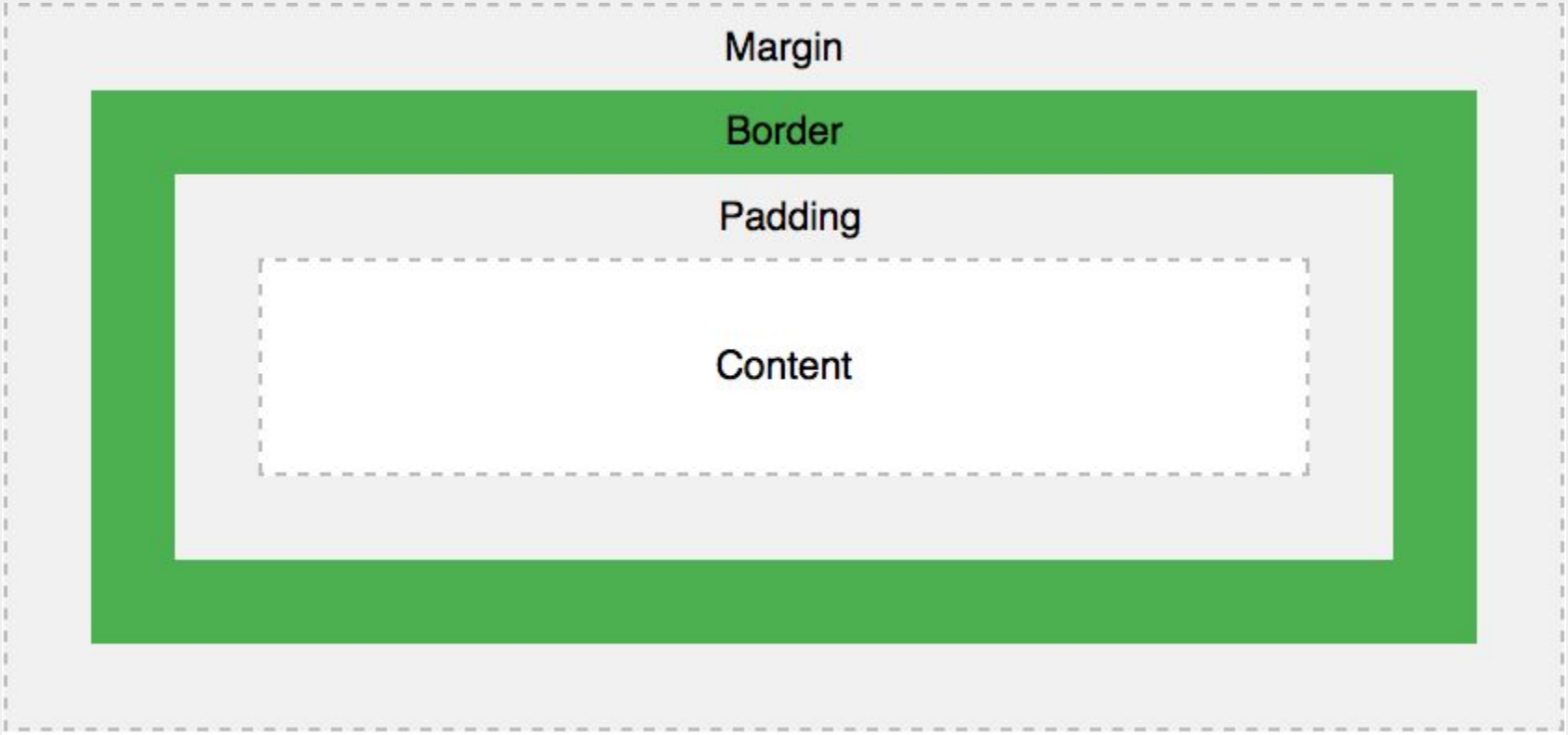
MAKE it REAL
.camp

classroom session

MAKE it REAL .camp

Box Model



Padding

MAKE it REAL .camp

- *Las propiedades de padding CSS se utilizan para generar espacio alrededor del contenido.*

- *El padding borra un área alrededor del contenido (dentro del borde) de un elemento.*

