left: 50px;*

*padding-right: 50px;*

*margin-left: 30px;*

*margin-right: 30px;*

*border: 10px solid black;*

*}*

De esta forma, la **anchura final** del **elemento** en pantalla sería igual a **la suma de la anchura original**, los **márgenes**, los **bordes** y los **rellenos**:

**30px + 10px + 50px + 300px + 50px + 10px + 30px = 480px’s**

Así, la **anchura/altura** **establecida con CSS** siempre **hace referencia a la anchura/altura del contenido** **(como lo muestra la imagen)**. La **anchura/altura total del elemento debe tener en cuenta además los valores del resto de partes que componen la caja**.

**ENSEÑANZA SOBRE EL “MODELO DE CAJA” … SEGÚN ÁLVARO FELIPE (EDTEAM.COM)**

Antes de comenzar se debe conocer que hay elementos HTML de tipo **bloque** (“**block**”), **en línea** (“**Inline**”) y **bloque en línea** (“**Inline-block**”), por ejemplo:

* **Bloque o block**: Los elementos HTML que trabajan en **bloque**, prácticamente todas las cajas contenedoras que pertenecen a la **parte semántica de HTML**, como: “**ul**”, “**ol**”, “**form**”, "**div**", "**p**", "**h1...h6**", "**header**", "**nav**", "**footer**", "**section**", "**aside**" o "**article**". Por default, el ancho de un elemento **bloque** está al 100% de todo el ancho de la ventana del navegador o de su contenedor padre. **Este tipo de elementos siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea**.

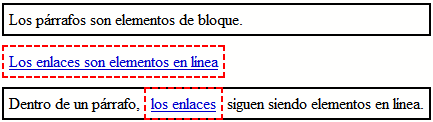
Sintaxis: “**Display: block;**”

* **Linea o inline**: Los elementos HTML que trabajan **en línea**, prácticamente todas aquellas etiquetas que le dan **formato a los textos**, como: “**textarea**”, “**label**”, "**span**", “**img**”, “**strong**”, “**audio**”, “**video**”, "**b**", "**strong**", "**li**" o "**a**". Por default, el ancho de un elemento **en línea** se ajusta de acuerdo con su contenido. **Este tipo de elementos no empiezan necesariamente en una nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.**

Sintaxis: “**Display: inline;**”

* **Bloque en línea o inline-block:** Los elementos HTML que trabajan en **bloque en línea** toman/muestran un comportamiento muy similar a como si utilizáramos "**display:flex**" y "**flex-direction:row|row-reverse**", pues pone los elementos en fila (horizontal) uno alado del otro, a comparación de la propiedad "**float**", esta propiedad si agrega una pequeña separación entre los elementos que están juntos.

Sintaxis: “**display: inline-block;**”



Ahora, el “**Modelo De Caja**” es el conjunto de algoritmos que posee el navegador web con los cuales podrá reflejar/dibujar una caja en pantalla/ventana con todas sus propiedades.

* **Caja**: Una caja es un rectángulo definido por todo elemento HTML al renderizarse (**esto es cuando el navegador dibuja la página en la pantalla**) la página. A una **caja** se le puede ver/llamar también como un **bloque** o "**block**". Todo elemento HTML crea un **rectángulo**. Por ejemplo, un contenedor "**div**" es una caja de tipo **bloque**. Las **cajas** son **rectangulares**, no hay triangulares ni curvas, si algún día has visto un elemento curvo o triangular en alguna página no significa que la caja sea triangular, seguramente han utilizado **SVG** o **CANVAS**.

Todo elemento HTML define una **caja rectangular**, todos, por ejemplo: un "**a**", un "**span**", un "**p**", un "**section**", un "**header**" o un "**div**".

**PARTES QUE POSEE UNA CAJA/ELEMENTO HTML**

Hay 2 tipos de elementos que posee una caja:

* **Elementos internos**: Los que están dentro de la caja, y son los siguientes 3 tipos:
  + **content**: Es el contenido en si, por ejemplo, puede ser texto, como un párrafo o un título.
  + **padding**: Es el espacio entre el contenido (**content**) y el límite de la caja (**border**).
  + **border**: Es la línea que se dibuja alrededor de la caja y es la muestra del límite de la misma.

