

# Ingeniería en Informática / Ingeniería en Sistemas de Información / Ingeniería de Software

## Programación III



Framework



Bootstrap

# ¿qué es bootstrap?

- Bootstrap es un framework CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.
- Su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles.
- Bootstrap incluye plantillas de diseño basadas en HTML y CSS para tipografía, formularios, botones, tablas, navegación, modales, carruseles de imágenes y muchos otros, así como complementos de JavaScript opcionales.



# ¿Cómo funciona?

- Bootstrap está conformado por una serie de archivos CSS y JavaScript responsables de asignar características específicas a los elementos de la página.
- Básicamente, la estructura del framework se compone de dos carpetas:

# Características de Bootstrap

- Diseño responsivo
- Componentes
- Utilidades
- Permite un desarrollo web más rápido y sencillo



# ¿Por qué usar bootstrap?

- Mobile-first
- Estándar visual
- Reutilización de código
- Comunidad activa



# compatibilidad

- Bootstrap admite las últimas versiones estables de todos los principales navegadores y plataformas.
- Los navegadores alternativos que utilizan la última versión de WebKit, Blink o Gecko, ya sea directamente o mediante la API de vista web de la plataforma, no son compatibles explícitamente.
- Sin embargo, Bootstrap debería mostrarse y funcionar correctamente en estos navegadores también.

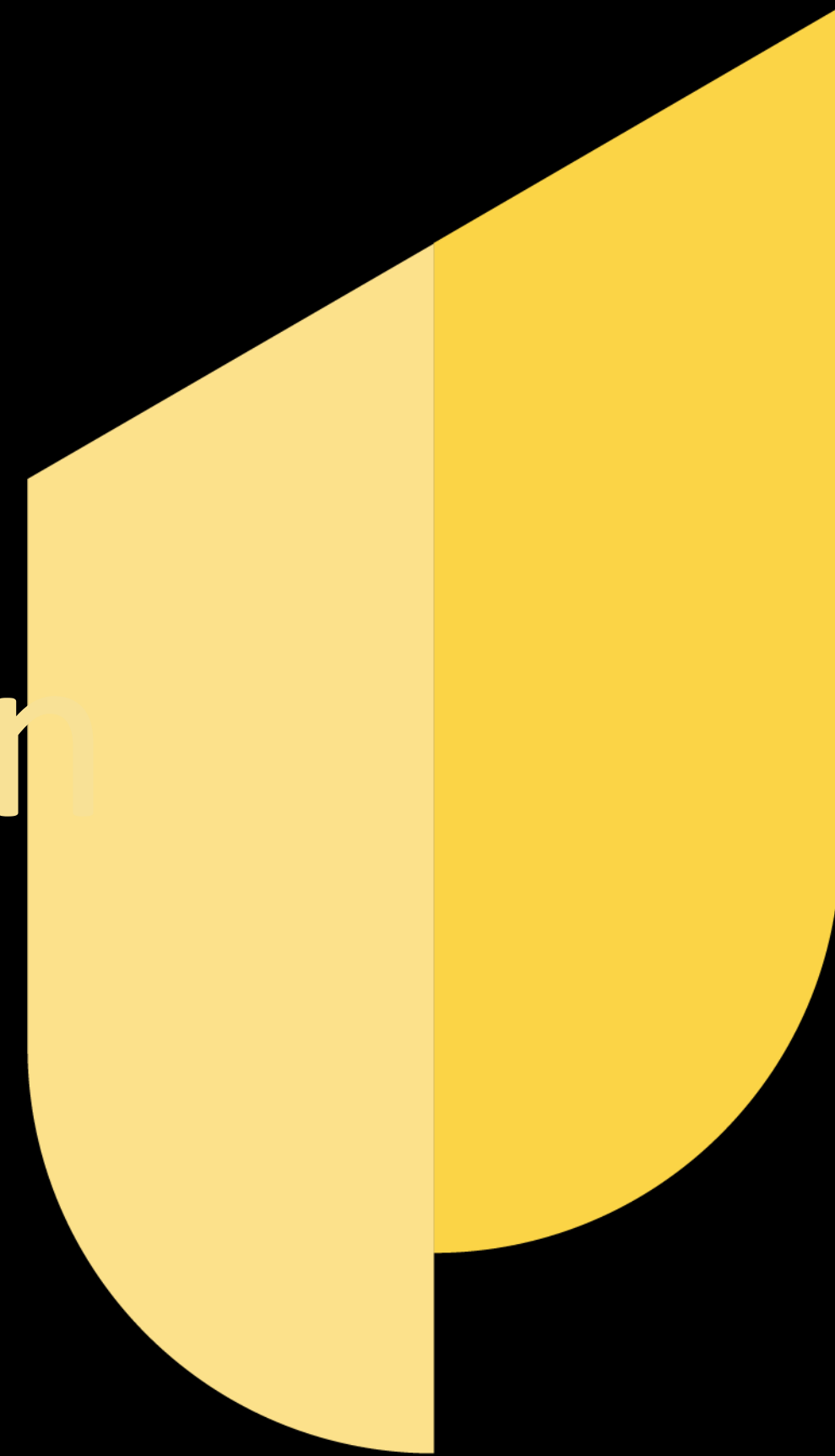


	Chrome	Firefox	Safari	Android Browser & WebView
Android	Supported	Supported	—	v6.0+
iOS	Supported	Supported	Supported	—

	Chrome	Firefox	Microsoft Edge	Opera	Safari
Mac	Supported	Supported	Supported	Supported	Supported
Windows	Supported	Supported	Supported	Supported	—



Instalación



Bootstrap

# Forma #1

- Descargar el código del [framework](#) y colocar los archivos bootstrap.min.css y bootstrap.bundle.js en las carpetas css y js respectivamente.
- Incluir los archivo anteriores en cada uno de los documentos html a utilizar el framework mediante la etiqueta link para el css y script para el js.



# Forma #2

- Omita la descarga de archivos y utilizar jsDelivr para cargar la versión en caché de CSS y JS compilados de Bootstrap a su proyecto.

Bootstrap



layout

# Breakpoints

- Los Breakpoints son anchos personalizables que determinan cómo se comporta el diseño responsivo en todos los tamaños de dispositivos o vistas en Bootstrap.
- Bootstrap incluye seis Breakpoints predeterminados, a veces denominados niveles de cuadrícula, para construir de manera responsiva.



Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

- Cada punto de interrupción se eligió para contener cómodamente contenedores cuyos anchos son múltiplos de 12.
- Los puntos de interrupción también son representativos de un subconjunto de tamaños de dispositivo y dimensiones de ventana gráfica comunes; no se orientan específicamente a cada caso de uso o dispositivo.
- En cambio, los rangos brindan una base sólida y consistente sobre la cual construir para casi cualquier dispositivo.



Bootstrap



Contenedores



# Containers

- Los contenedores son un bloque de construcción fundamental de Bootstrap que contienen, rellenan y alinean su contenido dentro de un dispositivo o ventana gráfica determinada.
- Hay dos tipos de contenedores en bootstrap:

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
max-width	100%	540px	720px	960px	1140px

# Como funcionan

- Los contenedores son el elemento de diseño más básico en Bootstrap y son necesarios cuando se usa el sistema de cuadrícula predeterminado.
- Los contenedores se utilizan para contener, rellenar y (a veces) centrar el contenido dentro de ellos.
- Si bien los contenedores se pueden anidar, la mayoría de los diseños no requieren un contenedor anidado.
- Los contenedores centran y rellenan horizontalmente su contenido.



	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX- Large ≥1400px
.container	100%	540px	720px	960px	1140px	1320px
.container-sm	100%	540px	720px	960px	1140px	1320px
.container-md	100%	100%	720px	960px	1140px	1320px
.container-lg	100%	100%	100%	960px	1140px	1320px
.container-xl	100%	100%	100%	100%	1140px	1320px
.container-xxl	100%	100%	100%	100%	100%	1320px
.container-fluid	100%	100%	100%	100%	100%	100%

Bootstrap



grid system

- Bootstrap posee una poderosa cuadrícula de flexbox para dispositivos móviles para crear diseños de todas las formas y tamaños gracias a un sistema de doce columnas, seis niveles de respuesta predeterminados, variables Sass y mixins, y docenas de clases predefinidas.
- El sistema de cuadrícula de Bootstrap utiliza una serie de contenedores, filas y columnas para diseñar y alinear el contenido. Está construido con flexbox y es totalmente receptivo.



```
<div class="container">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
</div>
```



Column	Column	Column
--------	--------	--------



- La cuadrícula admite seis puntos de interrupción receptivos.
- Los puntos de interrupción se basan en media queries de ancho mínimo, lo que significa que afectan ese punto de interrupción y todos los que están por encima de él (por ejemplo, `.col-sm-4` se aplica a `sm`, `md`, `lg`, `xl` y `xxl`).
- Permitiendo controlar el tamaño y el comportamiento del contenedor y de la columna por cada punto de interrupción.
- Las filas son envoltorios de columnas. Cada columna tiene un padding horizontal (llamado gutter) para controlar el espacio entre ellas.





- Hay 12 columnas de plantilla disponibles por fila, lo que le permite crear diferentes combinaciones de elementos que abarcan cualquier número de columnas.
- Las clases de columna indican el número de columnas de plantilla que se van a abarcar (p. Ej., col-4 abarca cuatro)



# Opciones de cuadrícula

- El sistema de cuadrícula de Bootstrap puede adaptarse a los seis puntos de interrupción predeterminados y a cualquier punto de interrupción que personalice.
- Los seis niveles de cuadrícula predeterminados son los siguientes:

• Extra large

• Extra extra





	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-
# of columns	12					
Gutter width	1.5rem (.75rem on left and right)					



# ejemplo

```
<div class="container">  
  <div class="row">  
    <div class="col">col</div>  
    <div class="col">col</div>  
    <div class="col">col</div>  
    <div class="col">col</div>  
  </div>  
  <div class="row">  
    <div class="col-8">col-8</div>  
    <div class="col-4">col-4</div>  
  </div>  
</div>
```



col	col	col	col
col-8			col-4





- Puede usarse una combinación de diferentes clases para

- 

.col-md-8

.col-6 .col-md-4

.col-6 .col-md-4

.col-6 .col-md-4

.col-6 .col-md-4

.col-6

.col-6



```
<div class="container">
```

```
<!-- Stack the columns on mobile by making one full-width and the other half
```

```
<div class="row">
```

```
  <div class="col-md-8">.col-md-8</div>
```

```
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
```

```
</div>
```

```
<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop
```

```
<div class="row">
```

```
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
```

```
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
```

```
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>
```

```
</div>
```

```
<!-- Columns are always 50% wide, on mobile and desktop -->
```

```
<div class="row">
```

```
  <div class="col-6">.col-6</div>
```

