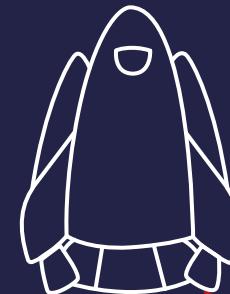
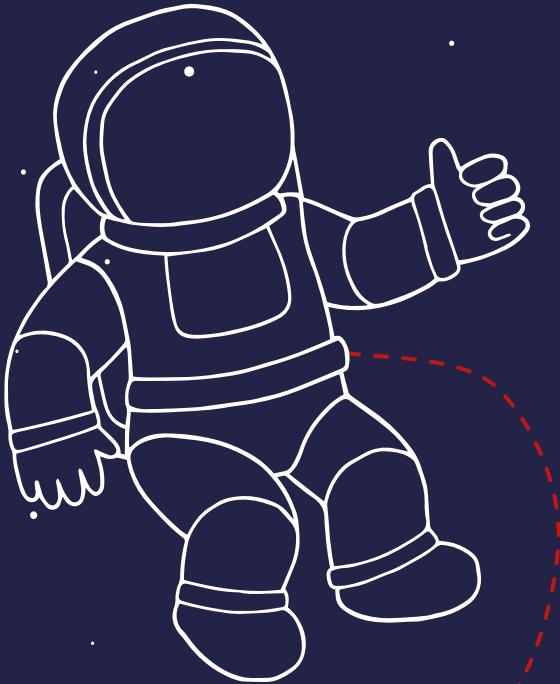




Programa académico **CAMPUS**

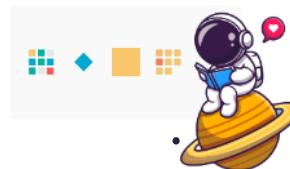
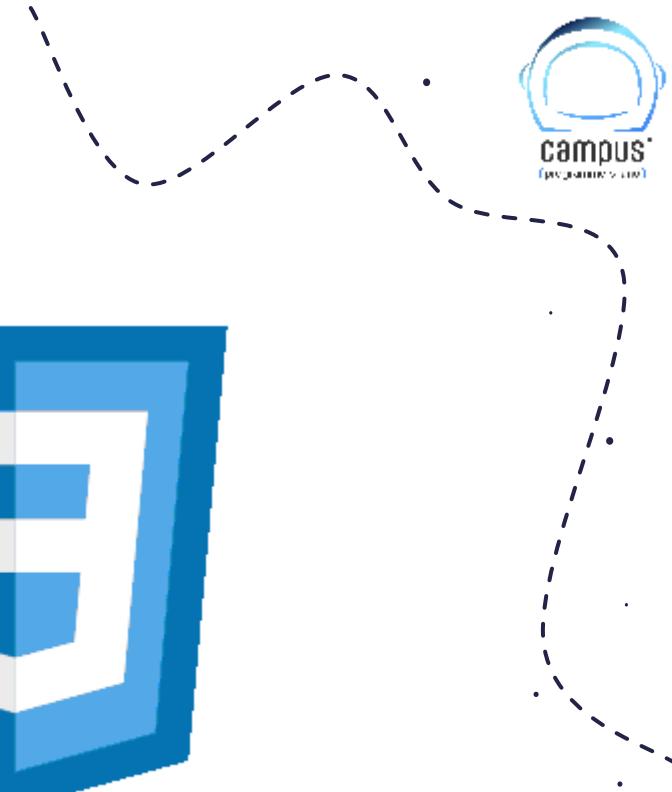
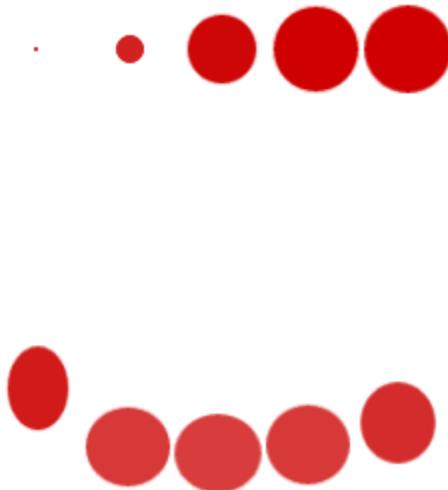
**Ciclo 2:
Box-sizing,
display,
visibilidad y
posiciónamiento**



CSS



CSS the Skin



Propiedad box-sizing



- Permite controlar el cálculo que hace el navegador a la hora de modificar las propiedades content, padding y border.

