| **TRABAJANDO CON MARGEN, RELLENO, BORDE Y ELEMENTO** | | |
| --- | --- | --- |
| Las cajas se crean automáticamente al definir cada elemento **HTML** | | |
|  | | Las cajas se crean automáticamente al definir cada elemento HTML |
| Ejemplo  http://es.html.net/tutorials/css/figure010.gif | | http://es.html.net/tutorials/css/figure009.gif |
| Ejemplos de márgenes | | **MARGEN**  body {  margin-top: 100px;  margin-right: 40px;  margin-bottom: 10px;  margin-left: 70px;  }  **O**  O podrías elegir usar la versión combinada de margin, que queda como más elegante:  body {  margin: 100px 40px 10px 70px;  } |
| **PADDING**  h1 {  background: yellow;  padding: 20px 20px 20px 80px;  }  h2 {  background: orange;  padding-left:120px;  } | | **BORDES**   * border-width * border-color * border-style  | **border-width** | **border-style**  Diferentes estilos de borde | | --- | --- | | Ejemplos de border-width | | h1 {  border-width: thick;  border-style: dotted;  border-color: gold;  } | |
|  |  |  |

| **TRABAJANDO CON LA CAJA DE HTML**  **Realizar las siguientes pruebas**  Al trabajar con un modelo de caja, debemos tener en cuenta que existen dos propiedades principales para definirla, height y width, que definirán el ancho y alto de nuestra caja. | |
| --- | --- |
| **Ejemplo 1 HTML** | **CSS** |
| <div></div> | div{  background: #5DEAA8;  height: 100px;  width: 200px;  } |
| **Ejemplo 2 HTML**  Adicionalmente a estas, tenemos el Margin, Padding y Border que nos brindaran mayor control al momento de organizar los elementos. | |
| <div></div> | **CSS**  div{  border: 10px solid #3EC483;  background: #5DEAA8;  height: 100px;  width: 200px;  padding: 10px;  margin: 10px;  } |
| **Ejemplo 3 HTML**  Si comparamos ambas imágenes, podremos observar que existe una pequeña variación de tamaño entre una y otra, esto es debido a que el padding y el border están agregando 10 píxeles respectivamente a cada lado de nuestra caja, haciendo que el tamaño de esta se incremente en 40 del tamaño original, siendo ahora de 240px width y 140 px en height. | |
| <div class="div1"></div>  <div class="div2"></div> | **CSS**  .div1{  background: #5DEAA8;  height: 100px;  width: 200px;  }  .div2{  border: 10px solid #3EC483;  background: #5DEAA8;  height: 100px;  width: 200px;  padding: 10px;  margin: 10px;  } |
| **Ejemplo 4 HTML**  También podemos observar que ahora nuestra caja se está separando del borde de la pantalla, esto pasa gracias al marginquien está agregando 10 pixeles por lado, para poder entender mejor veamos la siguiente imagen.  https://lh3.googleusercontent.com/HyM7vtKJfHDSOMADQ7U2IUC8KSod3lNuk3DrwZLPe2ZVbzHxQCT1MgZydlXZYaXk9NXVHF3A4WbG6m9a9YH2s4E6xGqMjkl02Uck__dcslz4LgFn23ny1P1iYgL_EAi3HlUXYKxH | |
| Entonces para saber el ancho real de nuestra caja debemos sumar lo siguiente.  margin-left + border-left + padding-left + width + padding-right + border-right + margin-right  De igual manera con el alto de la caja.  margin-top + border-top + padding-top + height + padding-bottom + border-bottom + margin-bottom  Lo que finalmente nos quedaría es:  Width: 260px = 10px + 10px + 10px + 200px + 10px + 10px + 10px  Height: 160px = 10px + 10px + 10px + 100px + 10px + 10px + 10px  El modelo de caja puede llegar a ser un poco confuso, ya que inicialmente definimos un alto y ancho específico para nuestra caja, pero gracias al padding, border y margin este tamaño varía. Esto ocurre porque el modelo de caja por defecto es aditivo; por lo tanto, para determinar el tamaño real, deberemos calcularlo según se muestra en el ejemplo anterior. | |
| **Ejemplo 5 HTML** |  |
| Las propiedades margin y padding pueden aceptar de uno a cuatro valores, como se muestra a continuación. | |
| <div></div>  https://lh3.googleusercontent.com/ZtIFZWwGh3nIvklkrLB16qX3Degu8nzuH8K2oL1mzijPlOtbev59b0W1_vqwWviaaGu4ghPQrquR_CoeG4cnWGpNmLQ1l363lXLNkRPH4O-6jXXWuXb5gbttWL3a1D07TFFi__X- | **CSS**  div{  background: #5DEAA8;  height: 100px;  width: 100px;  padding: 20px;  margin: 40px;  } |
