

#10

CSS y sus avances

“En el capítulo anterior...”

- Repasamos HTML y sus etiquetas más importantes
- Revisen el repositorio **clase-09** para más detalle
- Repasaremos CSS *en orden cronológico*
 - “Conocimiento incremental”

Introducción

CSS1

- Publicado en **1996**, “obsoleto” desde **2018**
 - `display: block, inline...`
 - **float**
 - Pseudo elementos (`:hover`, `:visited...`)
 - font, color, margin, padding, border
 - Unidades (cm, mm, px, em...)
 - ...

CSS2

- Publicado en **1998**, en uso
 - **position**: relative, absolute...
 - Selectores potentes (*, >, +...)
 - Capas (z-index)
 - @media

CSS3

- Publicado en **2012**, en uso
 - Agrega (o cambia) funcionalidades a CSS**2.3**
- Sigue en desarrollo

CSS3 – En progreso

Recomendado usar

Sujetas a cambios
y revisiones



2. Cascading Style Sheets (CSS) — The Official Definition

As of 2014, Cascading Style Sheets (CSS) is defined by the following specifications.

- [CSS Level 3, latest revision \(including errata\)](#) [C324]
- This defines the core of CSS, parts of which are overridden by later specifications. We recommend in particular reading [Chapter 3](#), which introduces some of the basic concepts of CSS and its design principles.
- [CSS Syntax Level 3](#) [C325-SYNTAX-3]
- Replaces [CSS211](#), [CSS212](#), [CSS213](#), [CSS214](#), and [CSS215](#), defining how CSS is parsed.
- [CSS Style Attribute](#) [C326-STYLE-ATTR]
- Defines how CSS declarations can be embedded in markup attributes.
- [Media Queries Level 3](#) [C327-MEDIA-QUERIES]
- Replaces [CSS217](#) and expands on the syntax for media-specific styles.
- [CSS Conditional Rules Level 3](#) [C328-CONDITIONALS]
- Replaces [CSS218](#), updating the definition of [@media](#) rules to allow nesting, and introduces [@supports](#) rules for feature-support queries.
- [CSS Namespaces](#) [C329-NAMESPACES]
- Introduces an [@namespace](#) rule to allow namespace-qualified selectors.
- [Selector Level 3](#) [C330-SELECTOR-3]
- Replaces [CSS219](#) and [CSS220](#), defining an extended range of selectors.
- [CSS Cascading and Inheritance Level 3](#) [C331-CASCADE-3]
- Replaces [CSS21](#), [CSS22](#), and [CSS23](#).
- [CSS Values and Units Level 3](#) [C332-VALUES-3]
- Replaces [CSS224](#), [CSS225](#), and [CSS226](#), defining CSS's property definition syntax and expanding its core units.
- [CSS Color Level 3](#) [C333-COLOR]
- Replaces [CSS227](#), [CSS228](#), and [CSS229](#), defining an extended range of color values. Also introduces the [opacity](#) property.
- [CSS Backgrounds and Borders Level 3](#) [C334-BACKGROUND]
- Replaces [CSS230](#) and [CSS231](#), providing more control of backgrounds and borders, including layered background images, image borders, and drop shadows.
- [CSS Image Values and Replaced Content Level 3](#) [C335-IMAGE]
- Provides a new foundation for the sizing of replaced elements (such as images), adds additional controls to their sizing and orientation, and introduces syntax for gradients as images in CSS.
- [CSS Fonts Level 3](#) [C336-FONTS-3]
- Replaces [CSS232](#) and provides more control over font choice and feature selection.
- [CSS Multi-column Layout Level 1](#) [C337-MULTICOL-1]
- Introduces multi-column flow to CSS layout.
- [CSS User Interface Module Level 3](#) [C338-UI-3]
- Replaces [CSS233](#) and [CSS234](#), defining [cursor](#), [outline](#), and several new CSS features that also enhance the user interface.
- [CSS Compositing and Blending Level 1](#) [COMPOSITING]
- Defines the compositing and blending of overlaid content and introduces features to control their modes.
- [CSS Writing Mode Level 3](#) [C339-WRITING-MODES-3]
- Defines CSS support for various international writing modes, such as left-to-right (e.g. Latin or Indic), right-to-left (e.g. Hebrew or Arabic), bidirectional (e.g. mixed Latin and Arabic) and vertical (e.g. Asian scripts).
- [CSS Flexible Box Module Level 1](#) [C340-FLEXBOX-1]
- Introduces a flexible linear layout model for CSS.
- [CSS Custom Properties for Cascading Variables Module Level 1](#) [C341-CUSTOM-PROPS-1]
- Introduces cascading variables as a new primitive value type fully accepted by all CSS properties, and custom properties for defining them.
- [CSS Transform Level 1](#) [C342-TRANSFORMS]
- Introduces graphical transformations to CSS.