- *Con CSS, usted tiene control total sobre el padding. Hay propiedades CSS para configurar el padding para cada lado de un elemento (arriba, derecha, abajo y izquierda).*

Padding

MAKE it REAL .camp

Lados individuales

```
p {
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
```

Padding

MAKE it REAL .camp

Lados individuales

```
div.ex1 {
    padding: 25px 50px 75px 100px;
}

div.ex2 {
    padding: 25px 50px 75px;
}

div.ex3 {
    padding: 25px 50px;
}

div.ex4 {
    padding: 25px;
}
```

Height - Width

- *La altura y el ancho pueden ajustarse a auto (esto es el valor predeterminado, significa que el navegador calcula la altura y el ancho), o se especifica en valores de longitud, como px, cm, etc., o en porcentaje (%) del bloque que contiene .*

Height - Width

MAKE it REAL .camp

```
div {
    height: 200px;
    width: 50%;
    background-color: powderblue;
}
```

Height - Width

MAKE it REAL .camp

```
div {
    height: 200px;
    width: 50%;
    background-color: powderblue;
}
```

```
div {
    height: 100px;
    width: 500px;
    background-color: powderblue;
}
```

Max-width

MAKE it REAL .camp

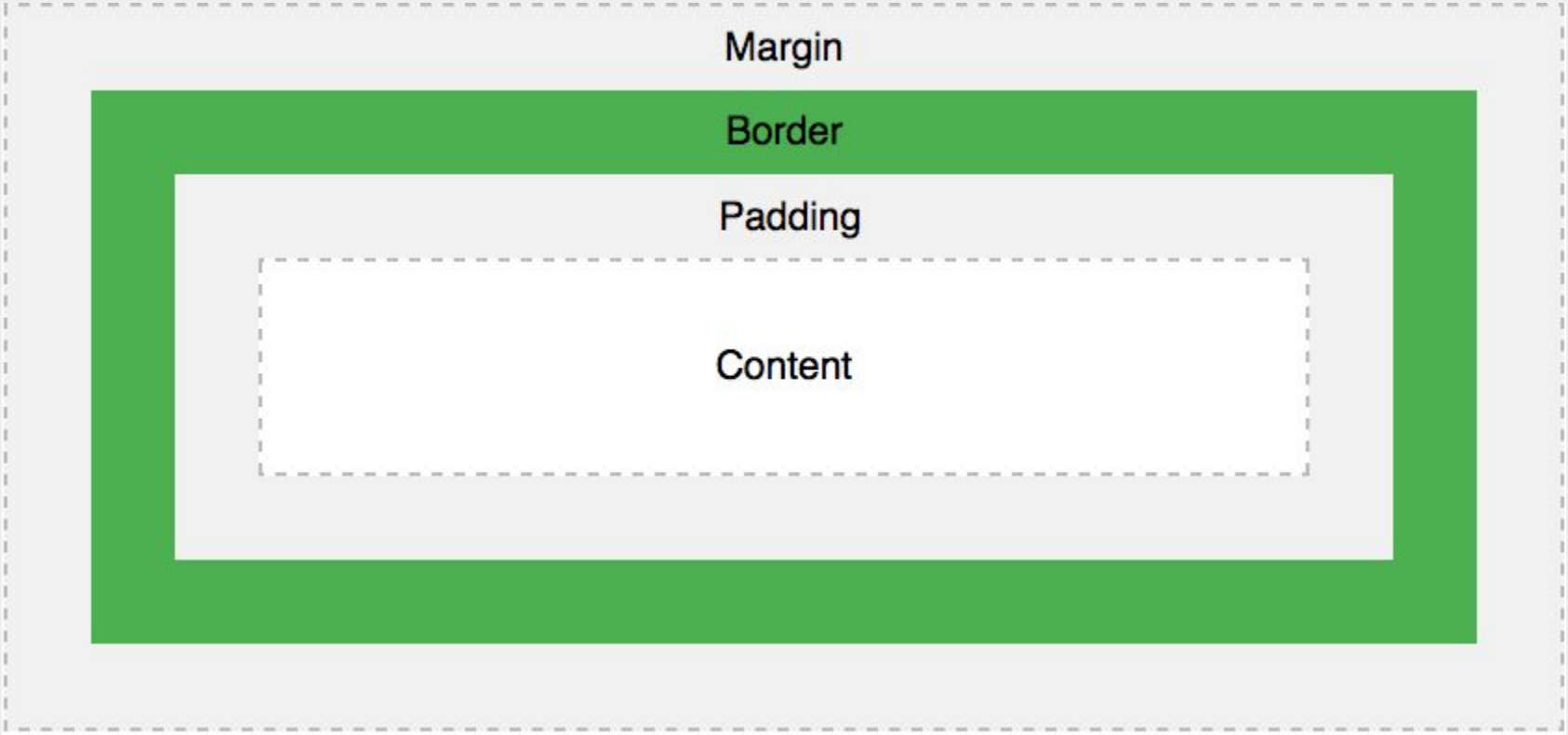
- *La propiedad max-width se utiliza para establecer el ancho máximo de un elemento.*
- *La anchura máxima se puede especificar en valores de longitud, como px, cm, etc., o en porcentaje (%) del bloque que contiene, o se establece en ninguno (esto es el valor predeterminado.*
- *El problema con el anterior se produce cuando la ventana del navegador es menor que el ancho del elemento (500px). A continuación, el navegador agrega una barra de desplazamiento horizontal a la página.*
- *El uso de max-width en su lugar, en esta situación, mejorará el manejo del navegador de pequeñas ventanas.*