CSS



CSS the Skin





Propiedad box-sizing

```
<style>
    .red-box {
        background-color: red;
        height: 400px;
        width: 100%;
        /*border: 10px solid blue;*/
        /*No deja pasar las dimensiones
        del navegador*/
        box-sizing: border-box;
        padding: 0 5%;
    }
</style>
</head>
<body>
    <div class="red-box"></div>
</body>
```

CSS



CSS the Skin

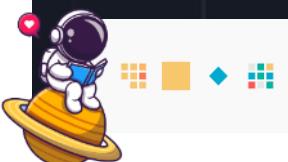


Propiedad box-sizing

Box-sizing:

Es la propiedad que nos permite controlar el cálculo que hace el navegador a la hora de modificar las propiedades content, padding y border.

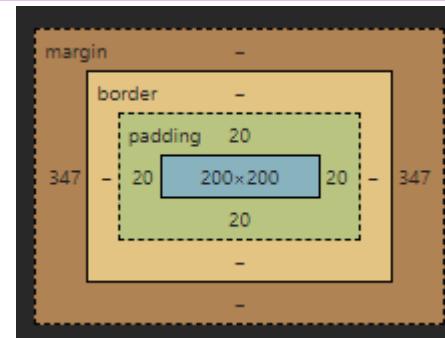
*Los 2 valores que podemos darle son
content-box -> Valor por defecto
border-box -> Cálculo de tamaño del elemento incluyendo el padding y el border.*



Propiedad box-sizing

```
<html>
  <head>
    <style>
      .red-box {
        background-color: purple;
        color: white;
        height: 200px;
        width: 200px;
        margin-left: auto;
        margin-right: auto;
        padding: 20px;
      }
    </style>
  </head>
  <body>
    <div class="red-box">
      Caja
    </div>
  </body>

```



CSS

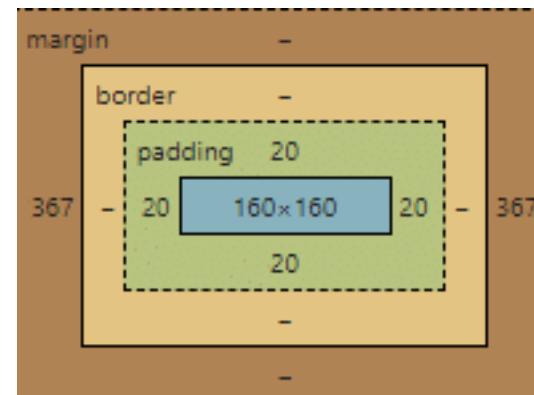
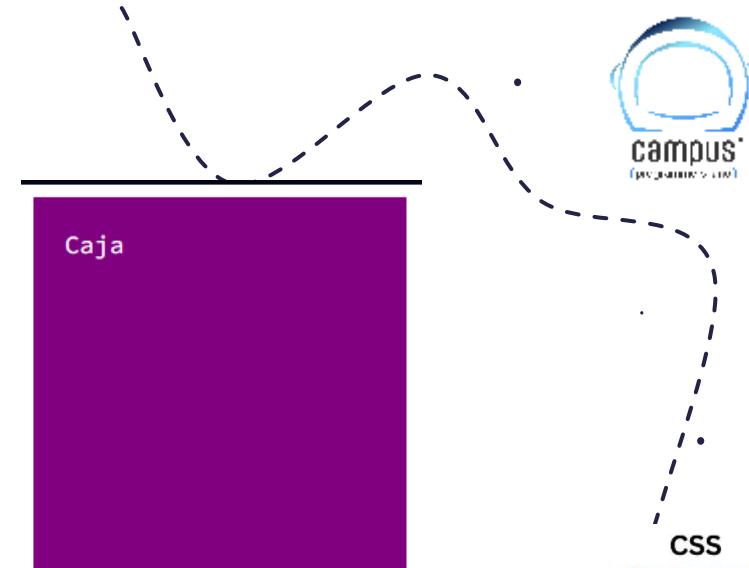