The following modules are widely deployed with [rough interoperability](#), but the details are not fully worked out and they need more testing and bugfixing.

- [CSS Transitions Level 1](#) [C343-TRANSITIONS] and [CSS Animations Level 1](#) [C344-ANIMATIONS], together with [C345-TRANSITION]
- Define mechanisms for transitioning the computed values of CSS properties over time.
- [CSS Grid Layout Module Level 1](#) [C346-GRID-1]
- Defines a two-dimensional grid-based layout system, optimized for user interface design. In the grid layout model, the children of a grid container can be positioned into arbitrary slots in a predefined flexible or fixed-size layout grid.
- [CSS Text Module Level 3](#) [C347-TEXT-3]
- Defines properties for text manipulation and specifies their processing model. It covers line breaking, justification and alignment, white space handling, and text transformation.
- [CSS Will Change Level 1](#) [C348-WILL-CHANGE-1]
- Introduces a performance hint property called [will-change](#).
- [Filter Effects Module Level 1](#) [FILTER-EFFECTS-1]
- [Filter Effects](#) are a way of processing an element's rendering before it is displayed in the document.

The following modules have completed design work, and are fairly stable, but have not received much testing and implementation experience yet.

- [CSS Counter Style Level 3](#) [C349-COUNTER-STYLE-3]
- Expands the possible values of [counter-style](#) and provides an [@counter-style](#) syntax for customized counter styles.
- [CSS Masking Level 1](#) [C350-MASKING-1]
- Replaces [CSS235](#) 1.1.2 and introduces more powerful ways of clipping and masking content.
- [CSS Shapes Module Level 1](#) [C351-SHAPES-1]
- Extends from [@media](#) to allow non-rectangular clipping shapes.
- [CSS Text Decoration Level 3](#) [C352-TEXT-DECOR-3]
- Replaces [CSS236](#) 1.2, providing more control over text decoration lines and adding the ability to specify text-emphasis marks and text shadows.
- [CSS Speech Module Level 1](#) [C353-SPEECH-1]
- Replaces [CSS237](#), or introducing the (non-normative) speech rendering chapter.
- [CSS Box Alignment Module Level 1](#) [C354-ALIGN-1]
- Combines the features of CSS relating to the alignment of boxes within their containers. In the various CSS box layout models: block layout, table layout, flex layout, and grid layout.
- [CSS Fragmentation Module Level 3](#) [C355-FRAGMENT-3]
- Describes the fragmentation model that partitions a flow into pages, columns, or regions.
- [CSS Containment Module Level 1](#) [C356-CONTAIN-1]
- Describes the [contain](#) property, which indicates that the element's content is independent of the rest of the page. This enables heavy optimizations by user agents when used well.
- [CSS Scroll Snap Module Level 1](#) [C357-SCROLL-SNAP-1]
- Defines features to support snapping and scrolling behavior with "snap positions".
- [Media Queries Level 4 \(MEDIA-QUERIES-4\)](#)
- Replaces and expands [CSS3-MEDIA-QUERIES](#), expanding the syntax, deprecating most media types, and introducing new media features.
- [CSS Cascading and Inheritance Level 4](#) [C358-CASCADE-4]
- Replaces and expands [CSS3-CASCADE](#), describing how to cascade style rules and assign values to all properties on all elements. It may deprecating and inheritance, values are prepended for all properties on all elements.

We hope to incorporate them into a future snapshot.