| Si declaramos un solo valor para Margin y Padding, este se establecerá ese valor para cada lado de la caja. | |
| <div></div>  https://lh3.googleusercontent.com/nBRpLVGWsGIV2IDn3AtzzovTjPXqxEfaoKalzr20PCAPmJGsNOLsRssG2ajINsGzw4owM-hdUl8_rlKJrAvWjERrtKBz0c35Hgo9zZc89sv5L2J_4GXu1VnxbWf-ssiLSR2QqnLk | **CSS**  div{  background: #5DEAA8;  height: 100px;  width: 200px;  padding: 20px 50px;  margin: 40px 30px;  } |
| **Ejemplo 6 HTML** | |
| Si declaramos 2 valores para Margin y Padding, se establecerá el primero para Top y Bottom y el segundo para Left y Right | |
| <div></div>  https://lh6.googleusercontent.com/qoCffzYKvO9BAwCZtOv7GvTxWILSjqCmKpsp4NTDcl3y68kJ00movXIUqEYuerhvBHwOWLDsJIWyT-PlOeGq8UTwMlTEvtzz3iTbkq_38ndTeIsIyjdxsoB1umCRtQuWzgmIhOfP | **CSS**  div{  background: #5DEAA8;  height: 100px;  width: 200px;  padding: 20px 50px 60px;  margin: 40px 30px 70px;  } |
| Con 3 valores estableceremos el Top, Right y Bottom de nuestro Padding y Margin, al no establecer el valor de Left este tomara el mismo que el de Right. | |
| <div></div>  https://lh5.googleusercontent.com/o2y52rFMYaUic_Em-yCjJwh0UiaMt5SGZQEU8Z99pVQn8v1tBsxUt5d2vWQg8TzVSA5cPssE9aHj_dnGO-pKLTTIK0bPOJ0F-xaNb2wWVXouqvL5oBZZh6cyNDN9Ex6ga6dvnN_q | **CSS**  div{  background: #5DEAA8;  height: 100px;  width: 200px;  padding: 20px 50px 60px 90px;  margin: 40px 30px 70px 80px;  } |
| Finalmente estableciendo 4 valores le asignamos un tamaño a cada lado del margin y padding.  Es importante mencionar que la asignación de valores va en sentido horario comenzando por el Topy finalizando en el Left  Entre el Margin y el Padding encontramos al Border el cual requiere de 3 valores (width, style y color) para poder ser visualizado correctamente. | |
| <div></div> | **CSS**  div{  border: 20px solid #3EC483;  background: #5DEAA8;  height: 100px;  width: 200px;  padding: 10px;  margin: 10px  } |
| Cuando deseamos hacer que un borde sea redondeado utilizamos border-radius. Esta propiedad modifica la forma en la que se muestran las esquinas de nuestra caja, border-radius acepta valores en pixeles y en porcentajes, además podemos establecer valores para cada borde por separado, muy parecido a lo que vimos anteriormente. | |
| <div class="div1"></div>  <div class="div2"></div> | **CSS**  .div1{  background: #5DEAA8;  height: 100px;  width: 100px;  border-radius: 25px;  }  .div2{  background: #5DEAA8;  height: 100px;  width: 100px;  margin-top: 10px;  border-top-left-radius: 50%;  border-top-right-radius: 50%;  border-bottom-right-radius: 50%;  border-bottom-left-radius: 10%;  } |
| Finalmente veremos el **Box-Sizing** propiedad que es usada para modificar las propiedades del Box Model visto anteriormente, brindándonos aún más control sobre nuestra caja.  box-sizing acepta principalmente 3 valores: **content-box, padding-box y border-box**, cada uno de estos modifica el cómo se calcula el tamaño de nuestra caja | |
| **Box-sizing: content-box**  **Box-sizing: padding-box**  **Box-sizing: border-box** | |
| **Content-Box**  Es el valor por defecto que tiene el Box Model, su características es que es aditivo, es decir, si agregamos un tamaño al margin, padding y border estos se sumaran al width y height establecidos, agrandando el tamaño real de nuestra caja. | |
| <div></div>  https://lh4.googleusercontent.com/ukUn177m7hXsYZ_jbhtRuARsFctynQFOTFigKWLovqMgUEP8llXQqo87oMYu4F4Pv8Tsx1I3DGTT2CW_TAjT1WKPmA80Q4qHpn-PgnAZuC0H-KC6ByWRpaSRoHd-4A0rL8DWqlZp  Este sería el tamaño de nuestra caja (sin considerar el margin):  Width: 110px  Height: 110px | **CSS**  div{  background: #5DEAA8;  height: 50px;  width: 50px;  padding: 20px;  border: 10px solid #43B17D;  margin: 60px;  box-sizing: content-box;    } |