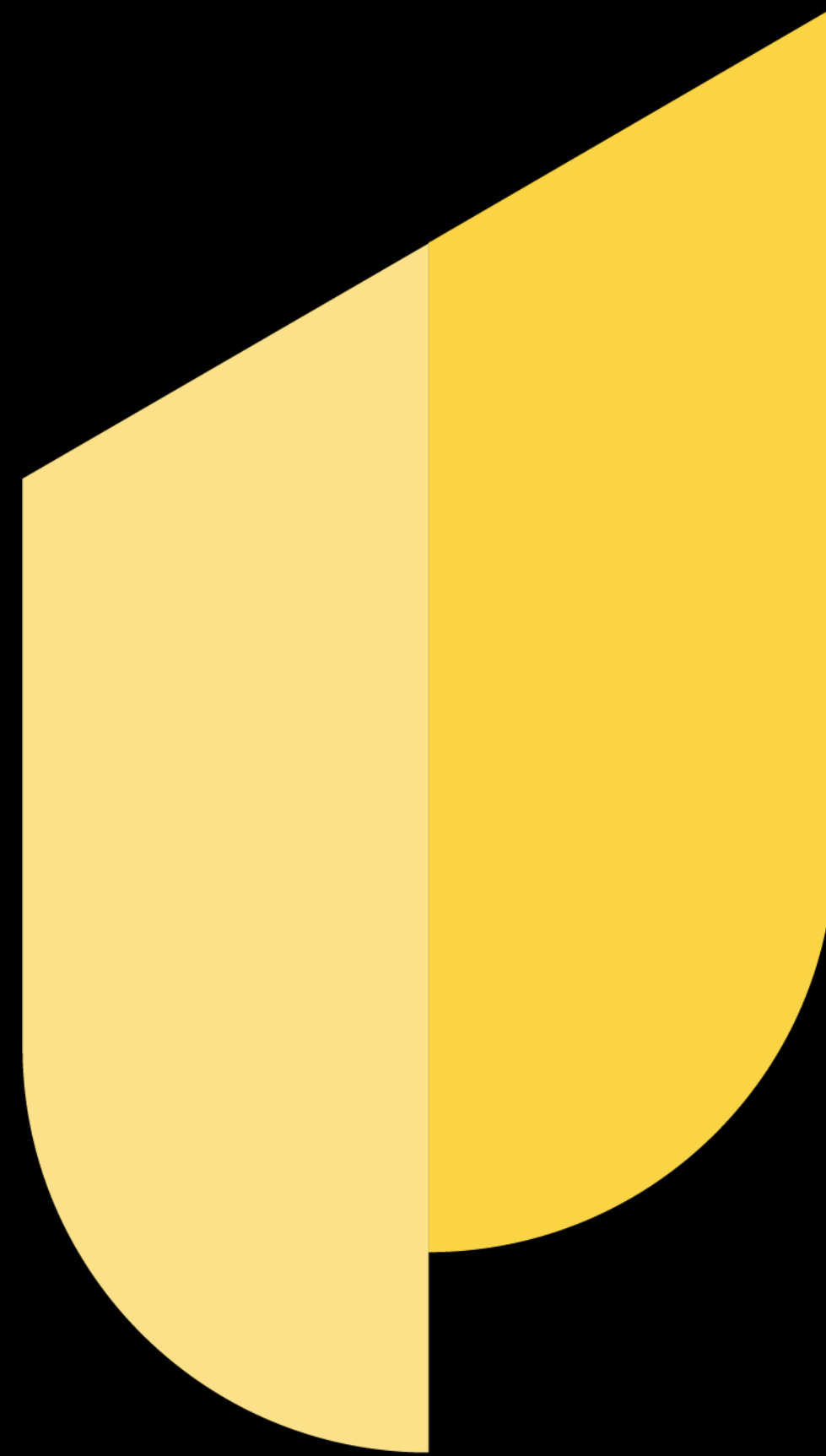
```
  <div class="col-6">.col-6</div>
```

```
</div>
```

```
</div>
```



CSS





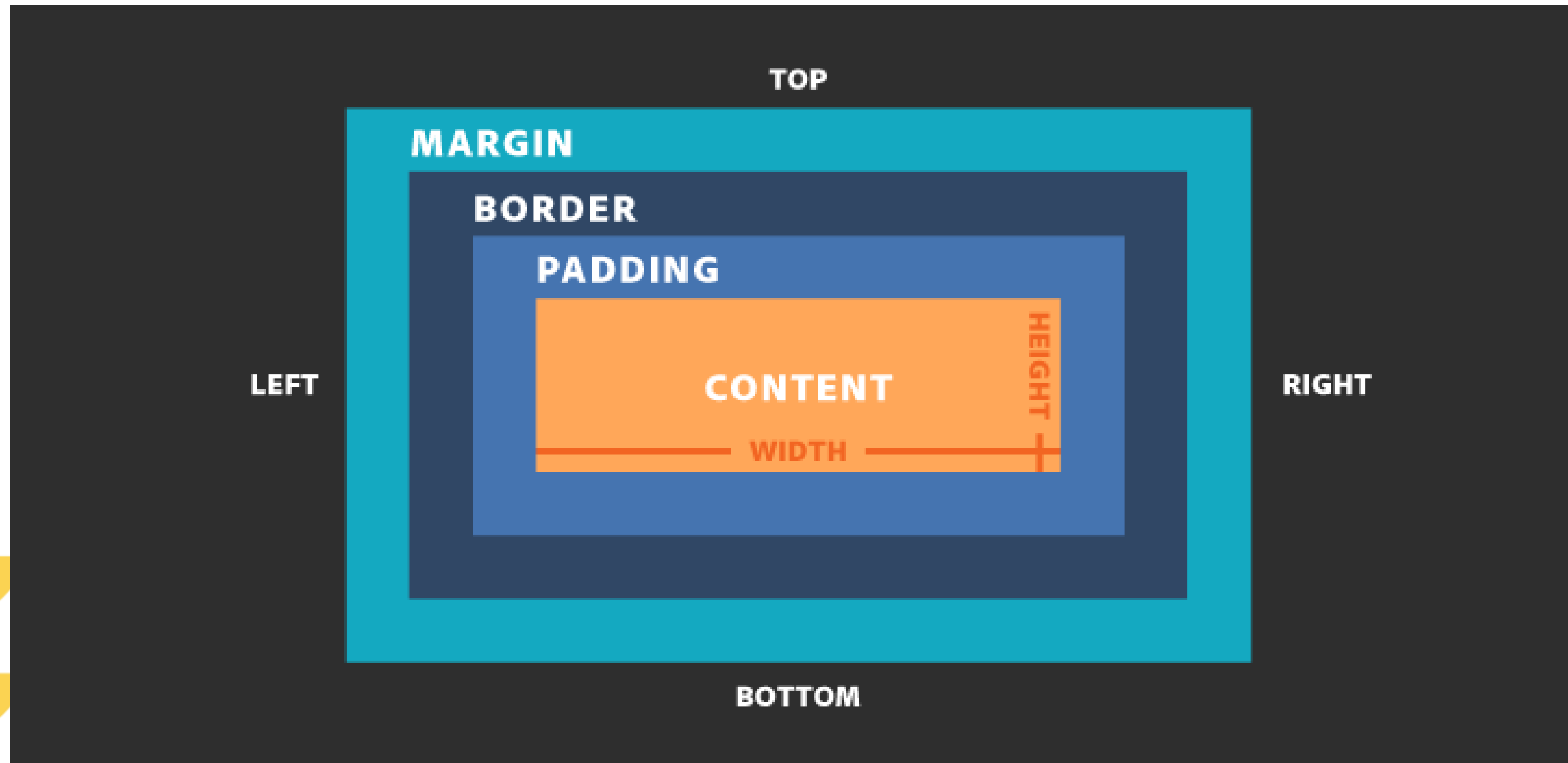
# ¿Qué es CSS?

- Lenguaje que se utiliza para diseñar una página web.
- CSS describe cómo se mostrarán los elementos HTML en el navegador, el papel o en otros medios.
- Permite controlar el diseño de varias páginas web a la vez, ahorrando tiempo de trabajo.
- Las hojas de estilo externas se almacenan en archivos CSS
- CSS trabaja con HTML, pero no es HTML

# Css box model (modelo de caja)

- En CSS, el término "modelo de caja" se utiliza cuando se habla de diseño y maquetación.
- Todos los elementos HTML se pueden considerar como cuadros.
- El modelo de caja CSS es esencialmente una caja que envuelve cada elemento HTML.
- Consiste en: márgenes, bordes, relleno y el contenido real.

# Representación box model



- Contenido (content): el contenido de la caja, donde aparecen el texto y las imágenes.
- Relleno (padding): área alrededor del contenido. El relleno es transparente
- Borde (border): un borde que rodea el relleno y el contenido.
- Margen (margin): área fuera del borde. El margen es transparente
- El modelo de caja nos permite agregar un borde alrededor de los elementos y definir el espacio entre los elementos.

CSS



Colores

# Nombre

- Todos los navegadores modernos admiten los siguientes 140 nombres de color
- <http://www.colors.commutercreative.com/grid/>



# Colores rgb

- En CSS, un color se puede especificar como un valor RGB, usando esta fórmula:
- `rgb(rojo, verde, azul)`
- Cada parámetro (rojo, verde y azul) define la intensidad del color entre 0 y 255.
- Por ejemplo, `rgb (255, 0, 0)` se muestra en rojo, porque el rojo se establece en su valor más alto (255) y los demás se establecen en 0.
- Para mostrar el negro, establezca todos los parámetros de color en 0, así: `rgb (0, 0, 0)`.
- Para mostrar el blanco, establezca todos los parámetros de color en 255, así: `rgb (255, 255, 255)`.



# Colores rgba

- Los valores de color RGBA son una extensión de los valores de color RGB con un canal alfa, que especifica la opacidad de un color.
- Un valor de color RGBA se especifica con:
- rgba (rojo, verde, azul, alfa)





# Colores hexadecimales

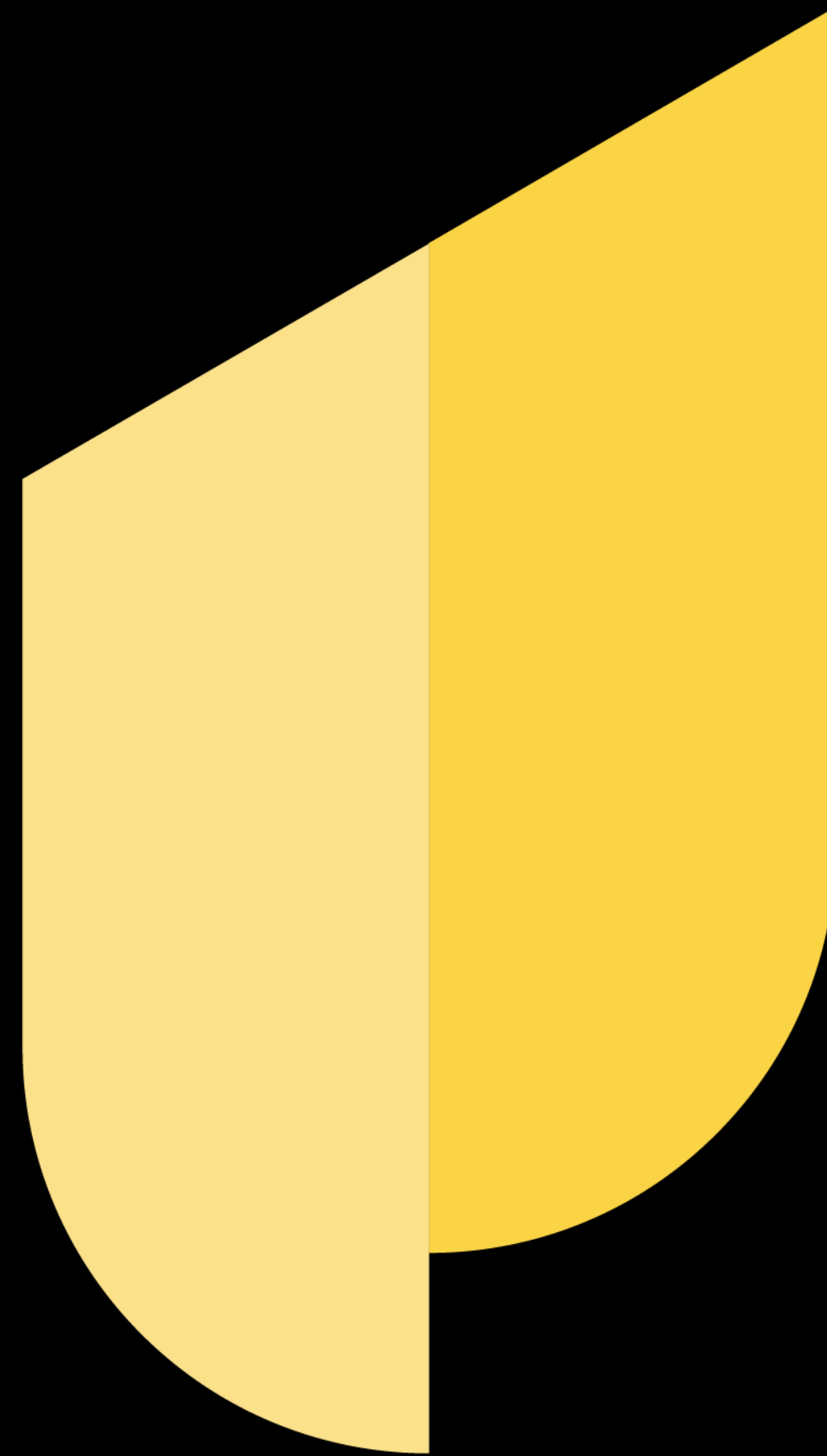
- En CSS, un color se puede especificar usando un valor hexadecimal en la forma:
- #rrggbb



# Colores hsl

- En CSS, un color se puede especificar usando tono, saturación y luminosidad (HSL) en la forma:
- hsl (tono, saturación, luminosidad)
- El tono es un grado en la rueda de colores de 0 a 360. 0 es rojo, 120 es verde y 240 es azul.
- La saturación es un valor porcentual, 0% significa un tono de gris y 100% es el color completo.
- La claridad también es un porcentaje, el 0% es negro, el 50% no es ni claro ni oscuro, el 100% es blanco

CSS



unidades de  
medida

- CSS tiene varias unidades diferentes para expresar una longitud.
- Muchas propiedades de CSS toman valores de "longitud", como ancho, margen, relleno, tamaño de fuente, etc.
- La longitud es un número seguido de una unidad de longitud, como 10px, 2em, etc.
- No pueden aparecer espacios en blanco entre el número y la unidad. Sin embargo, si el valor es 0, la unidad se puede omitir.
- Para algunas propiedades CSS, se permiten longitudes negativas.
- Hay dos tipos de unidades de longitud: absolutas y relativas.