CSS4

- No hay una publicación asociada
- Funciona igual a CSS**3**, agrega “módulos”, modificando funcionalidades existentes

CSS1

CSS1 – Unidades de medida

- Absolutas
 - No dependen del tamaño de pantalla
- Relativas
 - Sujetas al tamaño de pantalla
- Recomendado usar **relativas**

CSS1 – Unidades absolutas

cm	centímetros
mm	milímetros
in	pulgadas (1in = 96px = 2.54cm)
px	pixeles (1px = 1/96 de 1in)
pt	puntos (1pt = 1/72 de 1in)
pc	picas (1pc = 12 pt)

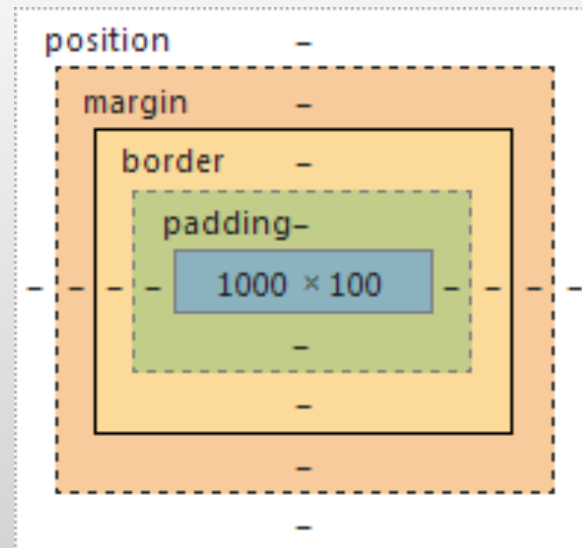
CSS1 – Unidades relativas

%	Porcentaje relativo al elemento padre
em / rem	Relativo al font-size del elemento. El rem es relativo al elemento raíz.
vw	Relativo al 1% del ancho del viewport
vh	Relativo al 1% del alto del viewport
vmin	Relativo el 1% de la dimensión menor del viewport
vmax	Relativo al 1% de la dimensión mayor del viewport

CSS1 - float

- Recordemos:
 - Todo en HTML son “box” (cajas)
 - Todo en CSS son “box-model” (cajas con estilos)

- Esto es un **box-model** →



CSS1 - float



CSS1 - float

- Alinea un elemento (izquierda, o derecha) de su contenedor (padre)
- Sus elementos “hermanos” “fluyen” alrededor del elemento flotante
- Literalmente, “es como si flotara”

CSS1 - float

Lorem ipsum dolor sit amet consectetur adipisicing elit. Exercitationem explicabo rem, tempora doloremque fugit corrupti voluptate illum! Blanditiis veritatis dolore ea impedit magni corporis nesciunt illo, quaerat alias explicabo. Obcaecati.

Similique ullam perferendis dolorum, laboriosam cumque dolores sint voluptates facilis impedit sit adipisci, quaerat veritatis inventore officiis porro, vel nobis suscipit quo repudiandae modi quidem rerum aliquam! Sequi, iusto aspernatur!

Quo sequi praesentium provident voluptas quas deserunt illum laborum quia, nisi voluptatem dolore molestiae, repellendus itaque, tenetur culpa. Vel animi officia molestias exercitationem soluta corporis iure, totam dolorem! Excepturi, nemo?

Cupiditate, ea, explicabo sunt dignissimos consectetur nostrum atque iusto, nihil earum laudantium amet autem! Maiores, mollitia ducimus earum nulla consequatur sapiente! Facere delectus repellat tempore blanditiis impedit perferendis ex velit.

Blanditiis nobis animi neque amet repellendus vel alias modi obcaecati atque eaque voluptas nulla, esse rerum tenetur. Expedita ullam vero similique veritatis quasi labore repellat eligendi, sunt officia, sed autem.

Ad harum iusto, consectetur sit quaerat unde sequi maxime! Ipsam voluptatem earum asperiores quisquam, blanditiis saepe facilis similique voluptate explicabo, incidunt amet commodi recusandae ullam soluta tempora quidem est! Impedit.