| **Paddding-box**  Incluye el tamaño del padding dentro del tamaño establecido para la caja, es decir que el padding no sumara pixeles al tamaño de la caja, sino que el tamaño de la caja se reducirá para que al agregarle el padding la caja mantenga el tamaño establecido inicialmente, dejando fuera a border y margin. | |
| <div></div>  https://lh4.googleusercontent.com/91NU7g8ul3VFwULkJOfpNgHMjq6Cbizhm5NbL7CEiiUcJukUv5fdGqqAp-tbpd8PPqwxnxei33d0-8oaC_mffR1GkeBfKfqFq8BwWuH_TQoLoB4HRThzmixAEle8lALpmyN39k-b | **CSS**  div{  background: #5DEAA8;  height: 50px;  width: 50px;  padding: 20px;  border: 10px solid #43B17D;  margin: 60px;  box-sizing: padding-box;    } |
| **Border-box**  Al igual que el anterior disminuirá su tamaño agregando al padding y border para que el tamaño establecido de la caja se mantenga, dejando fuera al margin. | |
| <div></div>  https://lh6.googleusercontent.com/GngsEIF5KkHCotYymG_ilLeEdSaJ5cyhvY9XdkEs5TcatS0e0mnizhKIYcso2E6AsbFBZl8DxOIMucsPI0PddBxdGQc470Spj4c8owU63zqO1CJtQ7eFt2ake_WDt6OP0WJMNBIs | **CSS**  div{  background: #5DEAA8;  height: 100px;  width: 100px;  padding: 20px;  border: 10px solid #43B17D;  margin: 60px;  box-sizing: border-box;    } |
| Al igual que en el ejemplo anterior el tamaño de nuestra caja disminuye para agregar al *padding* y *border*, manteniendo así el tamaño inicial de la caja.  Podemos comprobar esto sumando el tamaño de la caja con el *padding* y *border*:  Width: 10+20+40+20+10 = 100px  Height: 10+20+40+20+10 = 100px | |
|  |  |

| **Lectura adicional**  **Content-box**  La propiedad width y height solo se refiere al contenido del bloque. Para calcular el espacio total ocupado se le suman borde, relleno y márgenes. Es el valor por defecto  **Border-box**  En esta opción la propiedad width y height ya contienen el grosor del borde y del padding. Para calcular el espacio total ocupado por la caja se suman solo los márgenes.  En este ejemplo puedes ver dos cajas padre, con borde gris, con unas dimensiones de 200px de ancho (width) y 100px de alto (height), en su interior hay un bloque de contenido de 200px de ancho. Ambas cajas padre tienen el mismo estilo, solo se diferencian en el valor de **box-sizing**:   * En el caso de usar **content-box** el bloque de contenido entra perfectamente en la caja padre, pues el borde y el relleno se añaden al ancho de la caja padre, se dibujan por fuera de la caja padre. En total ocupa un ancho de 200 (width) + 20 (bordes derecho e izquierdo) + 24(rellenos izquierdo y derecho). Al rededore estará la banda de los márgenes. * En caso de utiliza **border-box** el contenido sobrepasa el tamaño de la caja padre, pues los 200px del bloque padre inclluyen el borde y el padding, es como decir que border y padding se situan por dentro de la caja padre. En total ocupa un ancho de 200 px, rodeado or el ancho de los márgenes.    Margin y padding El margen de cualquier contenedor se define mediante la propiedad **margin**, una propiedad compuesta por cuatro valores uno para cada lado de la caja. Y se definen con valores numéricos (absolutos o porcentajes) o como automáticos (**auto**). Margenes de contenedor  Estos definiciones de la clase márgenes son equivalentes:  .margenes{  margin-top: 2px;  margin-rigth: 4px;  margin-bottom: 6px;  margin-left: 8px;  }  .margenes{  margin: 2px 4px 6px 8px;  }  Este es común a otras propiedades CSS, el atajo asigna valores según las agujas del reloj: top-left-bottom-right. Aún se puede acortar más si se van a utilizar valores repetidos:  margin: 8px; /\*todos los márgenes 8px de ancho\*/ margin: 8px 12px; /\*Margen superior e inferior 8px y laterales a 12px\*/ margin: 0px auto /\*Sin margenes superior e inferior y laterales en auto\*/  La última definición se suele usar a menudo para centrar bloques horizontalmente.  El relleno o padding es el espacio entre el contenido de un elemento y el borde, es también un elmento compuesto y se define con valores numéricos o porcentajes. No usa el valor auto, por lo demás se define de manera idéntica a los márgenes:  Margenes de contenedor |
| --- |