# Medidas absolutas

- Las unidades de longitud absoluta son fijas y una longitud expresada en cualquiera de estas aparecerá exactamente como ese tamaño.
- No se recomienda el uso de unidades de longitud absoluta en la pantalla porque los tamaños de pantalla varían mucho. Sin embargo, se pueden utilizar si se conoce el medio de salida, como para el diseño de impresión.

# Medidas absolutas

- Las unidades de longitud absoluta son fijas y una longitud expresada en cualquiera de estas aparecerá exactamente como ese tamaño.
- No se recomienda el uso de unidades de longitud absoluta en la pantalla porque los tamaños de pantalla varían mucho. Sin embargo, se pueden utilizar si se conoce el medio de salida, como para el diseño de impresión.

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)



# Medidas relativas

- Las unidades de longitud relativa especifican una longitud relativa a otra propiedad de longitud.
- Las unidades de longitud relativa se escalan mejor entre diferentes medios de renderizado.





Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to the width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent element



# Nota

- Las unidades em y rem son prácticas para crear diseños escalables.
- \* Viewport = el tamaño de la ventana del navegador. Si la ventana tiene 50 cm de ancho,  $1\text{ vw} = 0,5\text{ cm}$ .

Incluyendo



CSS a un  
documento

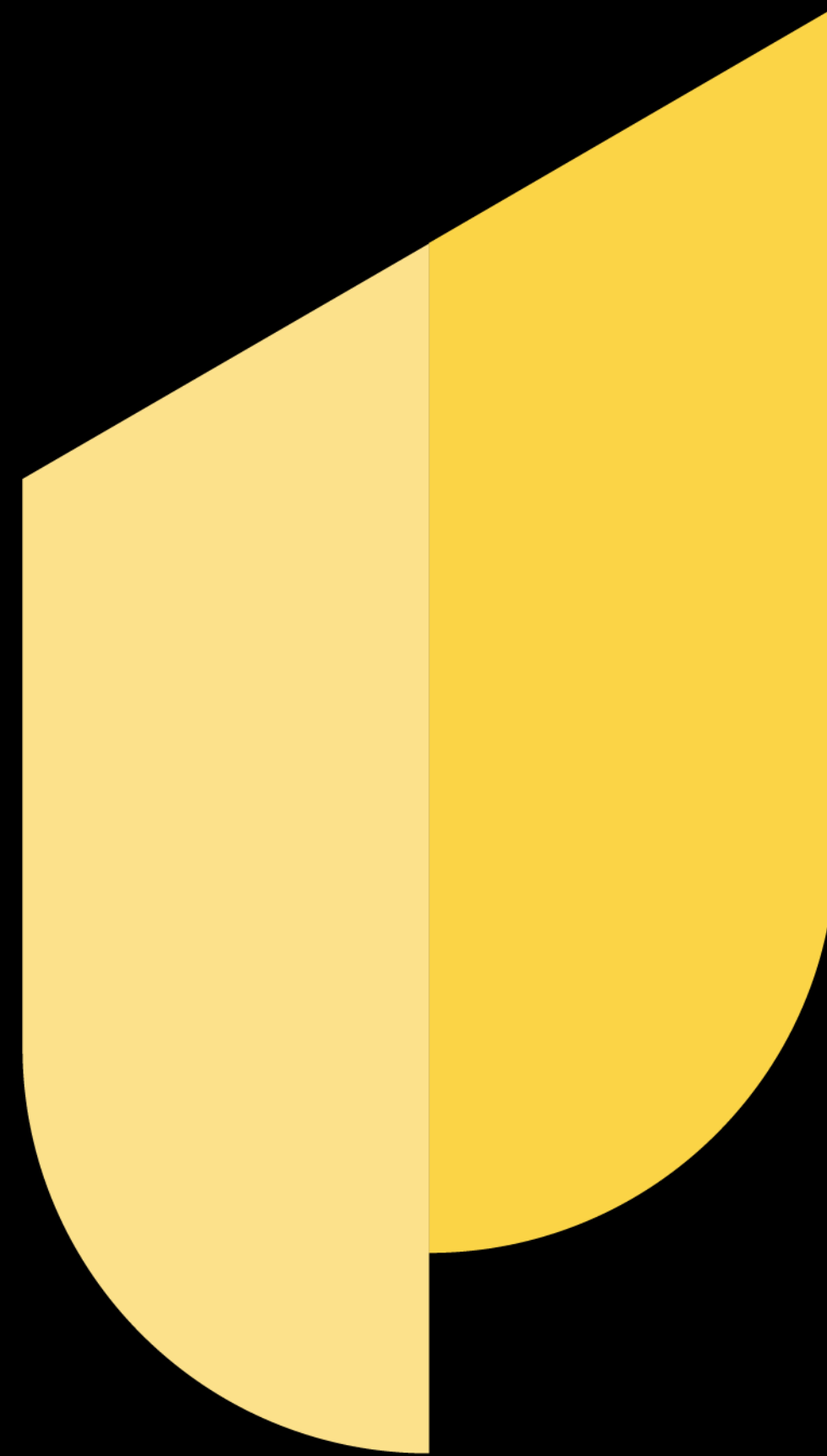
- Hay tres formas de insertar una hoja de estilo:
- CSS externo
- CSS interno
- CSS en línea



- Con una hoja de estilo externa, puede cambiar el aspecto de un sitio web completo cambiando solo un archivo.
- Cada página HTML debe incluir una referencia al archivo de hoja de estilo externo dentro del elemento `<link>`, dentro de la sección del encabezado.

```
<head>  
<link rel="stylesheet" href="mystyle.css">  
</head>
```

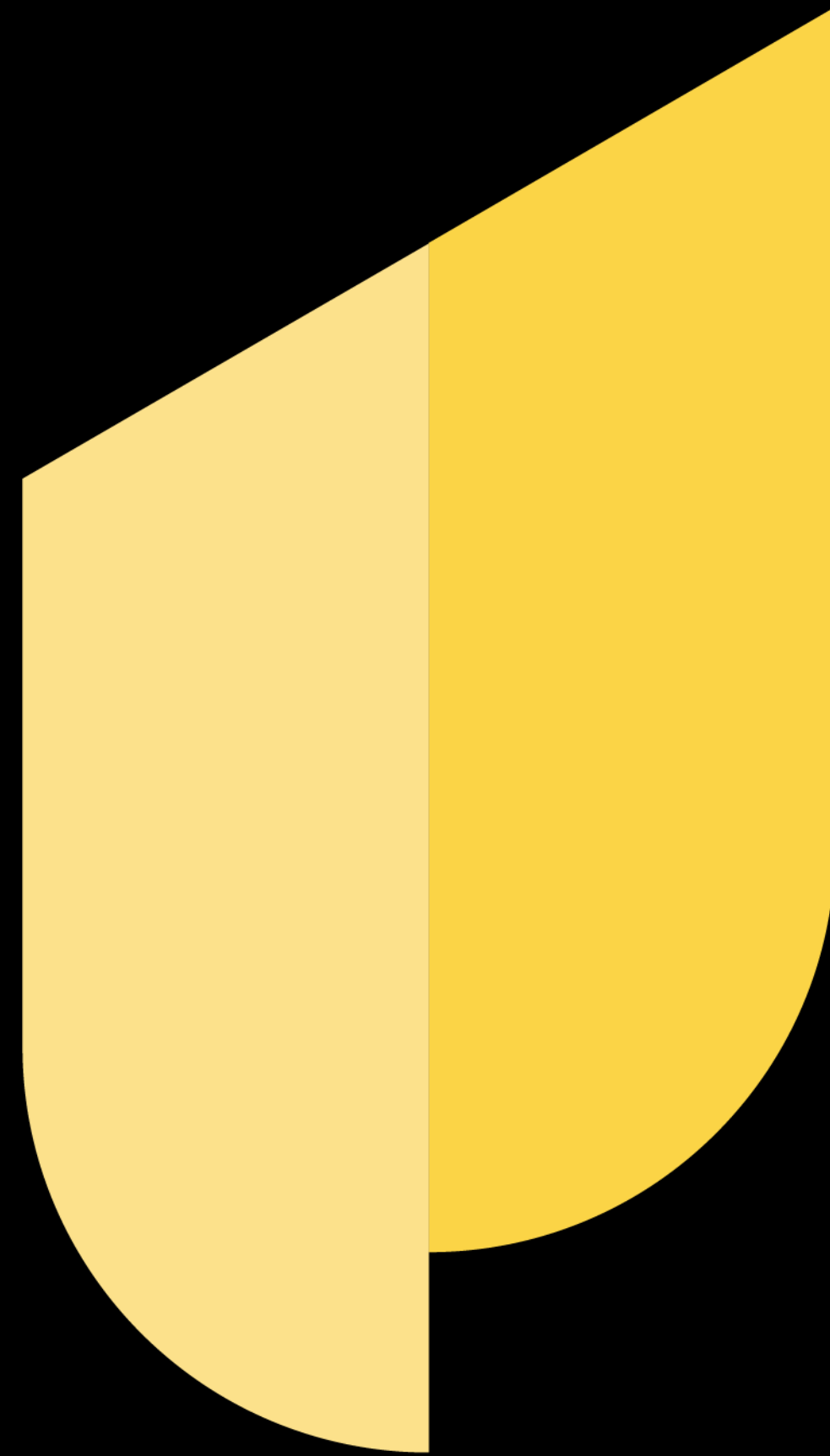
Comentarios



- Los comentarios se utilizan para explicar el código y pueden ayudar cuando edite el código fuente en una fecha posterior.
- Los navegadores ignoran los comentarios.
- Se coloca un comentario CSS dentro del elemento `<style>`, y comienza con `/*` y termina con `*/`:

```
/* This is a single-line comment */  
p {  
  color: red;  
}
```