Veniam praesentium a, amet quod omnis sit. Ad saepe tempora odio vel quod nulla velit ipsam, repellat veritatis suscipit quos reiciendis tenetur perferendis quisquam debitis? Error veritatis et aspernatur non.

Molestiae minima perspiciatis sit. Necessitatibus et ducimus velit sint nesciunt nisi modi repudiandae harum amet. Fugit nihil ut velit rem quasi debitis non dolorem officia accusantium! Consectetur accusamus exercitationem aliquam.

Incidunt sit harum, dolore dolorem iure ea consequatur, doloremque voluptate libero, quis tenetur labore? Excepturi nesciunt facilis dicta provident eaque in aspernatur! Error iure tempore alias aspernatur rerum, earum vel.

Doloremque molestias, illum nisi cupiditate quasi dolore! Eum, nisi consequuntur voluptates distinctio, dolorem consectetur excepturi expedita pariat qui cupiditate perspiciatis doloribus obcaecati sunt et nam facere repellat rerum! Sequi, veniam.

Doloremque alias culpa cumque nisi optio reiciendis esse minima tenetur natus! Ratione possimus voluptatum aliquid, suscipit excepturi ex laboriosam, quae accusamus magni fugiat totam impedit. Aliquam, exercitationem. Mollitia, iure earum.

Delectus, enim iste sequi quos, officiis cupiditate adipisci molestias numquam dolor facilis illo earum facere tempore porro iusto unde ullam obcaecati, doloribus recusandae ipsum laboriosam nihil neque ex molestiae. Reiciendis.

Alias porro saepe distinctio, repellat eum dignissimos sint reprehenderit laboriosam libero neque accusamus earum ipsa excepturi cumque animi quis consequuntur incidunt deleniti harum illum. Consectetur amet porro architecto voluptate qui.

Soluta nobis corrupti sed nihil modi sapiente laborum voluptatum. Voluptatibus natus provident eligendi, hic, officia quo velit quam eum, sunt odit obcaecati minus autem distinctio? Vitae rerum similique dicta suscipit!

Nisi laborum ipsum necessitatibus voluptatibus quam. Blanditiis quasi recusandae ad fugit ipsam maiores sed iste, rerum sit modi ipsa voluptatem dolores laudantium eveniet quaerat. Rem ut adipisci tempora quam. Molestias.

Nam ut soluta perferendis, delectus quo tempora perspiciatis qui at illo dolorum eveniet dignissimos expedita repellat maxime, itaque magnam ratione a aliquam? Minima provident, quas eligendi hic corrupti voluptatem dicta.

Eligendi vitae dolor vero inventore facilis eaque, maxime nisi blanditiis voluptates? Illum vero hic nam fuga repellat ullam tenetur, cum neque magnam quas temporibus beatae laborum blanditiis ratione laboriosam saepe.

Doloribus eveniet dolorem soluta reiciendis aspernatur optio sint a aut eum cum impedit veritatis laborum, sed omnis ipsam eligendi voluptate. Repellat eius sit nesciunt voluptatibus quae cumque eligendi laboriosam perspiciatis.

Magni fugiat repellendus architecto ducimus asperiores excepturi rerum quos, obcaecati totam modi. Ab blanditiis, accusantium alias nihil, ipsa consequuntur impedit soluta velit officiis quos, cumque vel veniam necessitatibus ad facilis?

CSS1 - float

```
<body>
<p id="oferta">¡¡Suscribite a nuestras ofertas!! </p>
<p>
Lorem ipsum dolor sit amet consectetur adipisicing elit. Exercitationem explicabo r
```

¡¡Suscribite a nuestras ofertas!!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Exercitationem explicabo r
alias explicabo. Obcaecati.

Similique ullam perferendis dolorum, laboriosam cumque dolores sint voluptates fa
Sed qui iusto consectetur

CSS1 - float

```
<style>
#oferta {
  float: left;
  border: 4px solid red;
}
</style>
```

¡¡Suscribite a nuestras ofertas!!

Lorem ipsum dolor sit amet consectetur ad
corporis nesciunt illo, quaerat alias explicat

Similique ullam perferendis dolorum, labor
repudiandae modi quidem rerum aliquam! Sequi, iusto aspernatur!

