

{desafío}
latam_

Posicionamiento y layout _

Parte I



Proyecto a realizar

Maquetar entrada de blog



TOURS NOSOTROS BLOG CONTACTO

Los Imperdonables Santiaguinos

20 enero de 2018 - Wikixplora

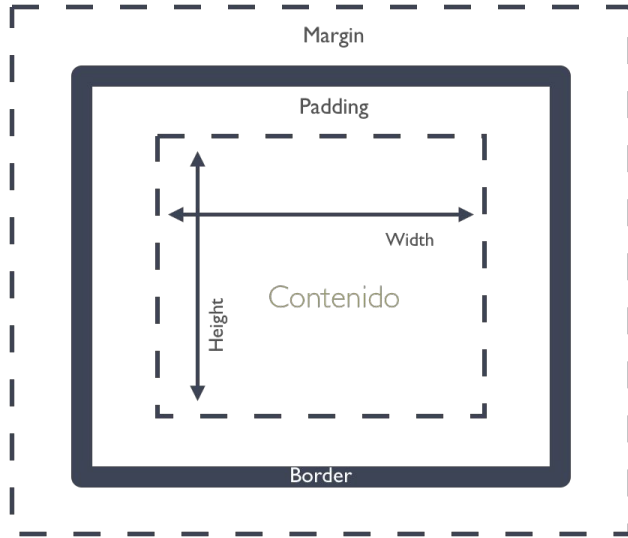


Vista de la cordillera de los Andes vista desde Santiago

- “Montaña” es una empresa de transporte dedicada al traslado de turistas de diversos puntos de interés alrededor de Chile.
- Nosotros deberemos terminar de maquetar la entrada usando el mockup de la vista y la guía de estilos.

Conociendo el modelo de cajas

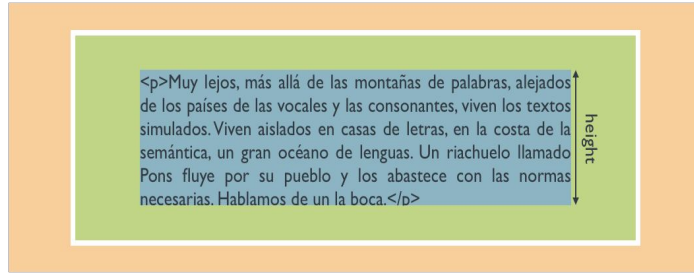
¿Qué es el modelo de cajas?



- El modelo de cajas nos ayuda a manejar las propiedades asociadas al tamaño, borde y relleno de los elementos incluidos en una página web.

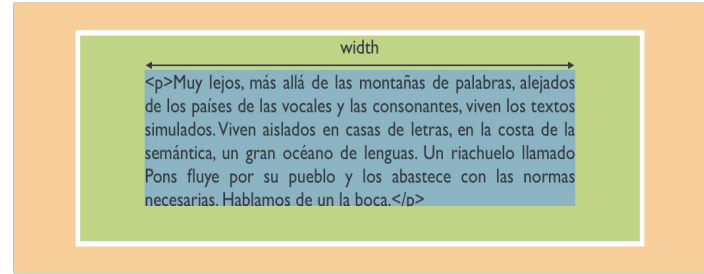
Dimensiones de una caja

Width



Indica cuánto ancho tenemos para agregar contenido dentro de nuestra caja.

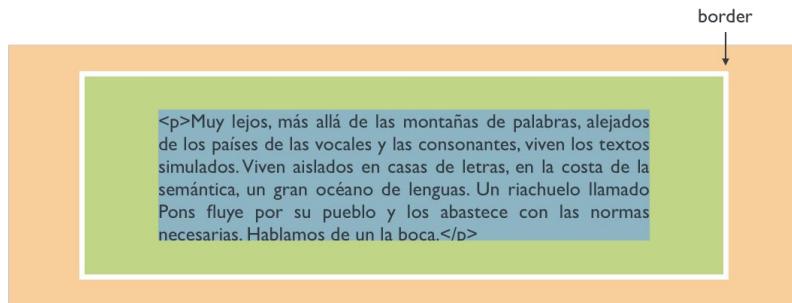
Height



Indica cuánto alto hay disponible para el contenido dentro de una caja.

Controlando el modelo de cajas

Border



- Nos permite especificar las características del borde de un elemento.
- Estas características pueden ser el estilo del borde, su ancho y color.