CSS the Skin



Propiedad box-sizing

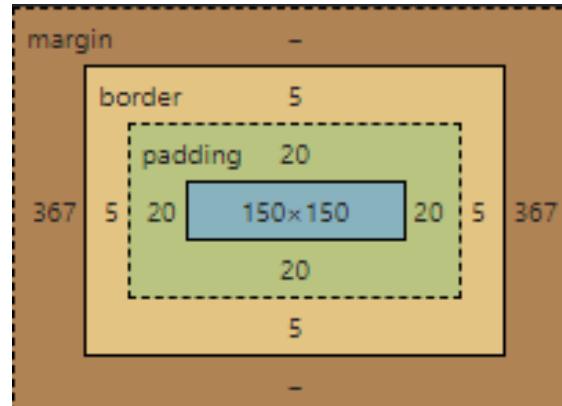
```
background-color: purple;  
width: 200px;  
height: 200px;  
margin-left: auto;  
margin-right: auto;  
padding: 20px;  
box-sizing: border-box;
```



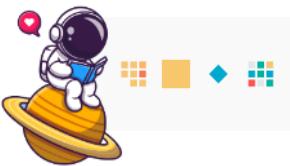


Propiedad box-sizing

- ```
background-color: #purple;
color: white;
height: 200px;
width: 200px;
```
- ```
margin-left: auto;
margin-right: auto;
box-sizing: border-box;
border: 5px solid #red;
```
- ```
padding: 20px;
```



CSS the Skin





# Propiedad display

- La propiedad **display** de CSS permite modificar el comportamiento de un elemento HTML,
- cambiándolo al que le indiquemos, como por ejemplo **inline** o **block** (*u otro de los que veremos más adelante*).

css



CSS the Skin



# Propiedad display

| Tipo de caja              | Características                                                                                                                   |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <code>block</code>        | Se apila en vertical. Ocupa todo el ancho disponible de su etiqueta contenedora.                                                  |
| Versión en línea          |                                                                                                                                   |
| <code>inline</code>       | Se coloca en horizontal. Se adapta al ancho de su contenido.<br>Ignora <code>width</code> o <code>height</code> .                 |
| <code>inline-block</code> | Combinación de los dos anteriores. Se comporta como <code>inline</code> pero no ignora <code>width</code> o <code>height</code> . |

CSS



CSS the Skin



# Propiedad display

## Versión flexible

`flex`

Utiliza el modelo de cajas flexibles de CSS. Ideal para estructuras de 1 dimensión.

`inline-flex`

Versión en línea (ocupa sólo su contenido) del modelo de cajas flexibles de CSS.

## Versión de cuadrículas

`grid`

Utiliza cuadrículas o rejillas con el modelo de cajas Grid CSS.

`inline-grid`

La versión en línea (ocupa sólo su contenido) del modelo de cajas grid css.

## Versión de listas y tablas



CSS



CSS the Skin



# Propiedad display



| Versión de listas y tablas |                                                                                                                              |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <code>list-item</code>     | Actúa como un ítem de una lista. Es el comportamiento de etiquetas como <code>&lt;li&gt;</code> .                            |
| <code>table</code>         | Actúa como una tabla. Es el comportamiento de etiquetas como <code>&lt;table&gt;</code> .                                    |
| <code>table-cell</code>    | Actúa como la celda de una tabla. Es el comportamiento de etiquetas como <code>&lt;th&gt;</code> o <code>&lt;td&gt;</code> . |
| <code>table-row</code>     | Actúa como la fila de una tabla. Es el comportamiento de etiquetas como <code>&lt;tr&gt;</code> .                            |
| Otros                      |                                                                                                                              |
| <code>contents</code>      | Ignora la caja del elemento. Útil para mantener Grid/Flex aún teniendo un wrapper intermedio.                                |
| <code>none</code>          | Oculta el elemento visualmente, como si no existiera en el HTML.                                                             |