Max-width

MAKE it REAL .camp

```
div {
    max-width: 500px;
    height: 100px;
    background-color: powderblue;
}
```

Box Model



Box Model

MAKE it REAL .camp

```
div {
    width: 300px;
    border: 25px solid green;
    padding: 25px;
    margin: 25px;
}
```

Box Model

MAKE it REAL .camp

- *Con el fin de establecer el ancho y la altura de un elemento correctamente en todos los navegadores, es necesario saber cómo funciona el modelo de caja.*

- *Importante: Cuando establece las propiedades de ancho y alto de un elemento con CSS, sólo tiene que definir el ancho y la altura del área de contenido. Para calcular el tamaño completo de un elemento, también debe agregar padding, bordes y márgenes.*

- *Supongamos que queremos diseñar un elemento para tener un ancho total de 350 px:*

Box Model

MAKE it REAL .camp

```
div {
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;
}
```

Box Model

MAKE it REAL .camp

```
div {
    width: 320px;
    padding: 10px;
    border: 5px solid gray;
    margin: 0;
}
```

320px (width)
+ 20px (left + right padding)
+ 10px (left + right border)
+ 0px (left + right margin)
= 350px

Box Model

MAKE it REAL .camp

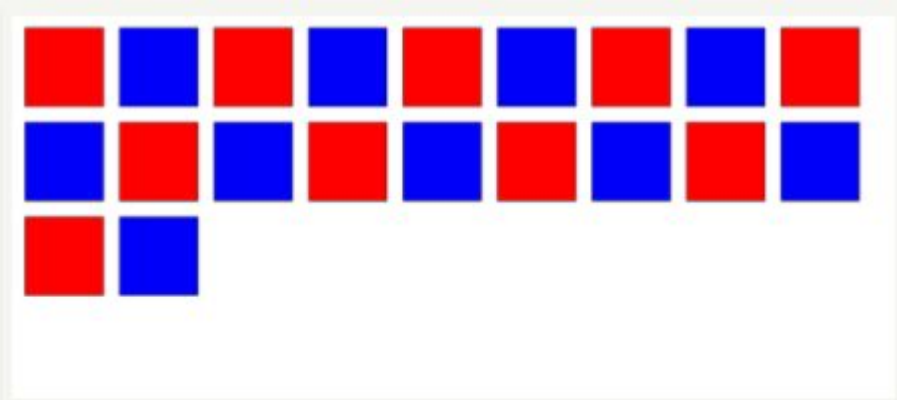
The total width of an element should be calculated like this:

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

The total height of an element should be calculated like this:

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

Challenge



Los cuadrados se van reposicionando a medida que se cambia el tamaño del navegador (pero el tamaño de los cuadrados se mantiene igual).