Parecerá algo confuso, pero no, y es que estos 3 elementos, de alguna forma, definen otras 3 cajas o pseudocajas: El **content box**, El **padding box** y El **border box**.

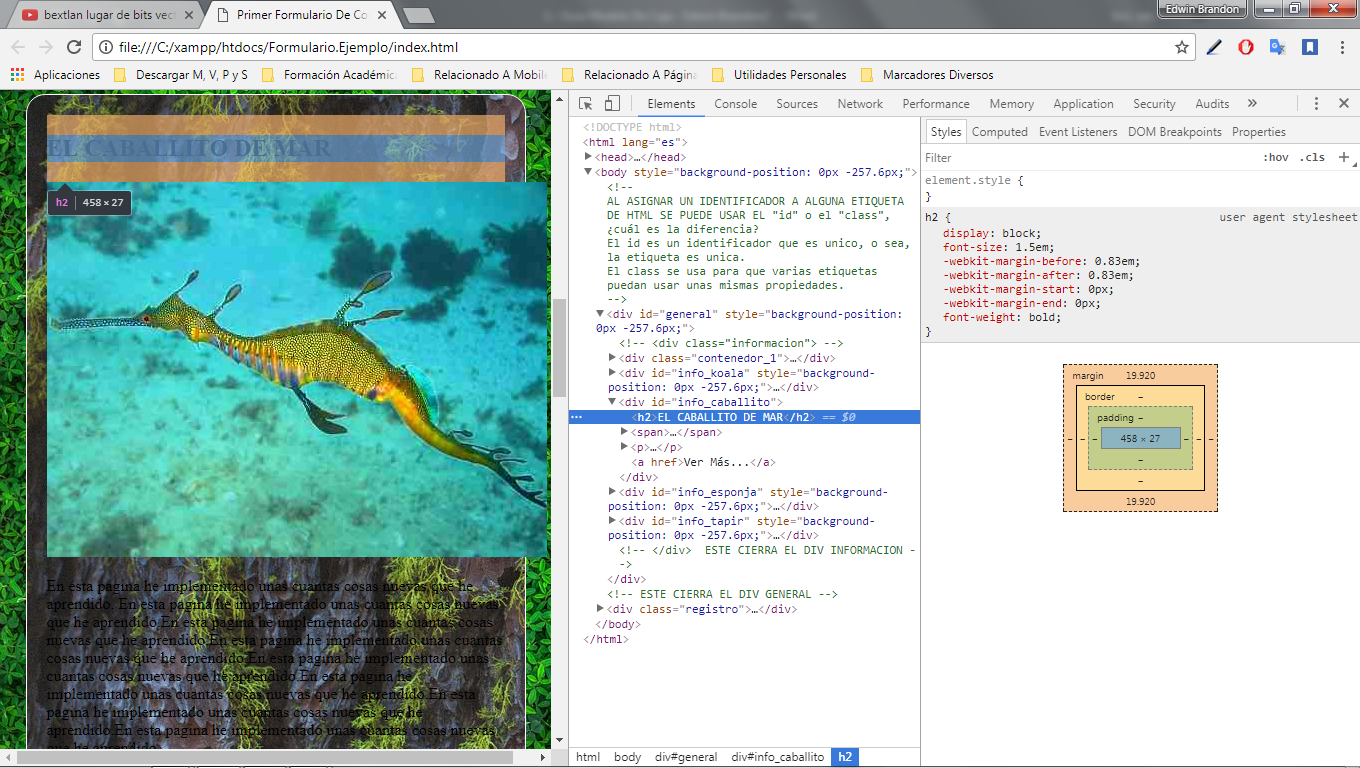
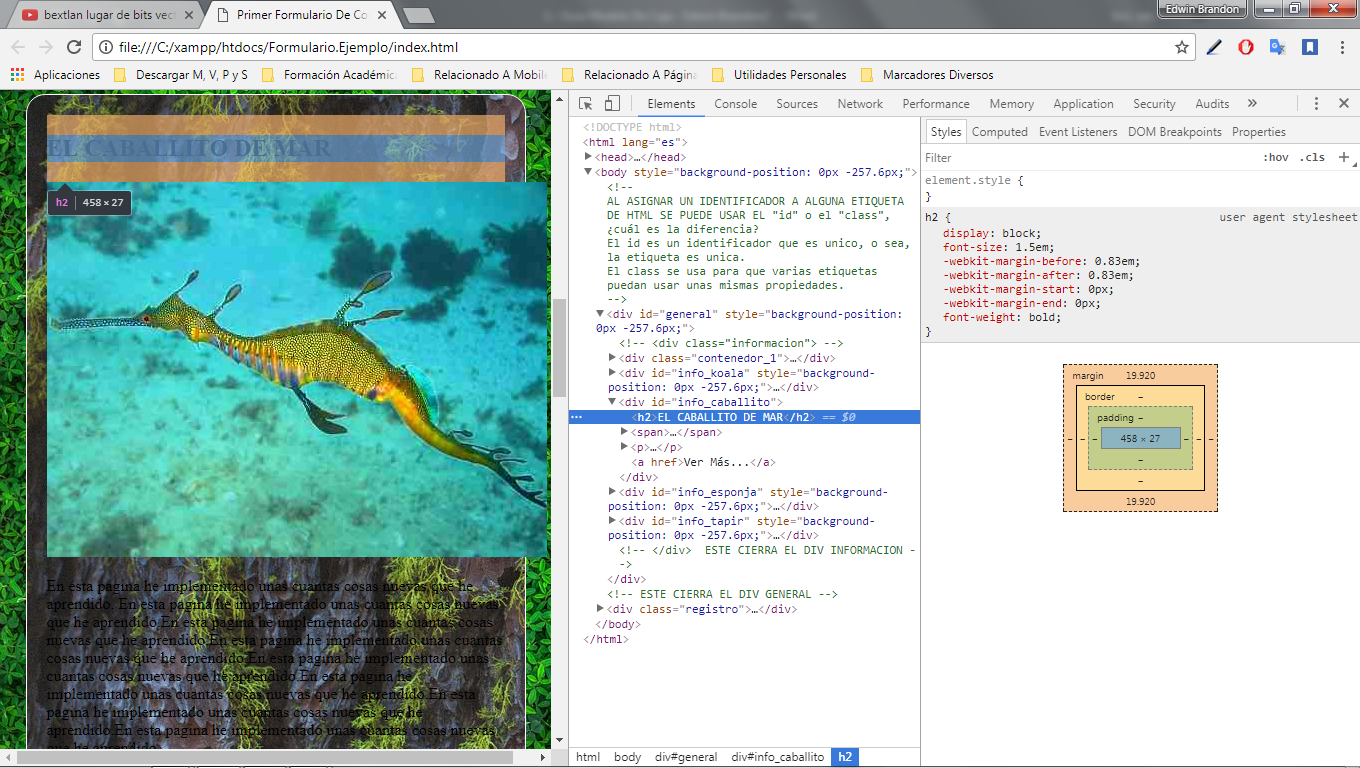
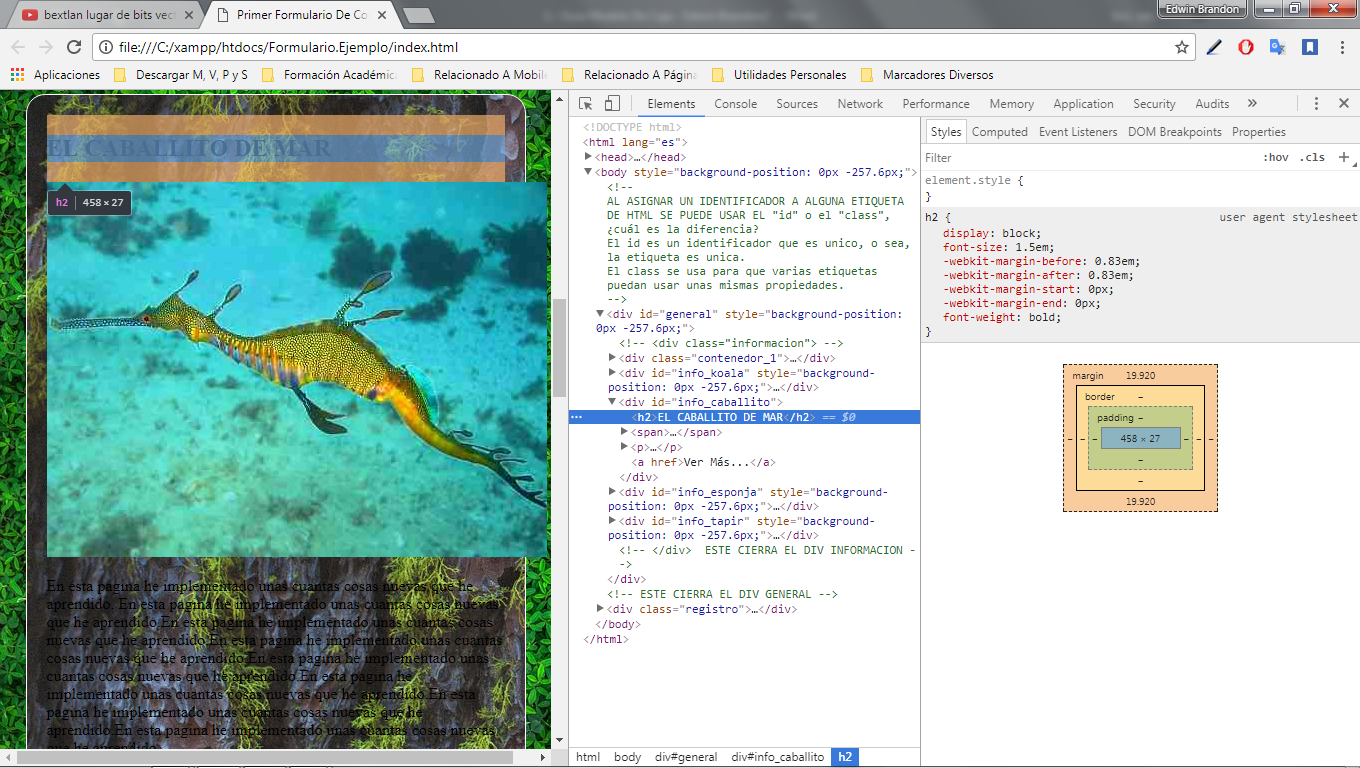
* **Elementos externos**: Los que están fuera de la caja pero que son parte del modelo de esa caja, y es 1 tipo de elemento:
  + **margin**: Es la separación entre una caja y las adyacentes, o sea, todas las otras cajas que están juntas según el flujo de escritura en el HTML. Este **margin** es externo.