CSS1 - clear

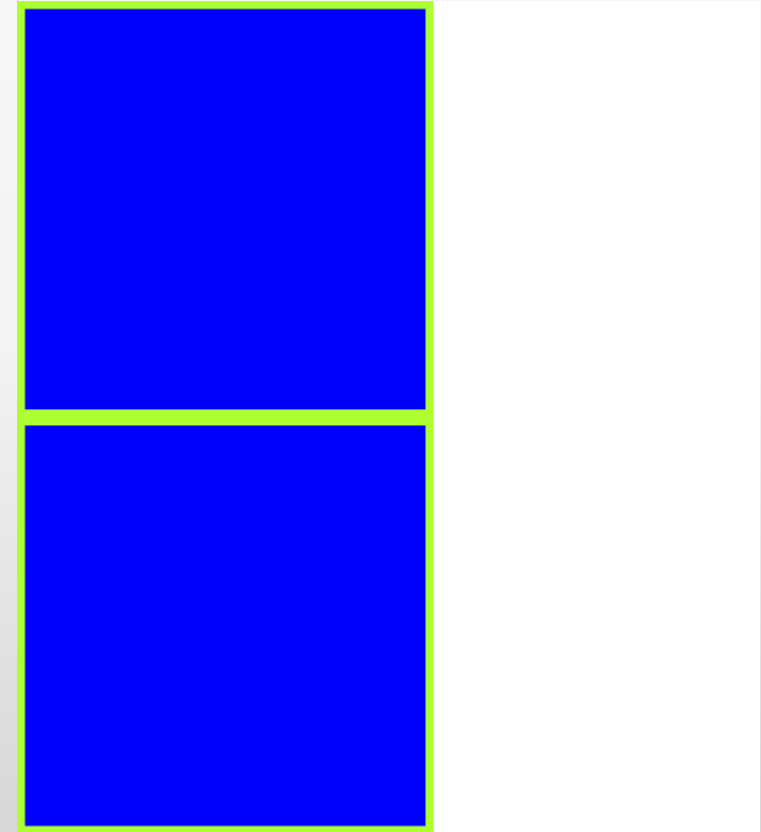
- Va de la mano con **float**
- “No quiero tener elementos flotando a mi alrededor 😞”
 - left
 - right
 - both
 - none

CSS2 – position: absolute

- Por defecto, todo **position** es **static**
- Con **absolute**:
 - Le decimos a la cajita dónde debe pararse
 - Se la saca del “flujo” del documento (*similar* a float)
- `top`, `right`, `bottom`, `left` son los valores a usar

CSS2 – position: absolute

```
... <style>
...   div {
...     background-color: blue;
...     border: 4px solid greenyellow;
...     width: 200px;
...     height: 200px;
...   }
... </style>
... </head>
... <body>
...   <div class="caja1"></div>
...   <div class="caja2"></div>
... </body>
```