Border

Para agregar bordes a un elemento hay tres propiedades claves involucradas.

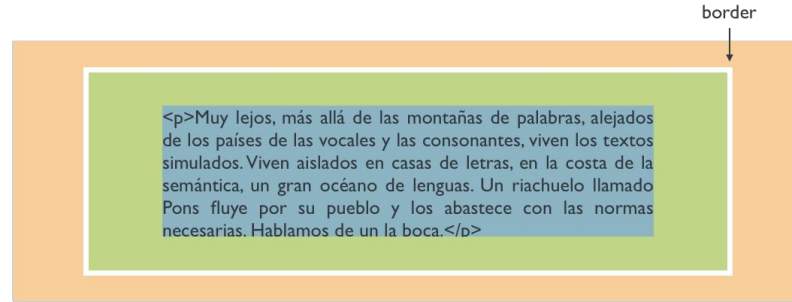
```
p {  
  border-style: solid;  
  border-width: 1px;  
  border-color: black;  
}
```

```
p {  
  border: solid 1px black;  
}
```

- **border-style:** Estilo de borde (dotted, dashed, solid, etc).
- **border-width:** Ancho de borde (px, cm, em, etc).
- **border-color:** Color de borde. (name, hex, RGB, transparent, RGBA).

Existe una forma corta para dar estos mismos valores usando border.

Padding



El *padding* es el espacio que hay entre el contenido y el borde dentro de un elemento.

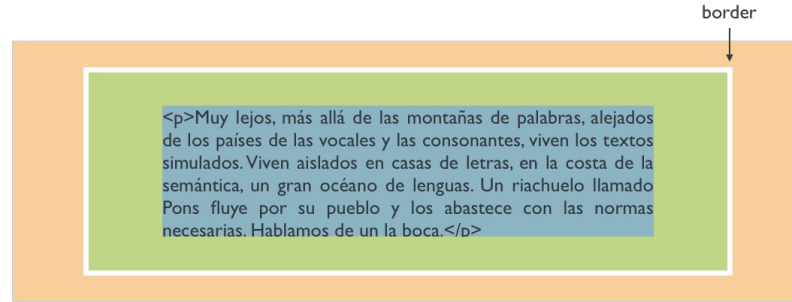
Padding

```
p {  
  padding: 200px 500px;  
}
```

```
p {  
  padding-left: 20px;  
  padding-top: 30px;  
}
```

- Existen diferentes propiedades que podemos usar para controlar el relleno del elemento.
- Cada una representa el lugar en donde se modificará, o sea, arriba, derecha, abajo e izquierda.
- Además, podremos usar una versión corta que incluye todas las reglas de padding que deberán seguir al sentido del reloj, o sea, arriba, derecha, abajo, izquierda.

Margin



Los márgenes de una caja son el espacio que hay entre el borde y los elementos que lo rodea.

Margin

```
p {  
  margin-top: 50px;  
  margin-bottom: 30px;  
}
```

```
p {  
  margin: 50px 100px;  
}
```

- Los márgenes, al igual que padding, tienen diferentes propiedades que podemos usar para controlar el margen.
- Cada una representa el lugar en donde se modificará el margen del elemento.
- Las opciones que tendremos serán margen hacia arriba, derecha, abajo e izquierda y una versión corta llamada margin.

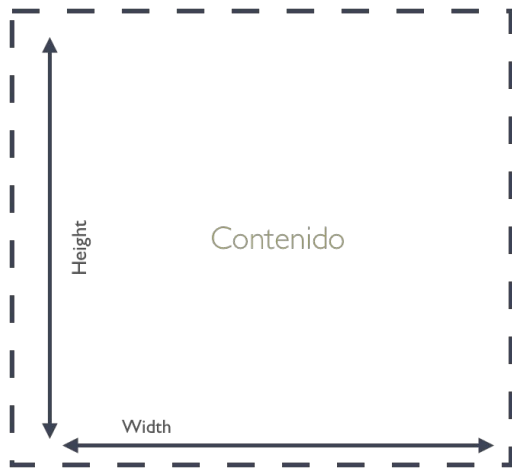
Calculando dimensiones de una caja

Box-sizing

```
div {  
  box-sizing: border-box;  
}
```

Con la propiedad *box-sizing* podremos definir cómo se calcula el ancho y alto de un elemento.