Toda caja posee estos elementos, aunque uno mismo **No los defina**.

Para poder ver de una forma más gráfica las partes de estos elementos HTML en su forma de cajas, los navegadores poseen un "**inspector**" o "**herramienta de desarrollador**" que se utiliza al presionar **F12**.

**En el inspector**:

* Pueden visualizarse los elementos HTML y se puede ubicar en cada uno de ellos al darle clic al que deseemos.
* El inspector colorea las partes de los elementos HTML cuando pasamos el cursor sobre ellos.
* Hay una sección en esta misma ventana del inspector que muestra la caja, tal cual, con sus elementos para entender mejor esto, los colores ayudan a entender mejor la caja y sus elementos:
  + **Azul**: Pertenece al “**content**”, y es el contenido que hay dentro de la caja.
  + **Verde**: Pertenece al “**padding**”, y es el relleno que se le puede dar a la caja, que puede ser por dentro (**border-box**) o por fuera (**border-content**).
  + **Naranja**: Pertenece al “**border**”, y es el borde que posee la caja.
  + **Melón**: Pertenece al “**margin**”, y es el margen o separación que tiene/hay entre el elemento y su contenedor padre.

**Ventana Del Inspector**

**RENDERIZADO DE PAGINA**

**(Elemento “h2” seleccionado, no tiene padding ni border definidos)**

**ELEMENTOS DE LA CAJA O CONTENEDOR**

**(Modelo de caja de un elemento “h2”)**

**CODIGO HTML**

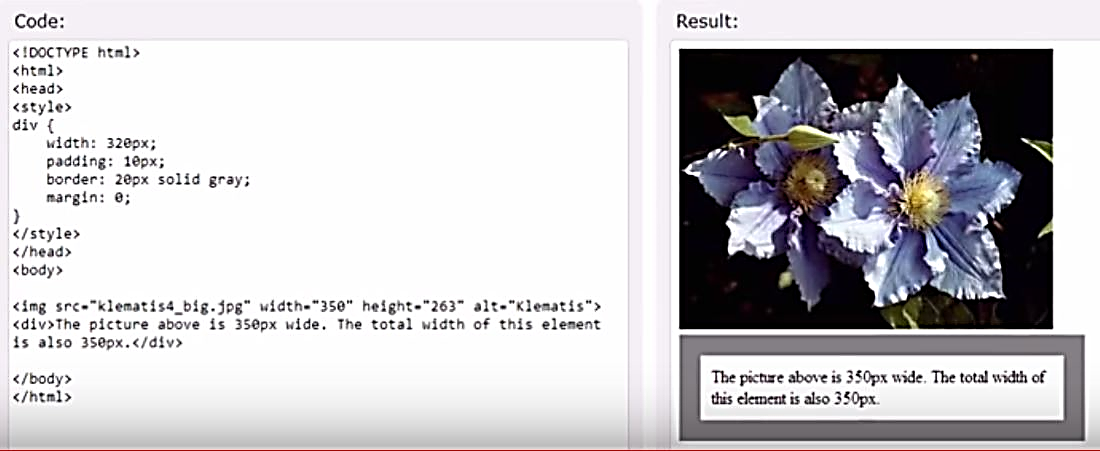
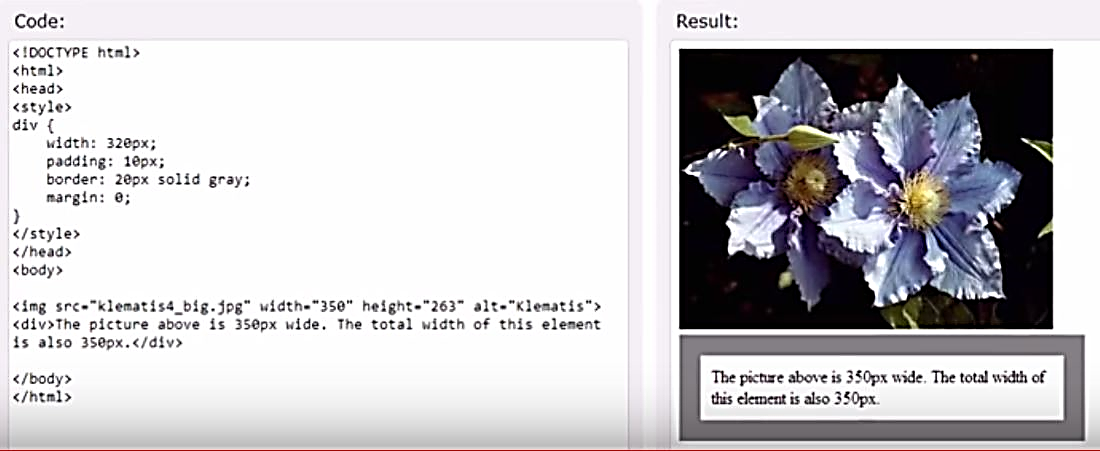
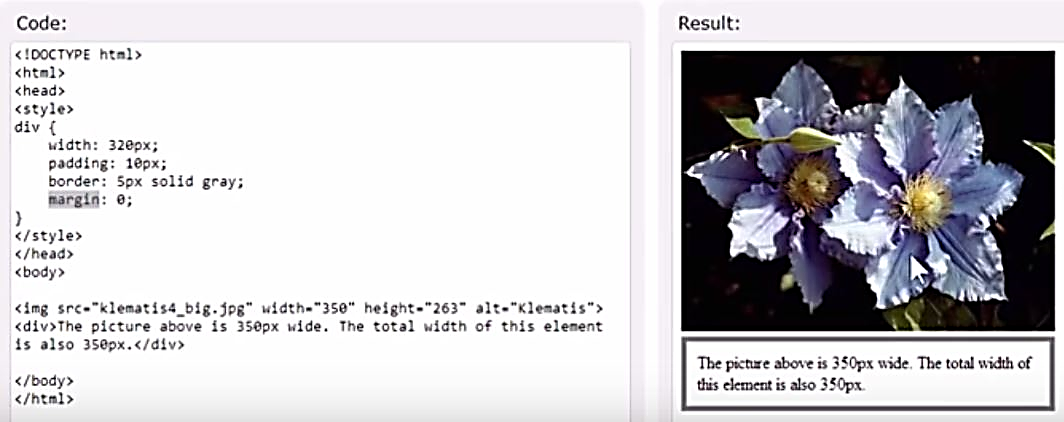
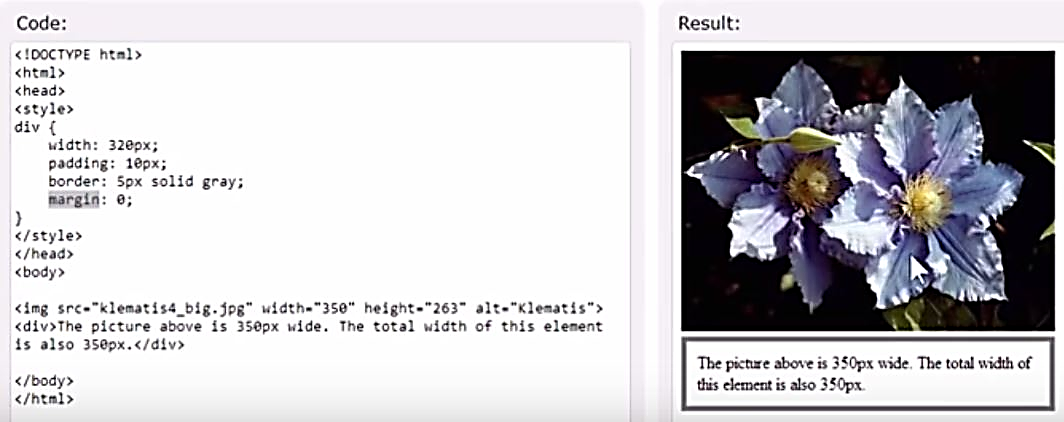
**(Seleccionando un “h2”)**