Sintaxis



CSS





Selector

Declaración

Declaración

h1

{ color:blue; font-size:12px; }

propiedad

valor

propiedad

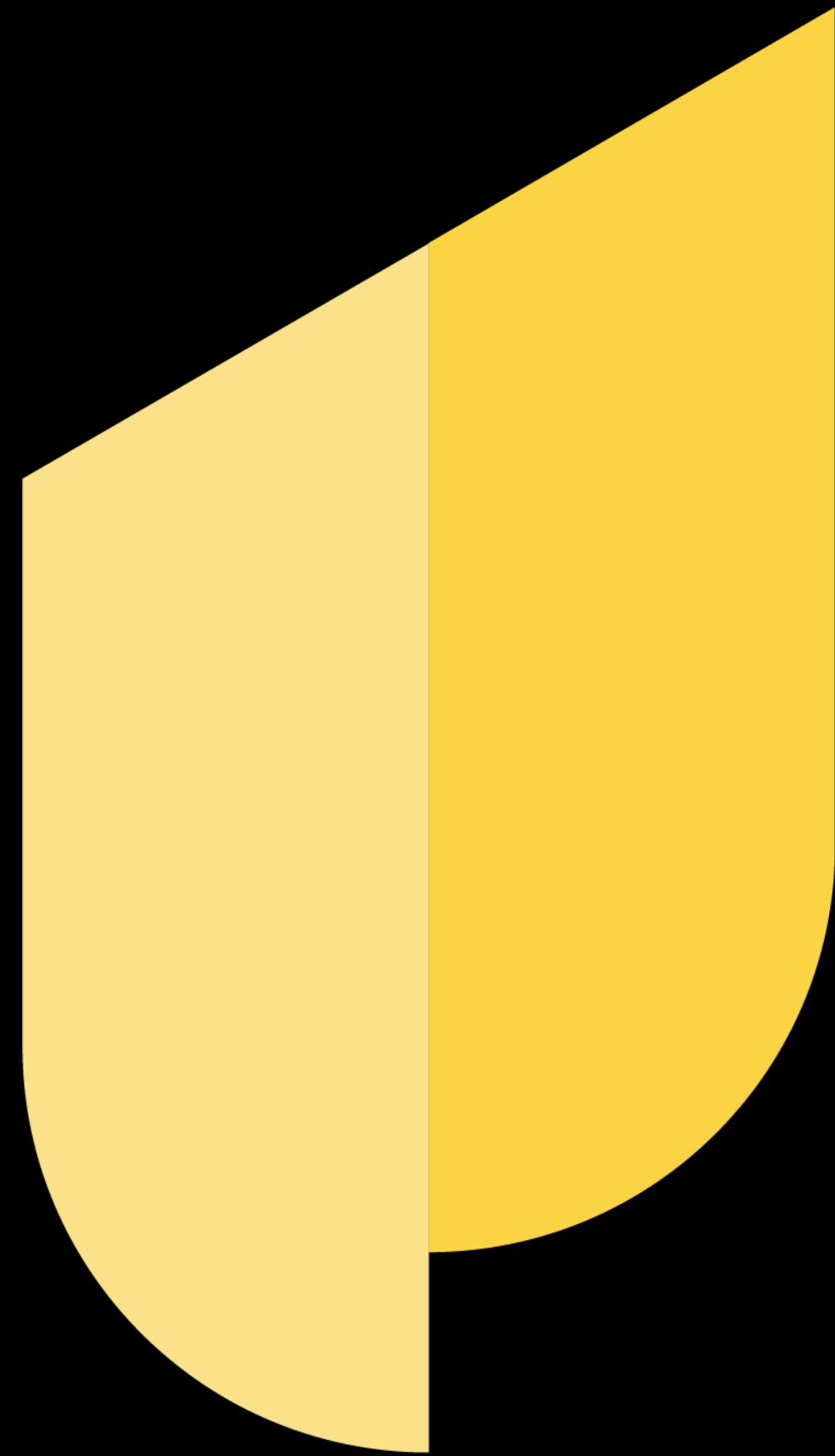
valor



# Consideraciones de las reglas CSS

- El selector apunta al elemento HTML al que desea aplicar estilo.
- El bloque de declaración contiene una o más declaraciones separadas por punto y coma.
- Cada declaración incluye un nombre de propiedad CSS y un valor, separados por dos puntos.
- Varias declaraciones de CSS se separan con punto y coma y los bloques de declaración están rodeados por llaves.

CSS



Selectores

Selector	Descripción	Sintaxis
Universal	Selecciona todos los elementos.	*
etiqueta o tipo	Selecciona todos los elementos que coinciden con el nombre del elemento especificado.	<i>tagName</i>
Clase	Selecciona todos los elementos que tienen el atributo de class especificado.	.classname
Id	Selecciona un elemento basándose en el valor de su atributo id. Solo puede haber un elemento con un determinado ID dentro de un documento.	<i>#idname</i>
Atributo	Selecciona elementos basándose en el valor de un determinado atributo.	[attr] [attr=value] [attr~=value] [attr =value] [attr^=value] [attr\$=value] [attr*=value]

CSS

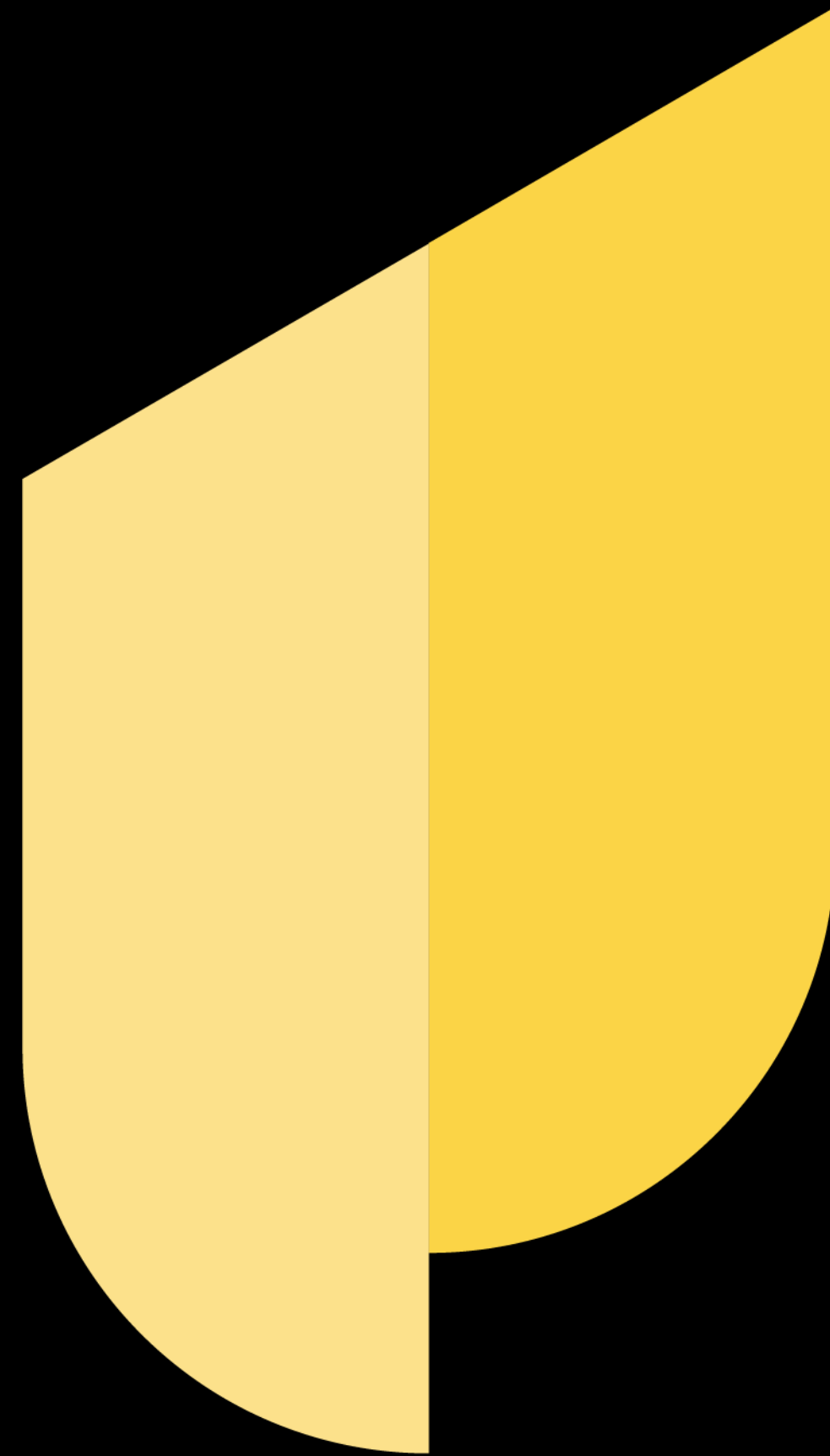


Agrupación

Selector	Descripción	Sintaxis
Combinación de selectores	Este combinador selecciona todos los elementos indicados	selector1, selector2, selector3
Descendente	El combinador (espacio) selecciona los elementos que son descendientes del primer elemento.	Selector1 selector2
Adyacente	El combinador + selecciona hermanos adyacentes. Esto quiere decir que el segundo elemento sigue directamente al primero y ambos comparten el mismo elemento padre	Selector1 + selector2
General de hermanos	El combinador ~ selecciona hermanos. Esto quiere decir que el segundo elemento sigue al primero (no necesariamente de forma inmediata) y ambos comparten el mismo elemento padre.	Selector1 ~ selector2

Selector	Descripción	Sintaxis
Hijo	El combinador > selecciona los elementos que son hijos directos del primer elemento.	Selector1 > selector2
Columna	El combinador    selecciona los elementos especificados pertenecientes a una columna.	selector1    selector2

CSS



Propiedades  
texto y fuente



# Color

- La propiedad de color se utiliza para establecer el color del texto. El color está especificado por:
  - Nombre de color
  - Valor HEX
  - Valor RGB
- El color por defecto de la pagina es definido por el selector body



# font-family

- Permite definir la familia de la fuente a aplicar.
- Sintaxis: nombre-familia|nombre-familia-genérica| \*



# font-size

- Define el tamaño de la fuente, posibles valores: xx-small | x-small | small | medium | large | x-large | xx-large | larger | smaller | longitud | porcentaje



# font-style

- Define el estilo de fuente: normal | italic | oblique



# font-variant

- Convierte a mayúsculas manteniendo un tamaño inferior:  
normal | small-caps



# font-weight

- Aplica negrita, posibles valores: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900



# CSS Text Alignment

- La propiedad text-align se utiliza para establecer la alineación horizontal de un texto.
- Un texto puede estar alineado a la izquierda o derecha, centrado o justificado (left, right aligned, centered, o justified.).



# Vertical Alignment

- La propiedad de vertical-alignment establece la alineación vertical de un elemento.
- Posibles valores: top, middle o bottom





# CSS Text Decoration

- La propiedad text-decoration se utiliza para establecer o eliminar decoraciones del texto.
- El valor text-decoration : ninguno; se utiliza a menudo para eliminar los subrayados de los enlaces.
- Otros valores usados para decorar texto son:
  - Overline
  - Line-through
  - Underline
- No se recomienda subrayar el texto que no es un enlace, ya que esto suele confundir al visitante.



# CSS Text Transformation

- La propiedad text-transform se utiliza para especificar letras mayúsculas y minúsculas en un texto.
- Puede usarse para convertir todo en letras mayúsculas o minúsculas, o poner en mayúscula la primera letra de cada palabra:
  - Uppercase
  - Lowercase
  - Capitalize



# CSS Text Spacing

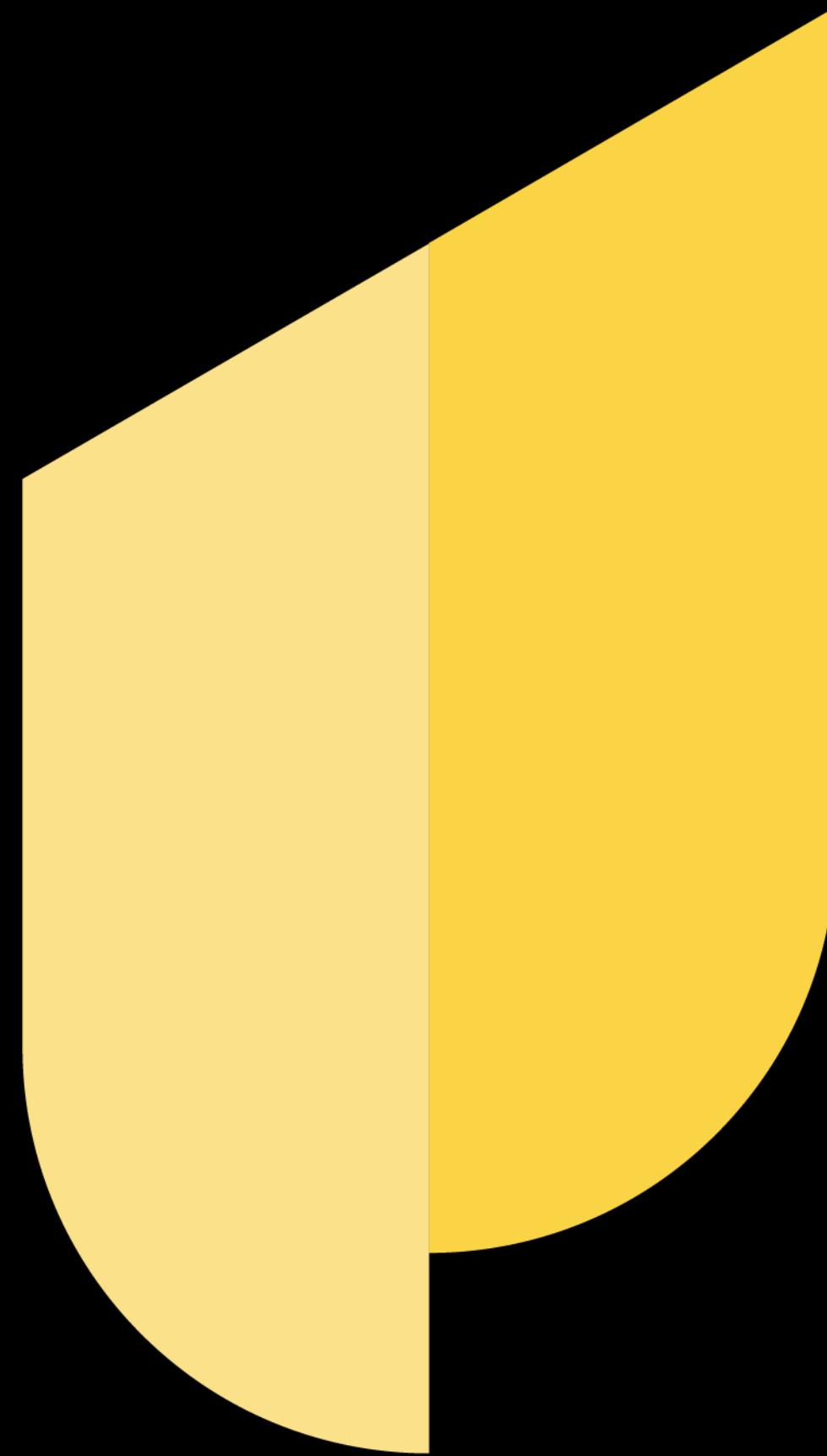
- Text Indentation: La propiedad text-indent se utiliza para especificar la sangría de la primera línea de un texto.
- Letter Spacing: La propiedad letter-spacing se utiliza para especificar el espacio entre los caracteres de un texto.
- Line Height: La propiedad line-height se utiliza para especificar el espacio entre líneas.
- Word Spacing: La propiedad de espaciado entre palabras se usa para especificar el espacio entre las palabras en un texto.
- White Space: La propiedad de white-space especifica cómo se maneja el espacio en blanco dentro de un elemento.