Box-sizing: content-box;

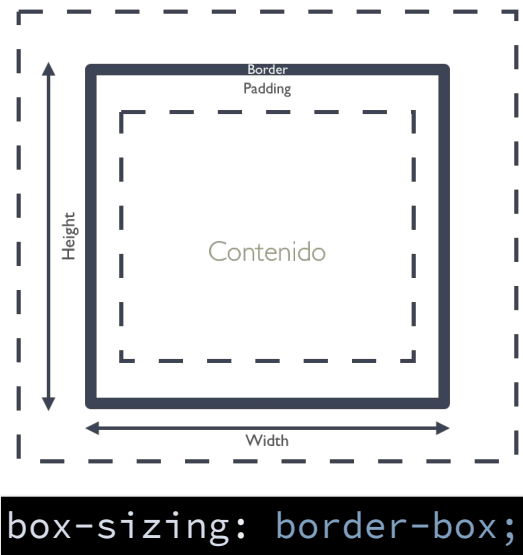


```
box-sizing:
```

```
content-box;
```

- Forma por defecto que el navegador calcula el modelo de cajas.
- Este valor suma solamente el alto y ancho definido en una caja, exceptuando el borde y padding.

Box-sizing: border-box;



- La propiedad `border-box` calcula el tamaño de la caja usando el alto y ancho, más el padding y el borde.

Alineando Elementos

Text-align

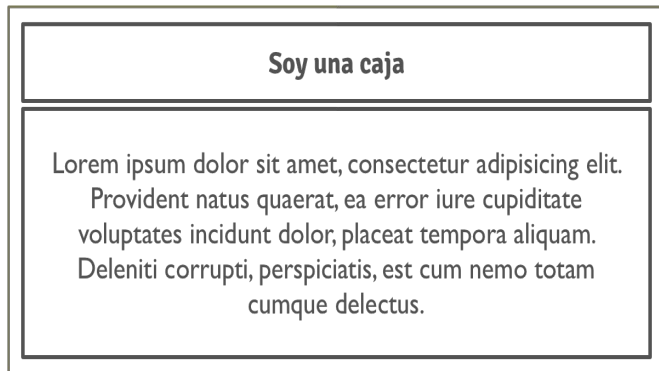
```
.container-button {  
  text-align: center;  
}
```

- Esta propiedad nos ayudará con el alineamiento del contenido dentro de un bloque.
- Las posibilidades que tenemos para posicionar el contenido dentro de una caja con esta propiedad son *left*, *center* y *right*.

Tipos de Caja

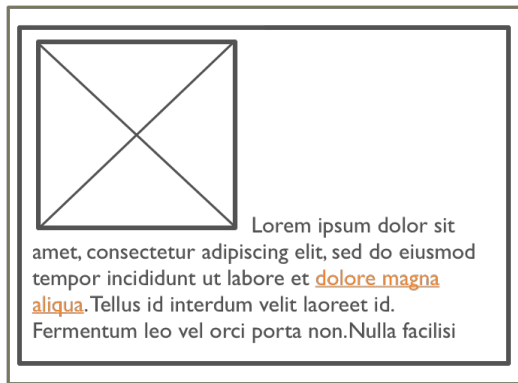
Los elementos que componen un documento *HTML*, por lo general, tienen un comportamiento que los define dentro del flujo del *HTML*. Este comportamiento depende del tipo de elemento que representa al elemento.

Elementos de bloque



- Los elementos de bloque siempre comienzan en una nueva línea.
- Ocupan todo el espacio disponible de la página, mientras no se defina un ancho para este elemento.
- Algunos ejemplos de elementos de bloque son las etiquetas: ``, `<h1>`, `<p>`, `<div>`, etc.

Elementos de línea



- Estos se caracterizan por fluir alrededor de un texto usando sólo el mínimo espacio.
- Alguno de los elementos HTML que tienen este comportamiento son las etiquetas: ****, ****, **<a>**, **<i>**, entre otras.

Cambiando el comportamiento de los elementos

Display

```
img {  
  display: block;  
}
```

- Con la propiedad *display* podremos cambiar el comportamiento de los elementos a modo de especificar el tipo de comportamiento que deseamos que tenga este dentro de un layout.

Display: block;

```
img {  
  display: block;  
}
```

- Hace que los elementos de línea se comporten como si fueran de bloque

Display: inline;

```
div {  
  display: inline;  
}
```

- Hace que los elementos de bloque se comporten como si fueran *inline*.