Challenge

MAKE it REAL .camp

Instrucciones

1. Agrega la estructura que debe tener todo HTML.
2. El título de la página debe ser **Cuadrados**.
3. Son 20 cuadrados. Cada cuadrado deben ser un `div`.
4. Cada cuadrado debe tener un ancho y alto de `100px`, un margin de `10px` y un borde negro de `1px`.
5. Los cuadrados impares deben tener un fondo de color rojo y los pares azul.

Text CSS

MAKE it REAL .camp

Text Color

```
body {
    color: blue;
}

h1 {
    color: green;
}
```

Text CSS

MAKE it REAL .camp

Text Alignment

```css
h1 {
    text-align: center;
}

h2 {
    text-align: left;
}

h3 {
    text-align: right;
}
```

```css
div {
    text-align: justify;
}
```

Text CSS

MAKE it REAL .camp

Text Decoration

```css
h1 {
    text-decoration: overline;
}

h2 {
    text-decoration: line-through;
}

h3 {
    text-decoration: underline;
}
```

```css
a {
    text-decoration: none;
}
```

Text CSS

MAKE it REAL .camp

Text Transformation

```
p.uppercase {
    text-transform: uppercase;
}

p.lowercase {
    text-transform: lowercase;
}

p.capitalize {
    text-transform: capitalize;
}
```

Text CSS

MAKE it REAL .camp

Text Indentation

```
p {
    text-indent: 50px;
}
```

Text CSS

MAKE it REAL .camp

Text Indentation

```
h1 {
    letter-spacing: 3px;
}

h2 {
    letter-spacing: -3px;
}
```

Text CSS

MAKE it REAL .camp

Line Height

```
p.small {
    line-height: 0.8;
}

p.big {
    line-height: 1.8;
}
```

Text CSS

MAKE it REAL .camp

Text Shadow

```
h1 {
    text-shadow: 3px 2px red;
}
```

CSS Fonts

MAKE it REAL .camp

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width

CSS Fonts

MAKE it REAL .camp

Font Family

```
p {
    font-family: "Times New Roman", Times, serif;
}
```

CSS Fonts

MAKE it REAL .camp

Font Style

```
p.normal {
    font-style: normal;
}

p.italic {
    font-style: italic;
}

p.oblique {
    font-style: oblique;
}
```

CSS Fonts

MAKE it REAL .camp

Font Size

```
h1 {
    font-size: 40px;
}

h2 {
    font-size: 30px;
}

p {
    font-size: 14px;
}
```

CSS Fonts

MAKE it REAL .camp

Font Weight

```
p.normal {
    font-weight: normal;
}

p.thick {
    font-weight: bold;
}
```

CSS Icons

MAKE it REAL .camp

Font-Awesome

CSS Icons

MAKE it REAL .camp

Font-Awesome

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>

<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars"></i>

</body>
</html>
```

CSS Links

MAKE it REAL .camp

Estilo para los links

Links can be styled with any CSS property (e.g. `color`, `font-family`, `background`, etc.).

Example

```
a {
    color: hotpink;
}
```

CSS Links

MAKE it REAL .camp

Según estado de los Links

The four links states are:

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

CSS Links

MAKE it REAL .camp

Según estado de los Links

```
/* unvisited link */
a:link {
    color: red;
}

/* visited link */
a:visited {
    color: green;
}

/* mouse over link */
a:hover {
    color: hotpink;
}

/* selected link */
a:active {
    color: blue;
}
```

CSS Links

MAKE it REAL .camp

Text decoration

```
a:link {
    text-decoration: none;
}

a:visited {
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

a:active {
    text-decoration: underline;
}
```

CSS Links

MAKE it REAL .camp

Background color

```
a:link {
    text-decoration: none;
}

a:visited {
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

a:active {
    text-decoration: underline;
}
```

CSS Tables

MAKE it REAL .camp

Borders