# Propiedad display



127.0.0.1:5500/block/index... Invitado

## Soy un elemento Bloque

Soy otro elemento bloque

```
<!DOCTYPE html>
<html>
 <head>
 <title>Soy un elemento Bloque</title>
 </head>
 <body>
 <h2 class="title">Soy un elemento Bloque</h2>
 <p class="paragraph">Soy otro elemento bloque</p>
 <!-- Code injected by live-server -->
 <script>...</script>
 </body>
</html>
```

html body p.paragraph

Styles Computed Layout Event Listeners DOM Breakpoints Properties >

Filter

:hover .cls + ↻ ↺ user agent stylesheets

p {

```
display: block;
margin-block-start: 1em;
```



CSS the Skin



# Visibilidad



- Existen múltiples formas de ocultar elementos del HTML, de forma que se ocultan visualmente pero el marcado de etiquetas HTML continua existiendo en la página.

CSS



CSS the Skin



# Visibilidad



Propiedad	Valores	Descripción
<code>display</code>	<code>none</code>	Oculta de forma visual y por completo el elemento indicado.
<code>visibility</code>	<code>visible   hidden   collapse*</code>	Oculta un elemento manteniendo su espacio visual.
<code>opacity</code>	<code>1  </code>	Opacidad del elemento, medido en valores numéricos o porcentajes.

CSS



CSS the Skin





# La propiedad display: none

- Esto hace que el navegador ignore por completo ese elemento, no mostrándolo visualmente.

CSS



CSS the Skin

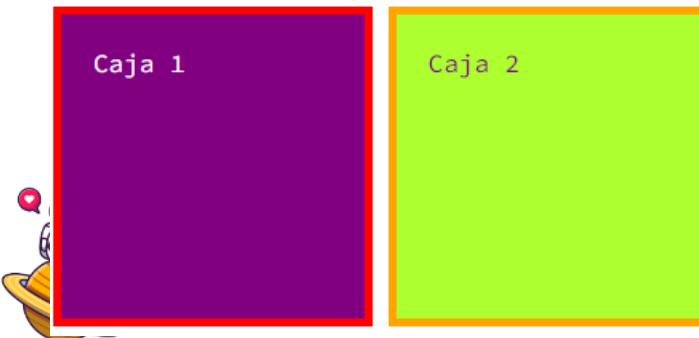


# La propiedad display: none



```
<body>
 <div class="red-box">
 Caja 1
 </div>

 <div class="green-box">
 Caja 2
 </div>
</body>
```



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

.red-box {
 background-color: purple;
 color: white;
 height: 200px;
 width: 200px;
 border: 5px solid red;

 padding: 20px;
 display: inline-block;
}

.green-box {
 background-color: greenyellow;
 color: purple;
 height: 200px;
 width: 200px;
 border: 5px solid orange;

 padding: 20px;
 display: inline-block;
}
```



CSS the Skin





# La propiedad display: none

```
.red-box {
 background-color: purple;
 color: white;
 height: 200px;
 width: 200px;
 border: 5px solid red;

 padding: 20px;
 display: none;
}
```



CSS



CSS the Skin





# La propiedad visibility

- Existe una propiedad CSS llamada **visibility** que podemos utilizar con el valor **hidden**. Realiza una acción similar a **display: none** con la ligera diferencia de que no sólo oculta el elemento, sino que además mantiene el «hueco», el espacio vacío (**con el mismo tamaño que cuando se mostraba**)



css  
the Skin



# La propiedad **visibility**

- Dicha propiedad **visibility** tiene los siguientes valores posibles:

Valor	Significado
<code>visible</code>	El elemento es visible. Valor por defecto.
<code>hidden</code>	El elemento no es visible pero sigue ocupando su espacio y posición.
<code>collapse</code>	Sólo para tablas. El elemento se contrae para no ocupar espacio.



CSS



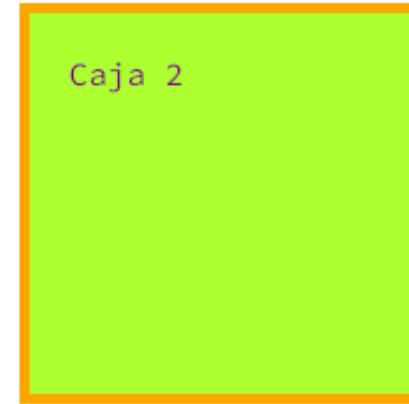
CSS the Skin



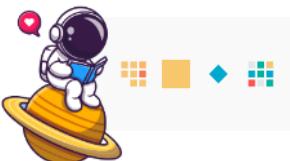
# La propiedad visibility