**Content box**

**Padding box**

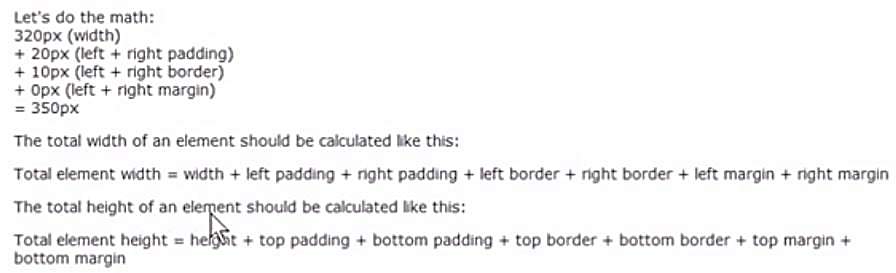
**Margin box**

**Border box**

**Calculando El Modelo De Caja (Ejemplo de la “W3C”)**

**Mal uso del Modelo De Caja**

**Buen uso del Modelo De Caja**

**Operación Para El Buen Uso Del Modelo De Caja (Del ejemplo anterior)**

**Fórmula Para Calcular el ancho**

**Fórmula Para Calcular el alto**

**REFERENCIAS**

* Principalmente vídeos guías de EDteam (Álvaro Felipe) y Bextlan (JonMircha)
* Otros vídeos relacionados con el tema de FalconMaster y la W3C
* El libro de “Introducción a CSS”, en su capítulo 4, de la página librosweb.es