# CSS Text Shadow

- La propiedad text-shadow agrega sombra al texto.
- En su uso más simple, se especifica la sombra horizontal (2px) y la sombra vertical (2px)
- También se puede agregar una longitud como efecto de blur en un tercer parámetro
- También se puede agregar un color como cuarto parámetro



CSS



Propiedades  
Bordes

Propiedad	Descripción	Valores
border-width	Establece el ancho de los cuatro bordes de un elemento. Esta propiedad puede tener de uno a cuatro valores.	<ul style="list-style-type: none"><li>• Medida, thin, medium, thick</li><li>• border-width: thin medium thick 10px;</li></ul>
border-style	Establece el estilo de los cuatro bordes de un elemento. Esta propiedad puede tener de uno a cuatro valores y es <b>obligatoria</b>	<ul style="list-style-type: none"><li>• none hidden dotted dashed solid double groove ridge inset outset</li><li>• border-style: dotted;</li></ul>
border-color	Establece el color de los cuatro bordes de un elemento. Esta propiedad puede tener de uno a cuatro valores.	<ul style="list-style-type: none"><li>• Color</li><li>• border-color: red green blue;</li></ul>
Border-radius	Propiedad abreviada para aplicar redondeo de bordes	
Border	Propiedad abreviada para aplicar borde	border: border-width border-style border-color



# Margin

- La propiedad margin establece los márgenes de un elemento y es una propiedad abreviada de las siguientes propiedades:
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left



# Padding

- El relleno crea espacio adicional dentro de un elemento, mientras que el margen crea un espacio adicional alrededor de un elemento.
- El padding es una propiedad abreviada para:
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left



# Combinaciones de valores para margin y padding

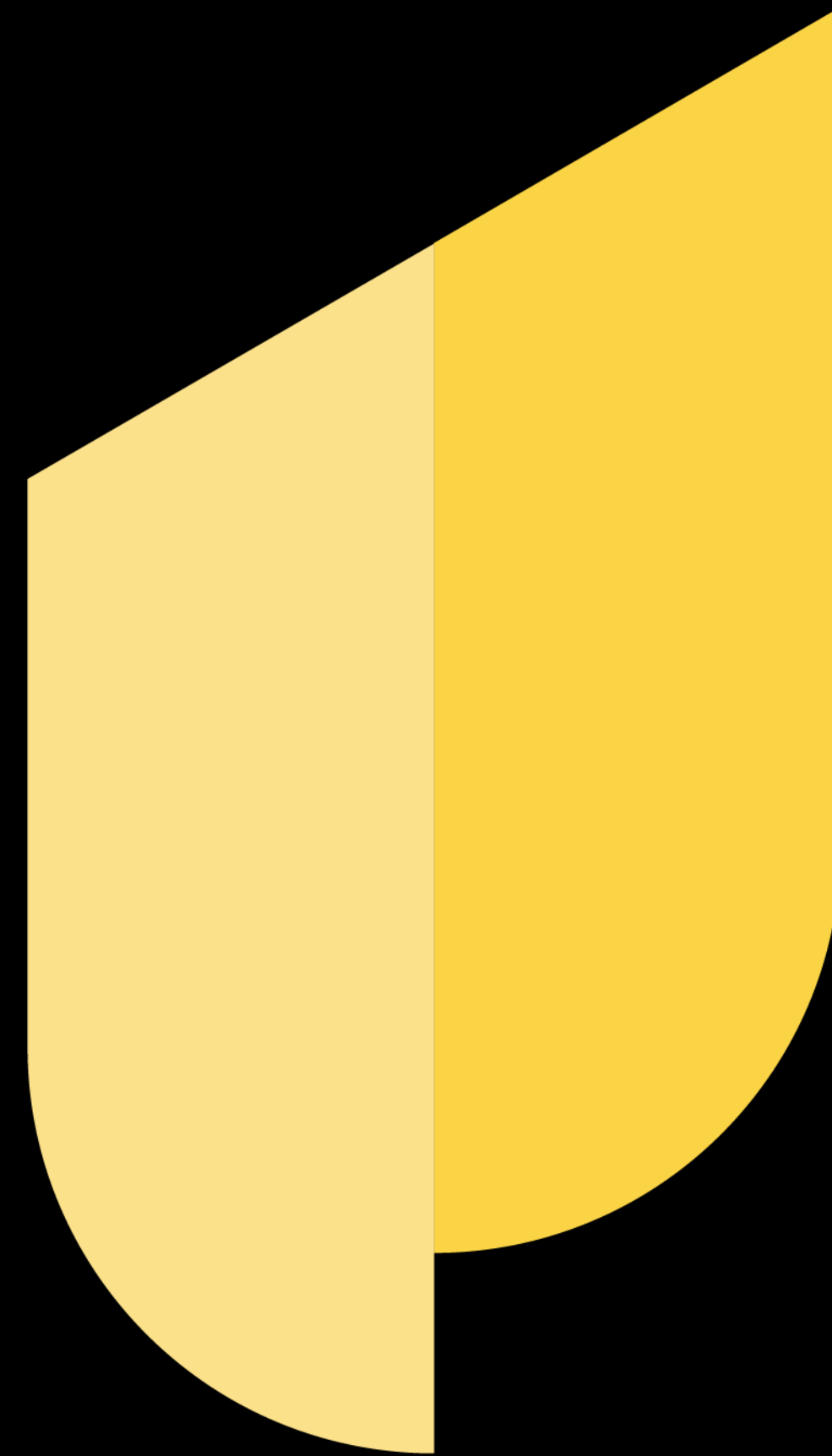
- Si se definen 4 valores:
  - se aplica la medida dada para el top, right, bottom y left
  - margin: 10px 5px 15px 20px; / padding: 10px 5px 15px 20px;
- Si se definen 3 valores
  - Se aplica el primer valor al top, el segundo para el right y left y el tercero al bottom
  - margin: 10px 5px 15px; / padding: 10px 5px 15px;
- Si se definen 2 valores
  - El primer valor se aplica al top y bottom y el segundo al right y left
  - margin: 10px 5px; / padding: 10px 5px;
- Si se define un valor
  - Se aplica el valor a los 4 márgenes

# Nota

- Tanto para el margen como para el relleno se permiten valores negativos.



CSS



Altos y ancho

- Las propiedades de height y width de CSS se utilizan para establecer el alto y el ancho de un elemento.
- La propiedad CSS max-width se utiliza para establecer el ancho máximo de un elemento.
- La propiedad CSS max-height se utiliza para establecer el alto máximo de un elemento.
- La propiedad CSS min-width se utiliza para establecer el ancho mínimo de un elemento.
- La propiedad CSS min-height se utiliza para establecer el alto mínimo de un elemento.



CSS



Fondos

- Las propiedades de fondo de CSS se utilizan para agregar efectos de fondo a los elementos
- Propiedades CSS:
  - background-color
  - background-image
  - background-repeat
  - background-attachment
  - background-position
  - background (propiedad shorthand)

# background-color

- La propiedad background-color especifica el color de fondo de un elemento.
- Con CSS, un color suele especificarse mediante:
- un nombre de color válido, como "rojo"
- un valor HEX, como "# ff0000"
- un valor RGB, como "rgb (255,0,0)"

```
body {  
    background-color: lightblue;  
}
```

# Opacity

- La propiedad de Opacity especifica la opacidad / transparencia de un elemento.
- Puede tomar un valor de 0,0 a 1,0. Cuanto 0,1 sea el valor, más transparente





# Transparencia con RGBA

- Si no se desea aplicar opacidad a los elementos secundarios, se utiliza valores de color RGBA.
- El siguiente ejemplo establece la opacidad del color de fondo y no del texto:



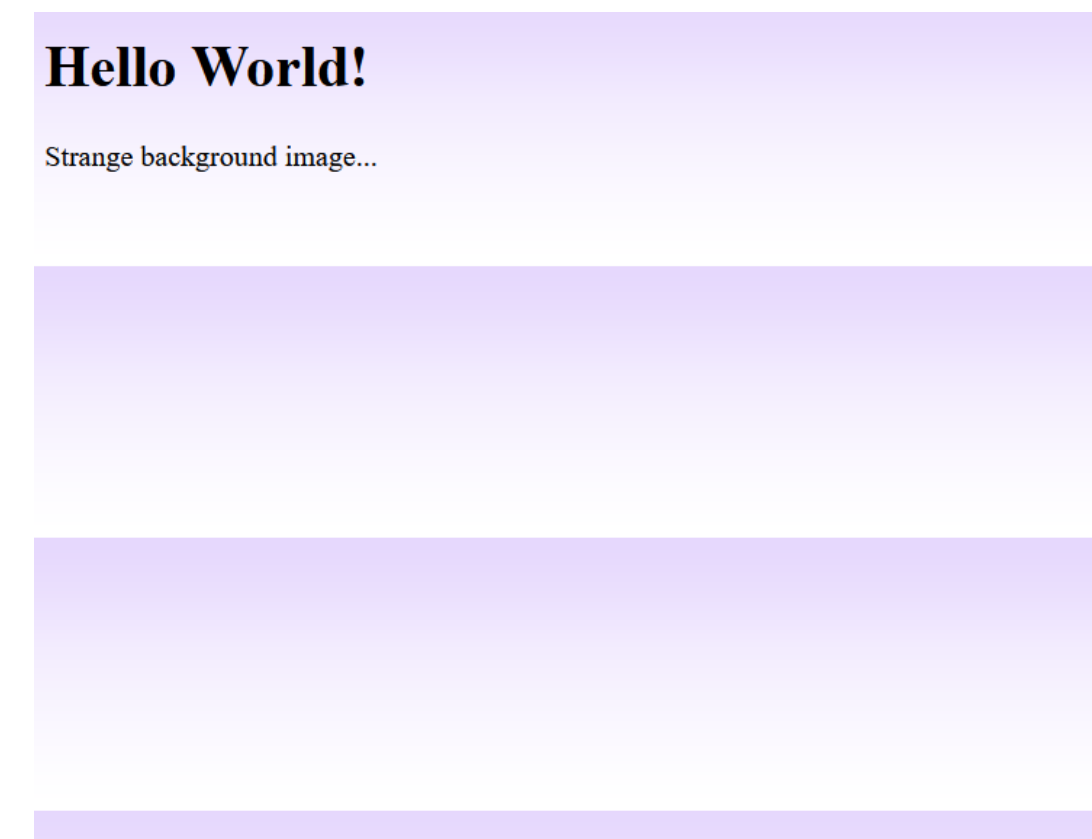
# background-image

- La propiedad background-image especifica una imagen para usar como fondo de un elemento.
- De forma predeterminada, la imagen se repite para que cubra todo el elemento.
- Cuando use una imagen de fondo, use una imagen que no altere el texto.

```
body {  
    background-image: url("paper.gif");  
}
```

# background-repeat

- De forma predeterminada, la propiedad background-image repite una imagen tanto horizontal como verticalmente.
- Algunas imágenes deben repetirse solo horizontal o verticalmente, o se verán extrañas, como esta:



# background-repeat

- Si la anterior se repite solo horizontalmente (`background-repeat: repeat-x;`), el fondo se verá mejor:

## Hello World!

Here, a background image is repeated only horizontally!