```
.red-box {
background-color: purple;
color: white;
height: 200px;
width: 200px;
border: 5px solid red;

padding: 20px;
display: inline-block;
visibility: hidden;
}
```



CSS the Skin



# Propiedad position



- Hasta ahora, hemos estado trabajando sin saberlo en lo que se denomina posicionamiento **estático** (**static**), donde todos los elementos aparecen con un orden natural según donde estén colocados en el HTML.

css



CSS the Skin



# Propiedad position

- A la propiedad **position** se le pueden indicar los siguientes valores:

Valor	Significado
static	Posicionamiento estático. Utiliza el orden natural de los elementos HTML.
relative	Posicionamiento relativo. Los elementos se mueven ligeramente en base a su posición estática.
absolute	Posicionamiento absoluto. Los elementos se colocan en base al contenedor padre.
fixed	Posicionamiento fijo. Idem al absoluto, pero aunque hagamos scroll no se mueve.
sticky	Posicionamiento «pegado». Similar al relativo, usado para pegar menús a la parte superior.





# Propiedad position

```
<h1>css Position</h1>
<div class="parent">
 <div class="static">Static</div>
 <div class="relative">Relative</div>
 <div class="absolute">Abosolute</div>
 <div class="fixed">Fixed</div>
 <div class="sticky">Stikcky</div>
</div>
```

css



CSS the Skin



# Propiedad position



```
* {
 box-sizing: border-box
}

body {
 margin: 2rem;
 font-size: 2rem;
 font-family: sans-serif;
 background-color: black;
}

h1 {
 color: white;
 font-size: 3rem;
 text-align: center;
}

div {
 border-radius: 5px;
}
```



```
.static {
 background-color: #dddddd;
}

.relative {
 background-color: #81c476;
}

.absolute {
 background-color: #fc9191;
}

.fixed {
 background-color: #a8d5e2;
}

.sticky {
 background-color: #e1e1ac;
}
```



CSS the Skin



css



# Propiedad position

## css Position

Static

Relative

Absolute

Fixed

Stikcky

CSS



CSS the Skin



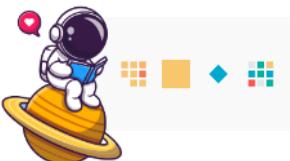
# Propiedad position



CSS



CSS the Skin



# Position static



- Es la posición por defecto de los elementos.

```
/* CSS Position */
.static {
 position: static;
}
```



# Position relative



- En este modo, los elementos se colocan exactamente igual que en el posicionamiento estático (*permanecen en la misma posición*), pero dependiendo del valor de las propiedades **top**, **bottom**, **left** o **right** variaremos ligeramente la posición del elemento.



CSS the Skin



# Position relative

```
relative {
 position: relative;
 top: 0;
 bottom: 0;
 left: 0;
 right: 0; }
}
```



CSS

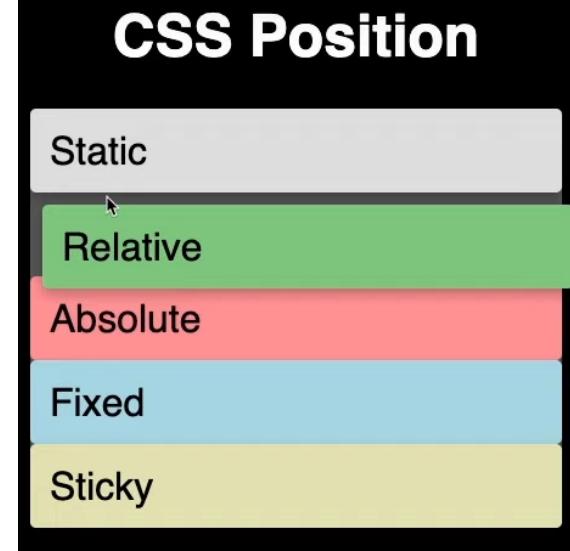


CSS the Skin



# Position relative