Display: inline-block;

```
p {  
  display:  
  inline-block;  
}
```

- Cambia el comportamiento los elementos de línea dándole atributos de elementos de bloque.
- Los valores *inline-block* permiten a los elementos en línea afectados agregar un **alto y/o ancho** dentro de otros elementos del mismo tipo

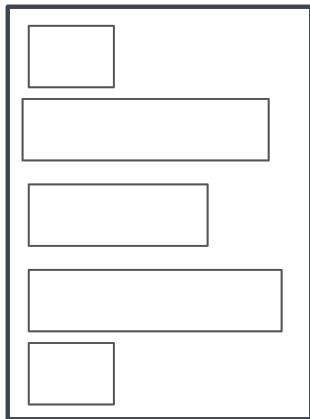
Conociendo el posicionamiento de elementos

Position

```
.container {  
  position: static;  
}
```

- Propiedad con la cual podremos especificar el tipo de posicionamiento a utilizar en un elemento.

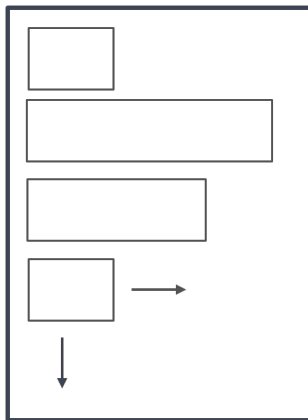
Position: static;



```
.container {  
  position: static;  
}
```

- Por defecto los navegadores utilizan este valor.
- Esto quiere decir que los elementos se renderizan o dibujan de acuerdo al flujo normal del *HTML*.
- Asimismo, cualquier elemento el cual no se especifique su posición será static.

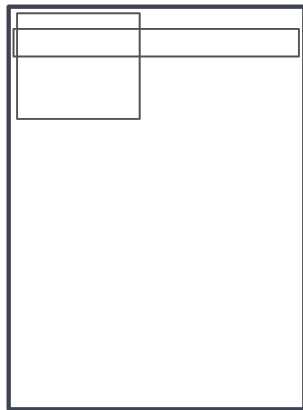
Position: relative;



```
.container {  
  position: relative;  
}
```

- Con las posiciones relativas podremos cambiar la posición de un elemento respecto al flujo normal del *HTML*.
- Podremos mover al elemento mediante las propiedades *top*, *right*, *bottom* y *left*.

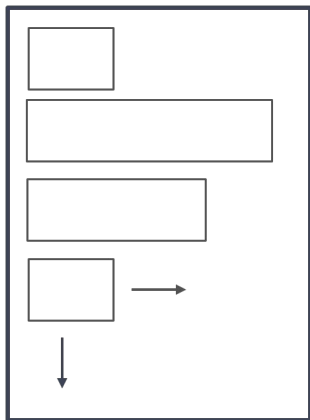
Position: absolute;



```
.container {  
  position: absolute;  
}
```

- Los valores absolutos nos permiten posicionar elementos en función a un elemento padre, siempre y cuando este tenga una posición distinta a static.
- Los elementos con posiciones absolutas se encuentran fuera del flujo normal del HTML, de modo que este no afecta a otros elementos posicionados en el documento.

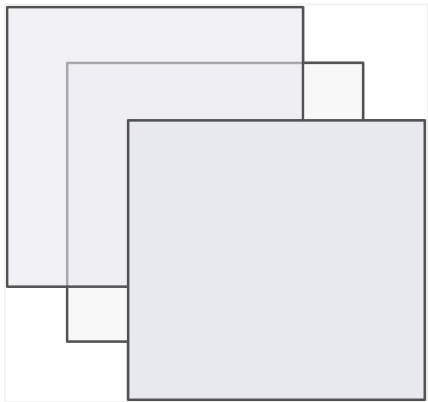
Position: fixed;



- Este valor permite fijar el contenido en función del *viewport*.

```
.container {  
  position: fixed;  
}
```

Z-index



```
.container {  
  z-index: 1;  
}
```

- Esta propiedad nos ayudará a definir si los elementos estarán encima o debajo de otros.
- Para hacerlo deberemos dar valores positivos para que estén arriba y negativos para que estén debajo.

{desafío}
latam_

*Academia de
talentos digitales*

www.desafiolatam.com