```
table, th, td {
   border: 1px solid black;
}
```

```
table {
    border-collapse: collapse;
}

table, th, td {
    border: 1px solid black;
}
```

CSS Display

MAKE it REAL .camp

- La propiedad de display es la propiedad CSS más importante para controlar el layout (diseño).

- La propiedad display especifica si y cómo se muestra un elemento.

- Cada elemento HTML tiene un valor de display predeterminado dependiendo del tipo de elemento que es. El valor de display predeterminado para la mayoría de los elementos es “block” o “inline”.

Block-level elements

Un elemento de nivel de bloque siempre comienza en una nueva línea y ocupa todo el ancho disponible (se extiende hacia la izquierda y la derecha hasta donde puede).

Block-level elements

MAKE it REAL .camp

The element is a block-level element.

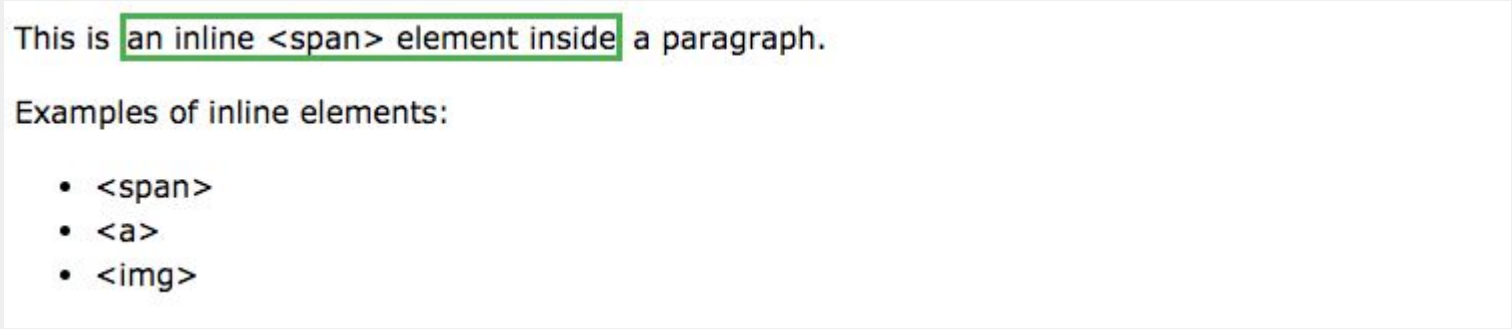
Examples of block-level elements:

-
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

Inline elements

MAKE it REAL .camp

Un elemento en línea no se inicia en una nueva línea y sólo ocupa tanto ancho como sea necesario.



CSS Display

MAKE it REAL .camp

display: none;

- Se utiliza comúnmente con JavaScript para ocultar y mostrar elementos sin eliminarlos ni volverlos a crear. El elemento <script> utiliza display: none; por defecto.

CSS Display

MAKE it REAL .camp

Sobre-escribir el valor de display predeterminado

- Como se mencionó, cada elemento tiene un valor display predeterminado. Sin embargo, podemos sobre-escribirlo.

- Cambiar un elemento inline a un elemento block, o viceversa, puede ser útil para hacer que la página se vea de una manera específica y siga los estándares web.

CSS Display

MAKE it REAL .camp

```css
li {
    display: inline;
}
```

```css
span {
    display: block;
}
```

```css
a {
    display: block;
}
```

```css
h1.hidden {
    display: none;
}
```

CSS Position

MAKE it REAL .camp

There are four different position values:

- `static`
- `relative`
- `fixed`
- `absolute`

CSS Position

MAKE it REAL .camp

stactic

- Los elementos HTML se colocan estáticos de forma predeterminada.

- Los elementos posicionados estáticamente no se ven afectados por las propiedades superior, inferior, izquierda y derecha.

- Un elemento con la posición: estático; No se coloca de ninguna manera especial; Siempre se posiciona de acuerdo con el flujo normal de la página:

CSS Position

MAKE it REAL .camp

stactic

```
div.static {
    position: static;
    border: 3px solid #73AD21;
}
```

CSS Position

MAKE it REAL .camp

relative

- Un elemento con la posición: relative; Está situado en relación con su posición normal.

- Ajustar las propiedades superior, derecha, inferior e izquierda de un elemento relativamente posicionado hará que se ajuste lejos de su posición normal. El resto del contenido no se ajustará para caber en cualquier hueco que deje el elemento.

CSS Position

MAKE it REAL .camp

relative