```
relative {
 position: relative;
 top: 10px;
 bottom: 0;
 left: 10px;
 right: 0;
}
```



CSS



CSS the SkIn



# Position absolute



- este tipo de posicionamiento coloca los elementos **utilizando como punto de origen el primer contenedor con posicionamiento diferente a estático.**

CSS



CSS the Skin



# Position absolute



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

## CSS Position

Static

Relative

Absolute

Sticky

CSS



CSS the Skin



# Position absolute

- ```
absolute {  
  position: absolute;  
  top: 10px;  
}
```
- Se posiciona con respecto al primer ancestro posicionado (*diferente de static*)



CSS



CSS the Skink



Position absolute



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

CSS Position

Absolute

Relative

Fixed

Sticky

CSS



CSS the Skin



Position fixed



- El **posicionamiento fijo** es hermano del **posicionamiento absoluto**. Funciona exactamente igual, salvo que hace que el elemento se muestre en una posición fija **dependiendo de la región visual del navegador**. Es decir, aunque el usuario haga scroll y se desplace hacia abajo en la página web, el elemento seguirá en el mismo sitio posicionado.

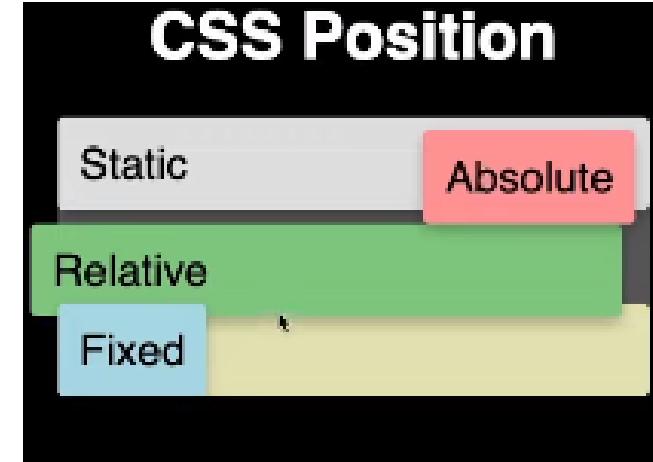


CSS the Skin



Position fixed

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



CSS



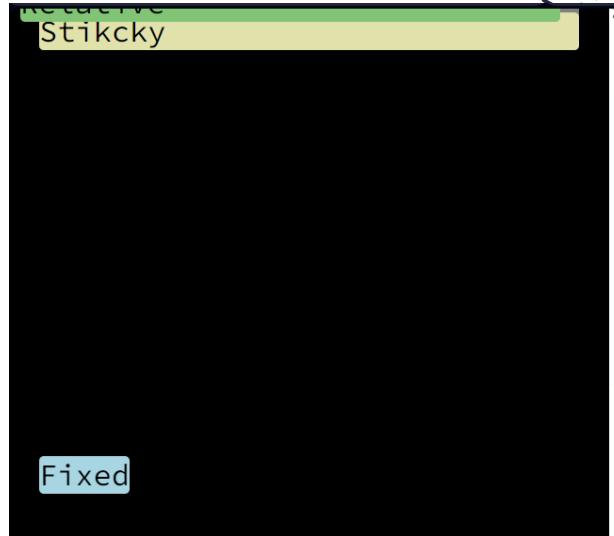
CSS the Skin



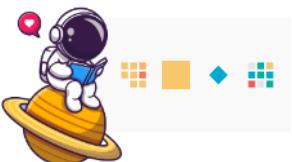
# Position fixed