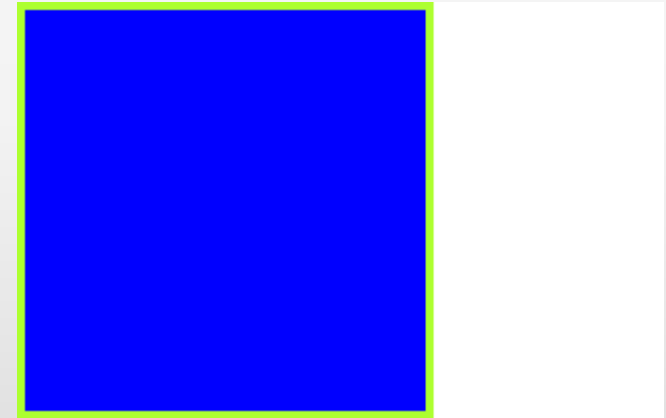


CSS2 – position: absolute

- Por defecto:
 - top: 0
 - right: 0
 - bottom: 0
 - left: 0

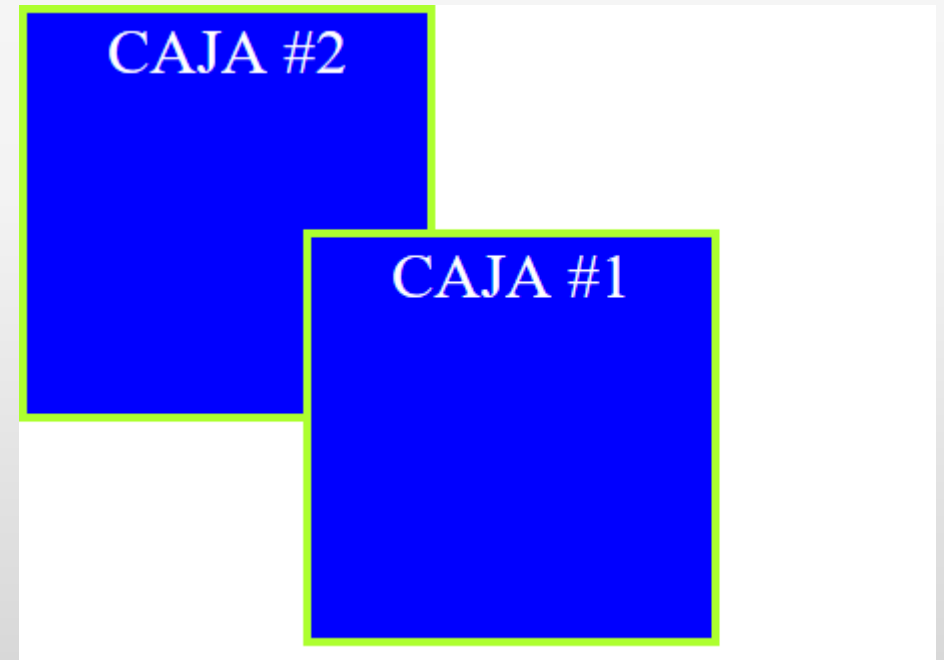
CSS2 – position: absolute

```
...}  
... .caja1 {  
...   position: absolute;  
... }  
</style>
```



CSS2 – position: absolute

```
.caja1 {  
  position: absolute;  
  top: 120px;  
  left: 150px;  
}
```



CSS2 – position: relative

- ¿Diferencia con float?
 - Podemos decirle a la cajita con más precisión dónde pararse
 - Las demás cajitas hermanas **no saben de nuestra cajita**
 - Con float nadie se nos pone encima, en cambio con absolute, si

CSS2 – position: relative

- Por defecto, todo **position** es **static**
- Con **relative**:
 - Le decimos a la cajita dónde debe pararse, según su posición original
 - Se la saca del “flujo” del documento (*similar* a float), pero le dejan un espacio vacío (como si estuviese ahí)
- top, right, bottom, left son los valores a usar

CSS2 – position: relative

```
<body>  
  <div class="caja1">CAJA #1</div>  
  <div class="caja2">CAJA #2</div>  
  <div class="caja3">CAJA #3</div>  
</body>
```