# CSS background-position

- La propiedad background-position se utiliza para especificar la posición de la imagen de fondo.
- background-position: [ <percentage> | <length> | left | center | right ] [ <percentage> | <length> | top | center | bottom ]
- background-position: [ top | center | bottom ];

# background-attachment

- La propiedad de background-attachment especifica si la imagen de fondo debe desplazarse o ser fija (no se desplazará con el resto de la página)
- background-attachment: scroll | fixed | inherit

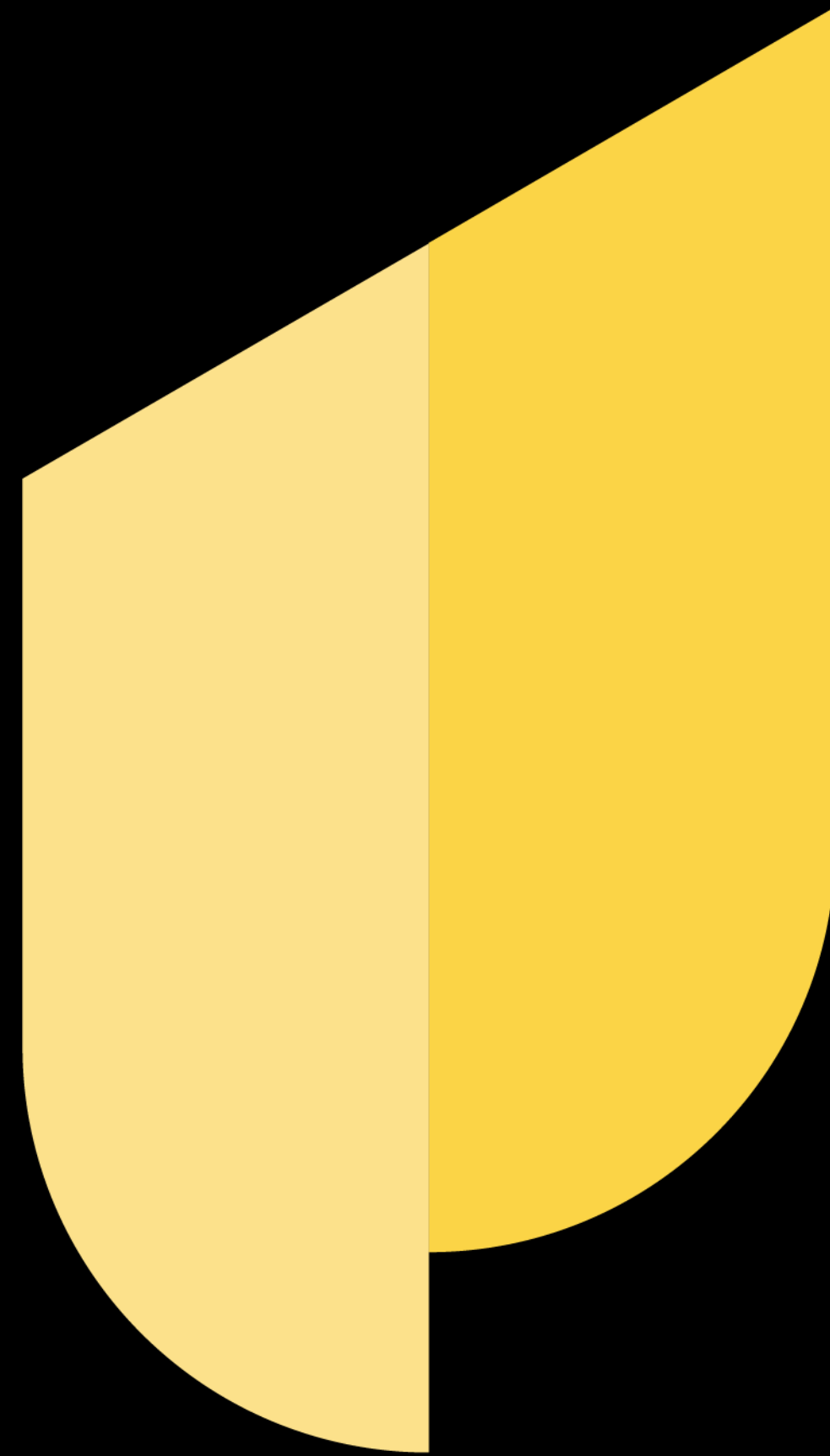


# Background Shorthand

- Para acortar el código, también es posible especificar todas las propiedades de fondo en una sola propiedad. A esto se le llama propiedad shorthand.

```
background: #ffffff url("img_tree.png") no-repeat right top;
```

CSS



Prefijos



- Actualmente los navegadores tienen implementadas muchas de las nuevas características de CSS3 utilizando sus propias versiones de cada propiedad mediante prefijos.
- Esto se hace así para evitar los posibles errores ocasionados por las primeras implementaciones que aún no son estables. Por ello, los navegadores proporcionan valores utilizando sus prefijos propios y una declaración sin prefijo.

# Prefijos

Prefijo	Navegador
-moz-	Firefox
-webkit-	Safari y Chrome
-o-	Opera
-khtml-	Konqueror
-ms-	Internet Explorer
-chrome-	Google Chrome

- Así por ejemplo para transformar un elemento en Firefox, es necesario utilizar la propiedad `-moz-transform`; en los navegadores basados en WebKit, como Safari y Google Chrome, se utiliza la propiedad `-webkit-transform`.
- Por este motivo, no es de extrañar que en algunos casos, tengamos que añadir hasta cuatro líneas de código para una única propiedad CSS.

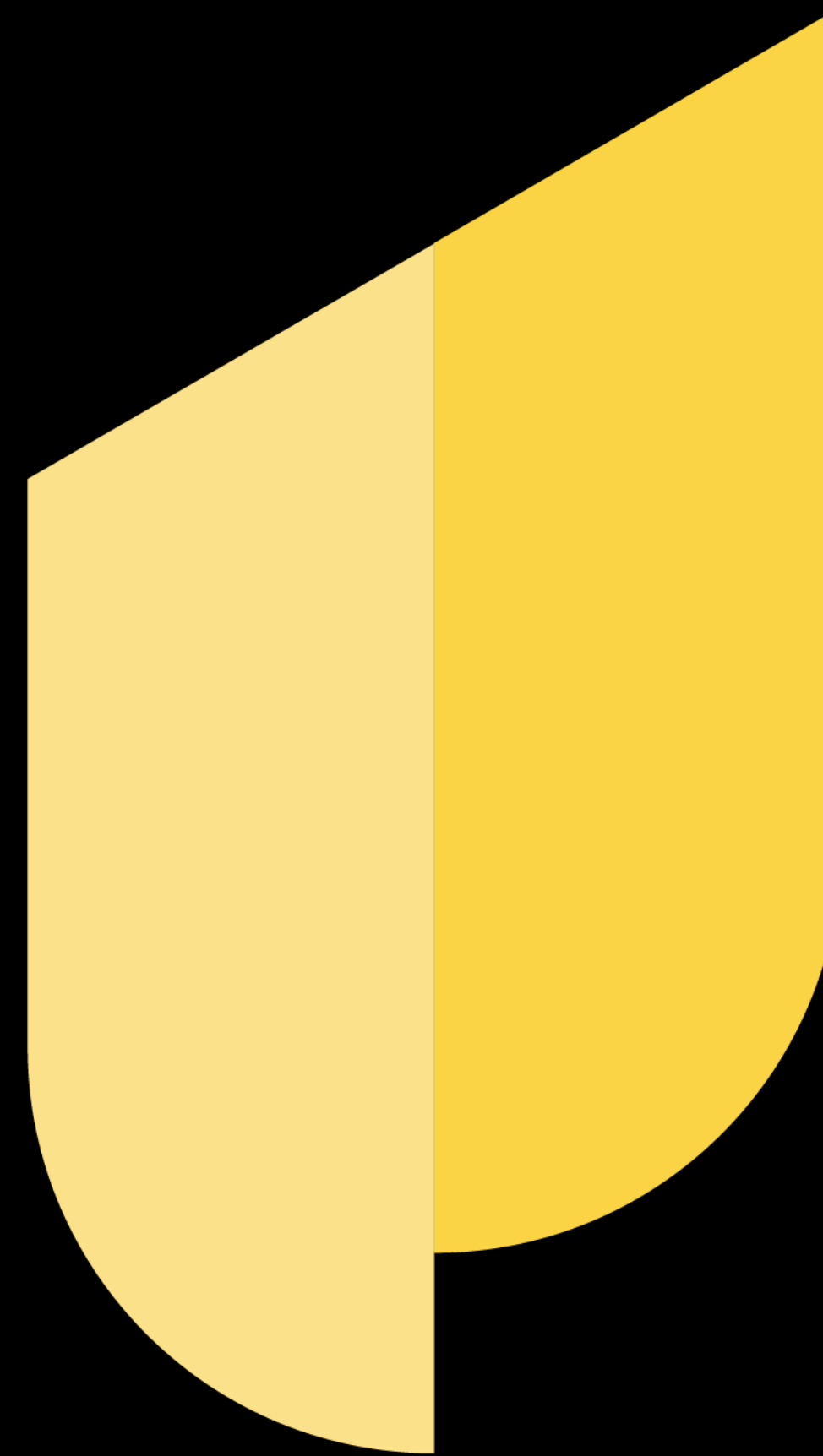
CSS



Filtros

- La propiedad de filter CSS agrega efectos visuales (como desenfoque y saturación) a un elemento, normalmente img.
- La propiedad de filter no es compatible con Internet Explorer, Edge 12 o Safari 5.1 y versiones anteriores.
- filter: none | blur(px) | brightness(%) | contrast(%) | drop-shadow(h-shadow v-shadow blur color) | grayscale(%) | hue-rotate(deg) | invert(%) | opacity(%) | saturate(%) | sepia(#) | url(xml);

CSS



Transformaciones

- La propiedad aplica una transformación 2D o 3D a un elemento.
- Esta propiedad le permite rotar, escalar, mover, sesgar, etc. elementos.
- `transform: none|transform-functions|initial|inherit;`

# transformaciones

- La propiedad aplica una transformación 2D o 3D a un elemento.
- Esta propiedad le permite rotar, escalar, mover, sesgar, etc. elementos.
- `transform: none|transform-functions|initial|inherit;`