```
body {
 height: 2600px;
}

.fixed {
 position: fixed;
 top: 50%;
}
```



CSS the Skin





# Position sticky

- El **posicionamiento sticky** se suele utilizar cuando queremos que un elemento se posicione en un lugar específico de forma fija (**«sticky»**, **pegajoso**), como por ejemplo, cuando al hacer scroll llegamos a un elemento y queremos que ese elemento se quede fijo en la parte superior mientras continuamos haciendo scroll. Este comportamiento es muy habitual con los menús superiores de las páginas o las cabeceras de secciones.



css



CSS the Skin

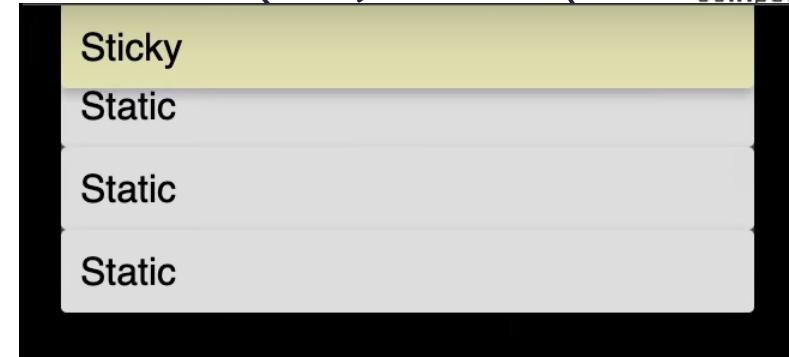




# Position sticky

```
sticky {
 position: sticky;
 top: 0;
}
```

```
<div class="sticky">Sticky</div>
<div class="static">Static</div>
<div class="static">Static</div>
<div class="static">Static</div>
<div class="static">Static</div>
<div class="static">Static</div>
```



Fixed





# Profundidad (niveles)

- Es interesante conocer también la existencia de la propiedad **z-index**, que establece el **nivel de profundidad** en el que está un elemento sobre los demás. De esta forma, podemos hacer que un elemento se coloque encima o debajo de otro.

css



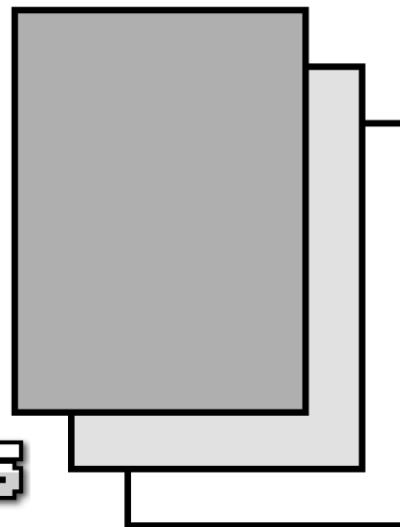
CSS the Skin





# Profundidad (niveles)

**z-index: 15**



**z-index: 5**

**z-index: -5**

CSS



CSS the Skin



# Profundidad (niveles)



```
relative {
 position: relative;
 right: 20px;
 top: 10px;
 bottom: 0;
 /* left: 10px; */
 z-index: 1;
}
```

## CSS Position

Static

Absolute

Relative

Sticky

CSS



CSS the Skin



# Posicionamiento Flotante



- Cuando una caja se posiciona con el modelo de posicionamiento flotante, automáticamente se convierte en una caja flotante, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.

css



CSS the Skin



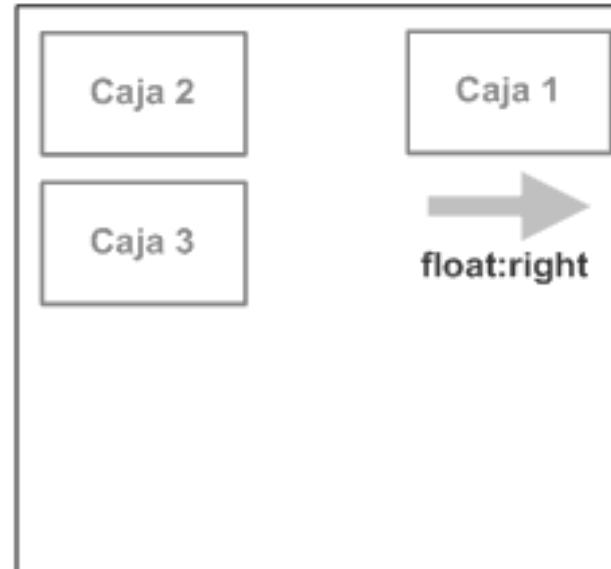
# Posicionamiento Flotante

- Elemento contenedor



Posicionamiento normal

- Elemento contenedor

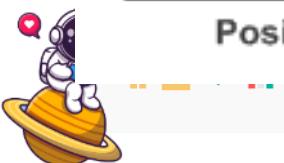


float:right

CSS



CSS the Skin



# Posicionamiento Flotante