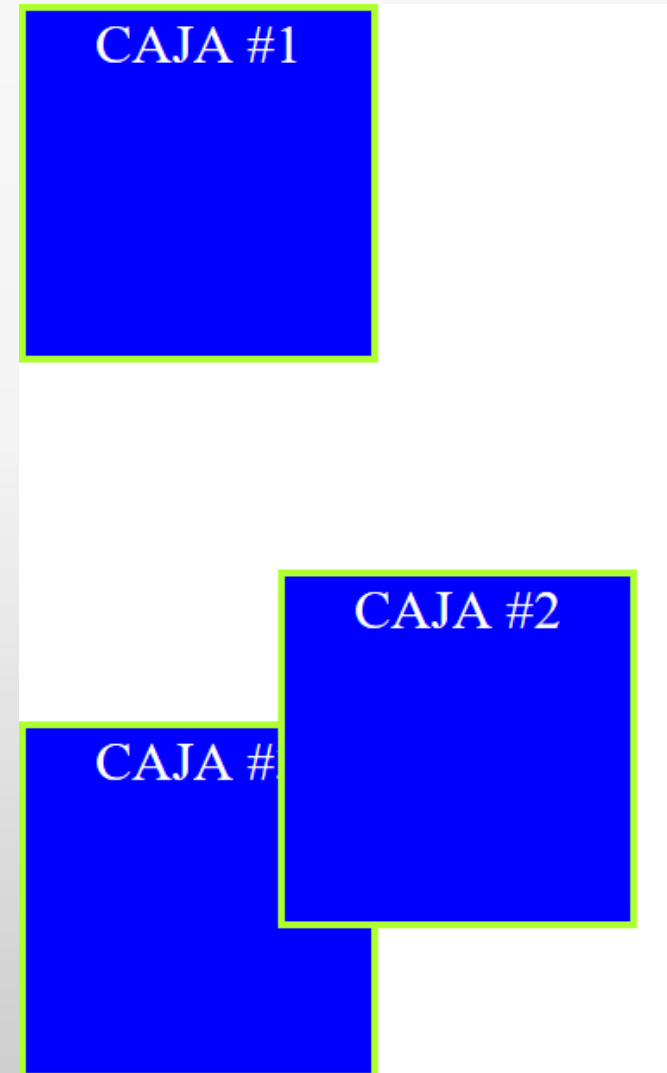
CAJA #1

CAJA #2

CAJA #3

CSS2 – position: relative

```
.caja2 {  
  position: relative;  
  top: 120px;  
  left: 150px;  
}
```



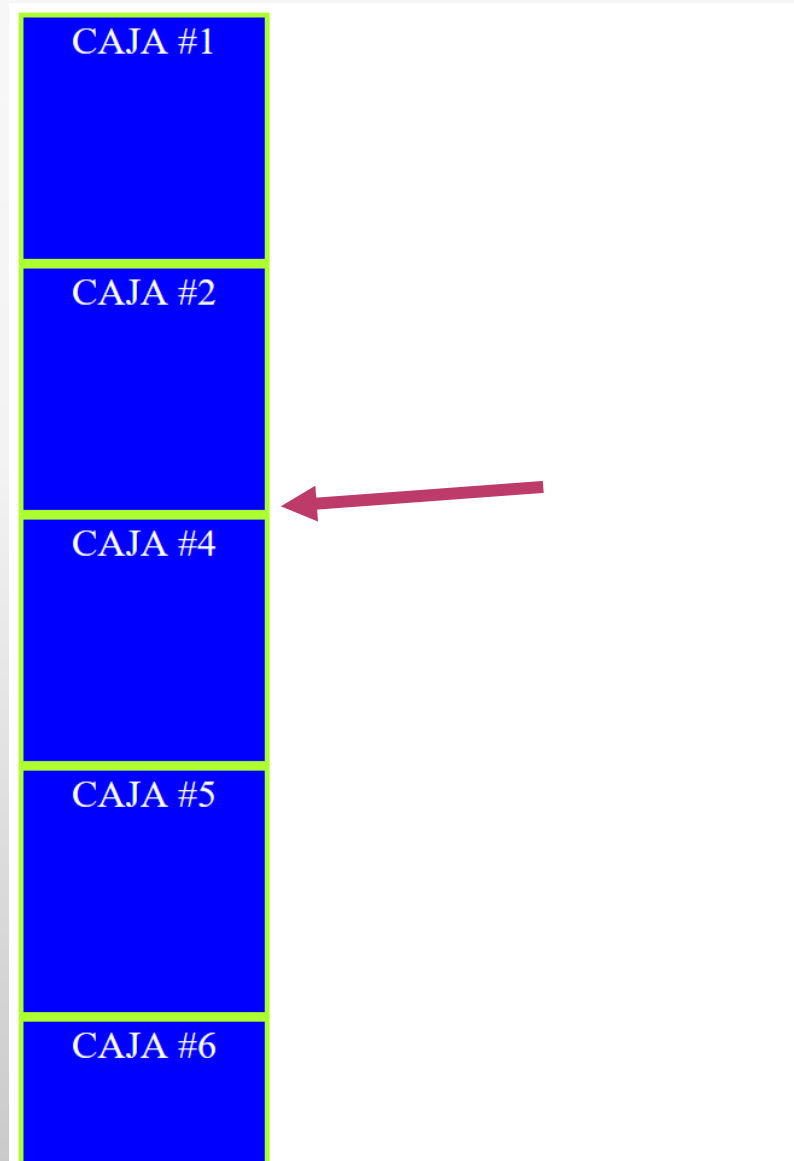
CSS2 – position: fixed

- Por defecto, todo **position** es **static**
- Con **fixed**:
 - Le decimos a la cajita que “se quede en su posición original, pero **que no se mueva de ahí**”
 - Se la saca del “flujo” del documento (*similar* a float), pero le dejan un espacio vacío (como si estuviese ahí)
- top, right, bottom, left son **opcionales**