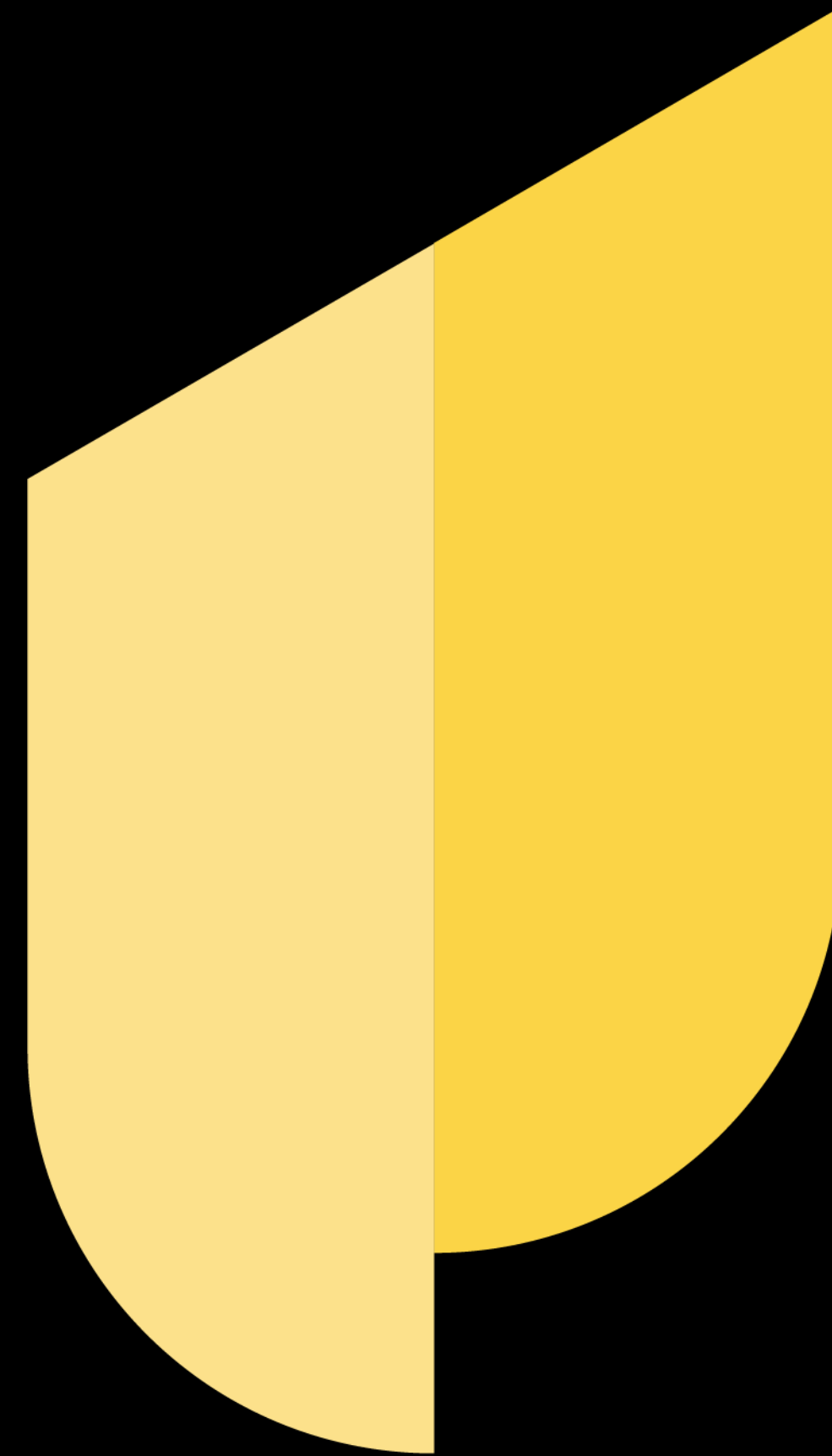


# transformaciones

- rotate(deg);
- rotateX(deg);
- rotateY(deg);
- rotateZ(deg);
- scale(sx[, sy]);
- scaleX(sx);
- scaleY(sy);

- skew(ax[, ay]);
- skewX(deg);
- skewY(deg);
- translate(tx[, ty]);
- translateX(tx);
- translateY(ty);

CSS



Pseudoclasses

- Una pseudo-clase se usa para definir un estado especial de un elemento.
- En CSS, un selector de pseudo-clase apunta a elementos dependiendo de su estado en lugar de en su información en el árbol del documento.
- Por ejemplo, el selector `a:visited` aplica estilos solamente a los links que el usuario ha visitado.
- Las pseudo-clases permiten:
  - Aplicar estilo a un elemento cuando un usuario pasa el mouse sobre él
  - Estilo de enlaces visitados y no visitados de forma diferente
  - Aplicar estilo a un elemento cuando se enfoca

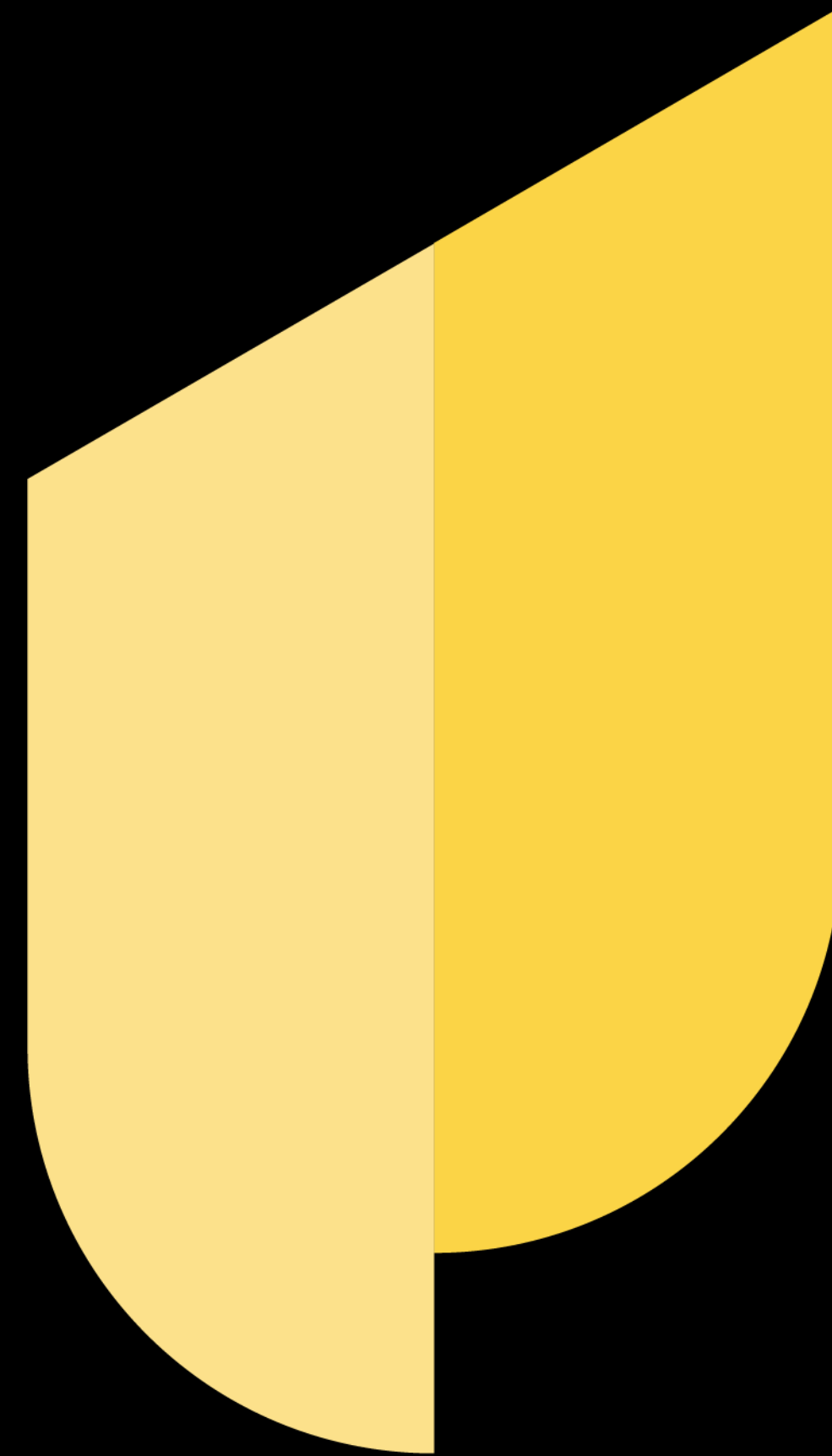
# Sintaxis

- selector:pseudo-class {
- propiedad: valor;
- }



- [:active](#)
- [:checked](#)
- [:default](#)
- [:dir\(\)](#)
- [:disabled](#)
- [:empty](#)
- [:enabled](#)
- [:first](#)
- [:first-child](#)
- [:first-of-type](#)
- [:fullscreen](#)
- [:focus](#)
- [:hover](#)
- [:indeterminate](#)
- [:in-range](#)
- [:invalid](#)
- [:lang\(\)](#)
- [:last-child](#)
- [:last-of-type](#)
- [:left](#)
- [:link](#)
- [:not\(\)](#)
- [:nth-child\(\)](#)
- [:nth-last-child\(\)](#)
- [:nth-last-of-type\(\)](#)
- [:nth-of-type\(\)](#)
- [:only-child](#)
- [:only-of-type](#)
- [:optional](#)
- [:out-of-range](#)
- [:read-only](#)
- [:read-write](#)
- [:required](#)
- [:right](#)
- [:root](#)
- [:scope \(en-US\)](#)
- [:target](#)
- [:valid](#)
- [:visited](#)

CSS



Display

- La propiedad de display es la propiedad CSS más importante para controlar el diseño.
- Display especifica si / cómo se muestra un elemento.
- Cada elemento HTML tiene un valor de visualización predeterminado según el tipo de elemento que sea. El valor de visualización predeterminado para la mayoría de los elementos es block o in-line.

# Block-level Elements

- Un elemento a nivel de bloque siempre comienza en una nueva línea y ocupa todo el ancho disponible (se extiende hacia la izquierda y hacia la derecha tanto como puede).
- Algunos ejemplos son:
  - `<div>`
  - `<h1>` - `<h6>`
  - `<p>`
  - `<form>`
  - `<header>`
  - `<footer>`
  - `<section>`



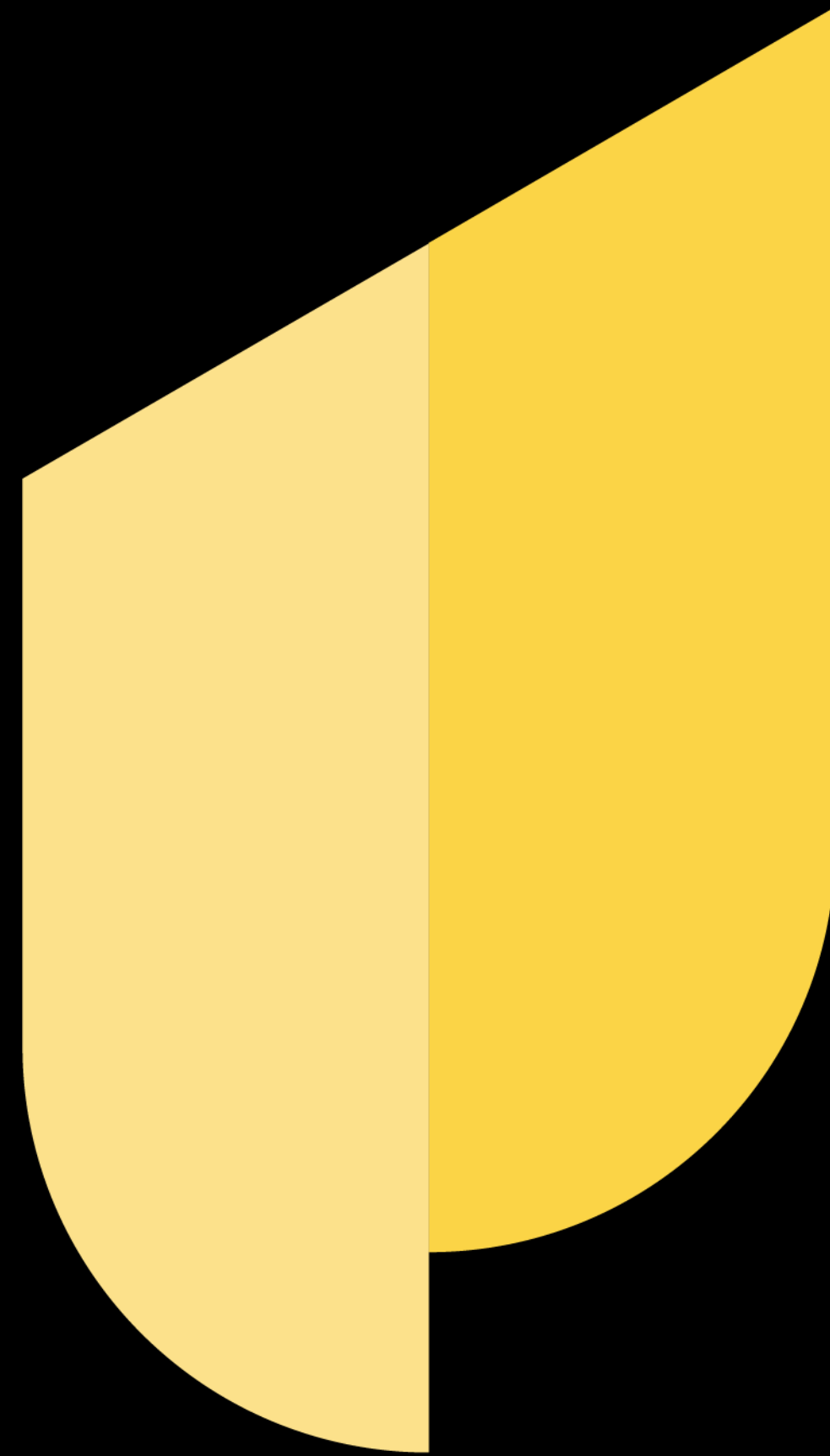
# Inline Elements

- Un elemento en línea no comienza en una nueva línea y solo ocupa el ancho necesario.
- Ejemplos:
  - `<span>`
  - `<a>`
  - `<img>`

# Display: none;

- Se usa comúnmente con JavaScript para ocultar y mostrar elementos sin eliminarlos y volver a crearlos.
- El elemento se ocultará y la página se mostrará como si el elemento no estuviera allí.