```
<body>
 <div class="contenedor-1">Caja 1</div>
 <div class="contenedor-2">Caja 2</div>
 <div class="contenedor-3">Caja 3</div>
</body>
```

```
div {
 border: 1px solid black;
 width: 100px;
 height: 100px;
 margin-bottom: 10px;
}
```

```
.contenedor-1 {
 float: right;
```

CSS



CSS the Skin



# Posicionamiento Flotante

- Caja 1
- Caja 2
- Caja 3



```
div {
 border: 1px solid black;
 width: 100px;
 height: 100px;
 margin: 10px;
}

.contenedor-1 {
 float: left;
}
.contenedor-2 {
 float: left;
}
.contenedor-3 {
 float: left;
}
```



CSS



CSS the Skin



# Posicionamiento Flotante



- Caja 1
- Caja 2
- Caja 3
- Si no existe sitio en la línea actual, la caja flotante baja a la línea inferior hasta que encuentra el sitio necesario



```
div {
 border: 1px solid black;
 width: 40%;
 height: 100px;
 margin : 10px;
}
```

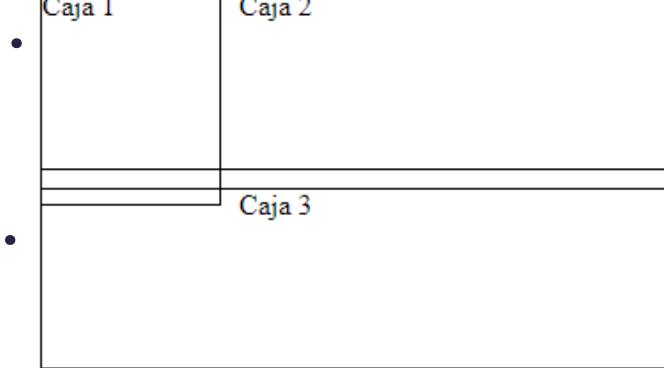
```
.contenedor-1 {
 float: left;
}
.contenedor-2 {
 float: left;
}
.contenedor-3 {
 float: left;
}
```



CSS the Skin



# Posicionamiento Flotante



Los elementos que se encuentran alrededor de una caja flotante adaptan sus contenidos para que fluyan alrededor del elemento posicionado:



```
div {
 border: 1px solid black;
 width: 80%;
 height: 100px;
 margin : 10px;
}

.contenedor-1 {
 width: 100px;
 height: 120px;
 float: left;
 margin-top: 0px;
}

.contenedor-2 {
}

.contenedor-3 {
}
```



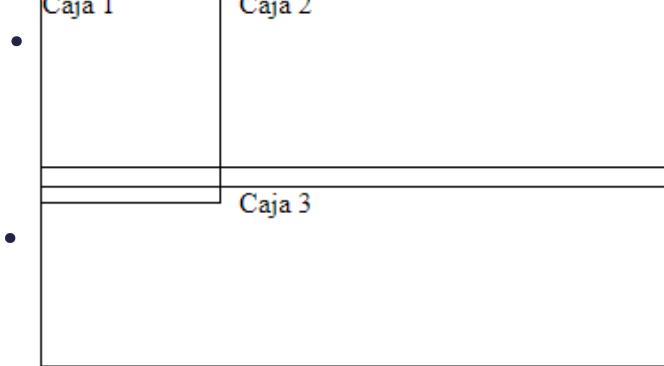
css



CSS the Skinn



# Posicionamiento Flotante



Los elementos que se encuentran alrededor de una caja flotante adaptan sus contenidos para que fluyan alrededor del elemento posicionado:



```
div {
 border: 1px solid black;
 width: 80%;
 height: 100px;
 margin : 10px;
}

.contenedor-1 {
 width: 100px;
 height: 120px;
 float: left;
 margin-top: 0px;
}

.contenedor-2 {
}

.contenedor-3 {
}
```



css



CSS the Skinn





# Programa académico **CAMPUS**