CSS2 – position: fixed

```
<body>  
  <div class="caja1">CAJA #1</div>  
  <div class="caja2">CAJA #2</div>  
  <div class="caja3">CAJA #3</div>  
  <div class="caja4">CAJA #4</div>  
  <div class="caja5">CAJA #5</div>  
  <div class="caja6">CAJA #6</div>  
  <div class="caja7">CAJA #7</div>  
  <div class="caja8">CAJA #8</div>  
  <div class="caja9">CAJA #9</div>  
  <div class="caja10">CAJA #10</div>  
</body>
```

```
.caja2 {  
  position: fixed;  
}
```



CSS3 – position: sticky

- Por defecto, todo **position** es **static**
- Con **sticky**:
 - Le decimos a la cajita que “se quede en su posición original, pero **que no se mueva de ahí**”
 - A diferencia de `fixed`, no se la saca del flujo, a menos que “desaparezca” de la visión del usuario (`overflow`)
- `top`, `right`, `bottom`, `left` son **obligatorios**

CSS3 – position: sticky

```
.caja2 {  
  position: sticky;  
  /* Si "desaparece" de la pantalla,  
  * le decimos en donde queremos que  
  * quede fija.  
  */  
  top: 0;  
}
```

Pista: y la .caja5?

CAJA #2

CAJA #4

CAJA #5

CAJA #6

CAJA #7

CSS4 – @media avanzado

No funciona en todos lados,
todavía...

```
@media (width ≥ 1200px) {  
  div {  
    color: ■green;  
  }  
}  
  
@media (width ≥ 400px) and (width < 1200px) {  
  div {  
    color: ■yellow;  
  }  
}  
  
@media (width < 400px) {  
  div {  
    color: ■red;  
  }  
}
```



WIKIPEDIA
The Free Encyclopedia

Titulo
Que son
Por que son tan
lindas
Mas informacion

Tortugas de tierra

Que son

Lorem ipsum dolor sit amet consectetur adipisicing elit. Assumenda, doloribus! Repudiandae, nobis tenetur voluptate id sequi officiis unde corrupti alias iure, quibusdam, sapiente natus aperiam cumque consequatur! Quae, maiores accusantium? Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quibusdam natus quasi voluptatem, quod impedit quas amet quia facilis eligendi architecto possimus enim facere dolorum vero veritatis. Ex quisquam dolor eveniet.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Assumenda, doloribus! Repudiandae, nobis tenetur voluptate id sequi officiis unde corrupti alias iure, quibusdam, sapiente natus aperiam cumque consequatur! Quae, maiores accusantium?

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fuga natus voluptas ducimus repellat enim? Qui, eos est cumque obcaecati tempora, nisi blanditiis nulla architecto tenetur, repudiandae cum aperiam laboriosam doloribus?

Distinctio dolores dolore expedita dicta facilis molestias aliquid reiciendis nostrum ex at dolor rem, assumenda debitis ipsum eius fugiat enim nobis doloremque saepe voluptates atque libero accusamus sequi. Sed, laudantium.

Illo expedita quis temporibus est alias praesentium at fugiat sed odio qui vel eligendi neque velit, quae nam soluta similique doloribus accusamus amet aut sunt porro ipsum mollitia! lusto, amet.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Assumenda, doloribus! Repudiandae, nobis tenetur voluptate id sequi officiis unde corrupti alias iure, quibusdam, sapiente natus aperiam cumque consequatur! Quae, maiores

Tortugas de tierra



Tipo	Tierra
Tipo	Tierra
Tipo	Tierra