```
div.relative {
    position: relative;
    left: 30px;
    border: 3px solid #73AD21;
}
```

CSS Position

MAKE it REAL .camp

fixed

- Un elemento con posición: fijo; Se posiciona en relación con la ventana de visualización, lo que significa que siempre permanece en el mismo lugar incluso si la página se desplaza. Las propiedades superior, derecha, inferior e izquierda se utilizan para posicionar el elemento.

- Un elemento fijo no deja un hueco en la página donde normalmente se habría localizado.

CSS Position

MAKE it REAL .camp

fixed

```
div.fixed {
    position: fixed;
    bottom: 0;
    right: 0;
    width: 300px;
    border: 3px solid #73AD21;
}
```

CSS Position

MAKE it REAL .camp

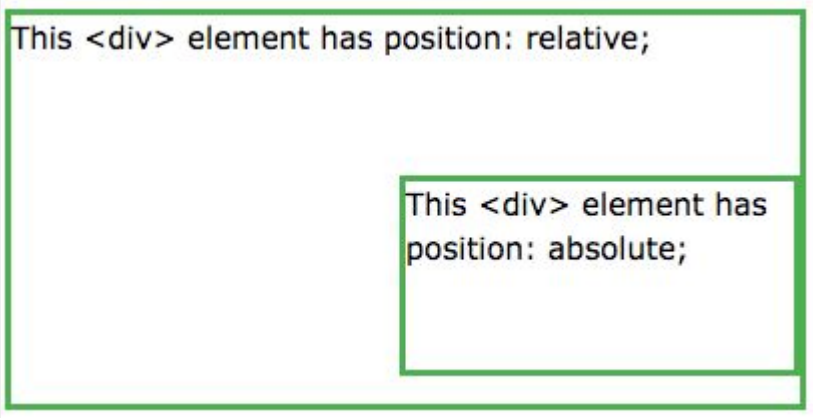
absolute

- Un elemento con la posición: absolute; Se posiciona en relación con el antepasado posicionado más próximo (en lugar de posicionado con respecto a la ventana gráfica, como fijo).
- Sin embargo; Si un elemento posicionado absolute no tiene ancestros posicionados, utiliza el cuerpo del documento y se mueve junto con el desplazamiento de la página.
- Nota: Un elemento "posicionado" es aquel cuya posición es cualquier cosa excepto estática.

CSS Position

MAKE it REAL .camp

absolute



CSS Position

MAKE it REAL .camp

absolute

```
div.relative {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;
}

div.absolute {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
}
```

CSS Float y clear

MAKE it REAL .camp

```
div {
    clear: left;
}
```

```
div {
    clear: left;
}
```

CSS Align

MAKE it REAL .camp

center horizontal

```
.center {
    margin: auto;
    width: 50%;
    border: 3px solid green;
    padding: 10px;
}
```

```
.center {
    text-align: center;
    border: 3px solid green;
}
```

CSS Align

MAKE it REAL .camp

center vertical - usando padding

```
.center {
    padding: 70px 0;
    border: 3px solid green;
}
```

CSS Combinators

MAKE it REAL .camp

- Un combinador es algo que explica la relación entre los selectores.
- Un selector CSS puede contener más de un selector simple. Entre los selectores simples, podemos incluir un combinador.
- Hay cuatro combinadores diferentes en CSS3:
  - Selector de descendientes (espacio)
  - Selector hijo (>)
  - Selector de hermanos adyacente (+)
  - Selector hermano (~)

CSS Selector Descendente

MAKE it REAL .camp

```
div p {
    background-color: yellow;
}
```

CSS Selector Hijo

MAKE it REAL .camp

```
div > p {
    background-color: yellow;
}
```

CSS Selector Hermanos Adyacentes

MAKE it REAL .camp

```css
div + p {
    background-color: yellow;
}
```

CSS Selector Hermano

MAKE it REAL .camp

```
div ~ p {
    background-color: yellow;
}
```

classroom rating