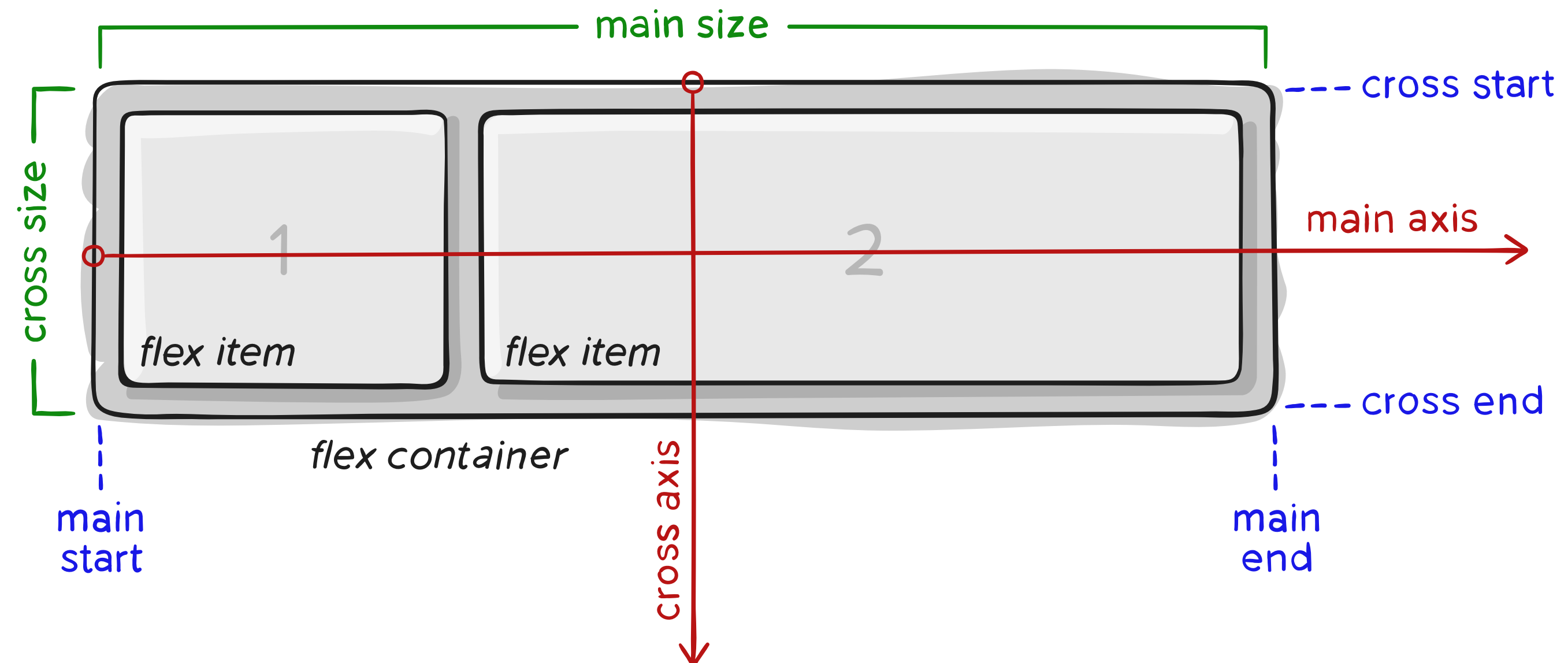
Display



Flexbox

- El Flexbox Layout(caja flexible) tiene como objetivo proporcionar una forma más eficiente de diseñar, alinear y distribuir el espacio entre los elementos de un contenedor, incluso cuando su tamaño es desconocido y / o dinámico (de ahí la palabra "flexible").
- La idea principal detrás del diseño flexible es darle al contenedor la capacidad de alterar el ancho / alto (y el orden) de sus elementos para llenar mejor el espacio disponible (principalmente para adaptarse a todo tipo de dispositivos de visualización y tamaños de pantalla).
- Un contenedor flexible expande los artículos para llenar el espacio libre disponible o los encoge para evitar que se desborden.

- Dado que flexbox es un módulo completo y no una sola propiedad, involucra muchas cosas, incluido todo su conjunto de propiedades.
- Algunos de ellos están destinados a establecerse en el contenedor (elemento padre, conocido como "contenedor flexible") mientras que otros están destinados a establecerse en los elementos secundarios (dichos "elementos flexibles").



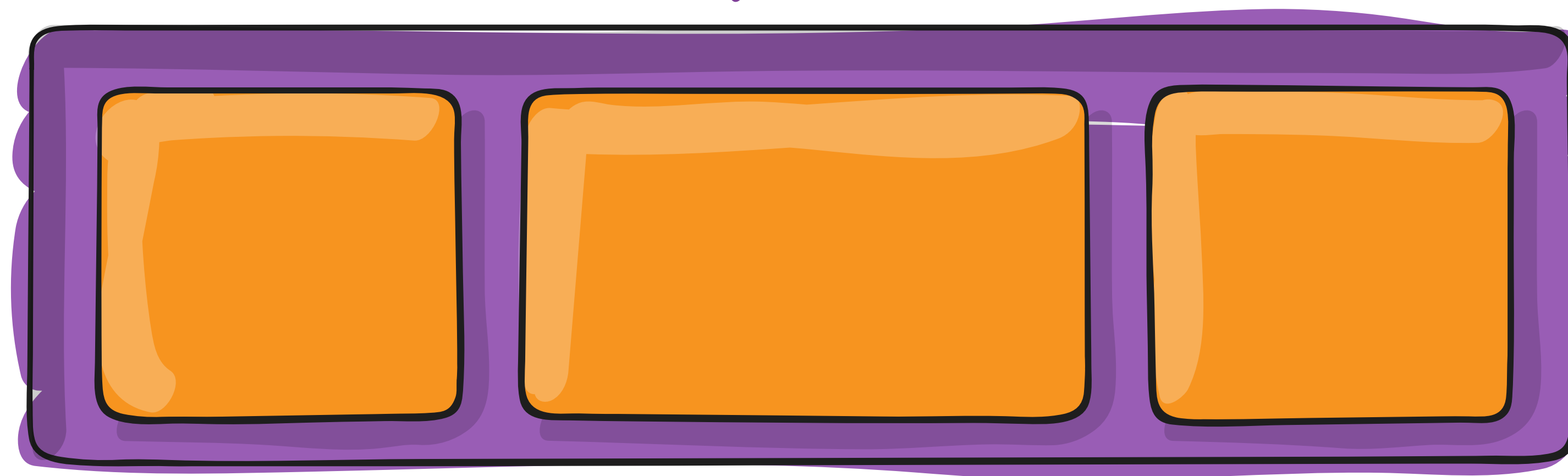
- Los elementos se distribuirán siguiendo el eje principal (desde el inicio principal hasta el final principal) o el eje transversal (desde el inicio transversal hasta el final transversal).
- eje principal: el eje principal de un contenedor flexible es el eje principal a lo largo del cual se colocan los elementos flexibles. Cuidado, no es necesariamente horizontal; depende de la propiedad flex-direction (ver más abajo).
- inicio principal | main-end: los elementos flexibles se colocan dentro del contenedor comenzando desde el inicio principal hasta el final principal.
- tamaño principal: el ancho o alto de un elemento flexible, cualquiera que sea la dimensión principal, es el tamaño principal del elemento. La propiedad de tamaño principal del elemento flexible es la propiedad "ancho" o "alto", la que se encuentre en la dimensión principal.
- eje transversal: el eje perpendicular al eje principal se denomina eje transversal. Su dirección depende de la dirección del eje principal.



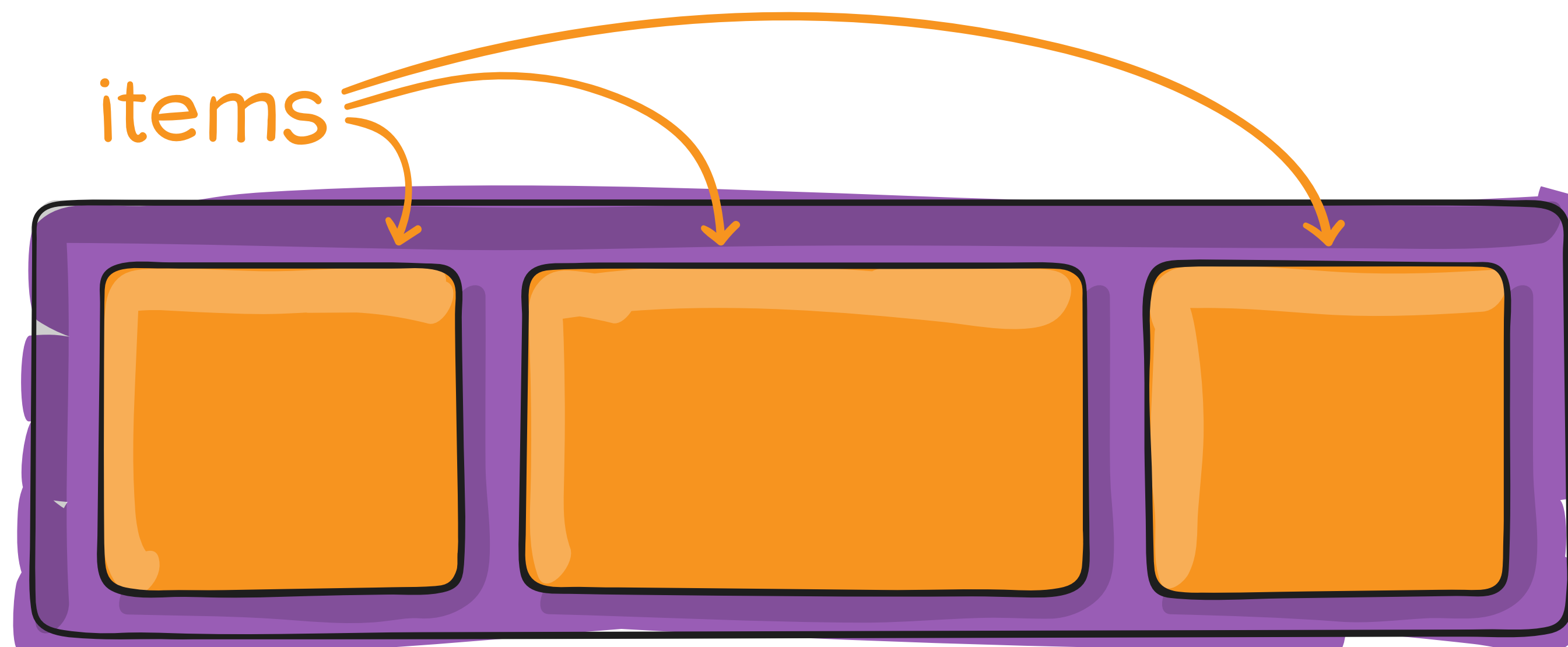
- inicio cruzado | Extremo transversal: las líneas flexibles se llenan con artículos y se colocan en el contenedor comenzando en el lado de inicio cruzado del contenedor flexible y yendo hacia el lado del extremo transversal.
- tamaño cruzado: el ancho o alto de un artículo flexible, cualquiera que esté en la dimensión cruzada, es el tamaño cruzado del artículo. La propiedad de tamaño cruzado es cualquiera de "ancho" o "alto" que esté en la dimensión cruzada.

# Display: none;

container



items





# Propiedades flex-container

- Esto define un contenedor flexible; en línea o en bloque según el valor dado. Permite un contexto flexible para todos sus hijos directos.

```
.container {  
  display: flex; /* or inline-flex */  
}
```

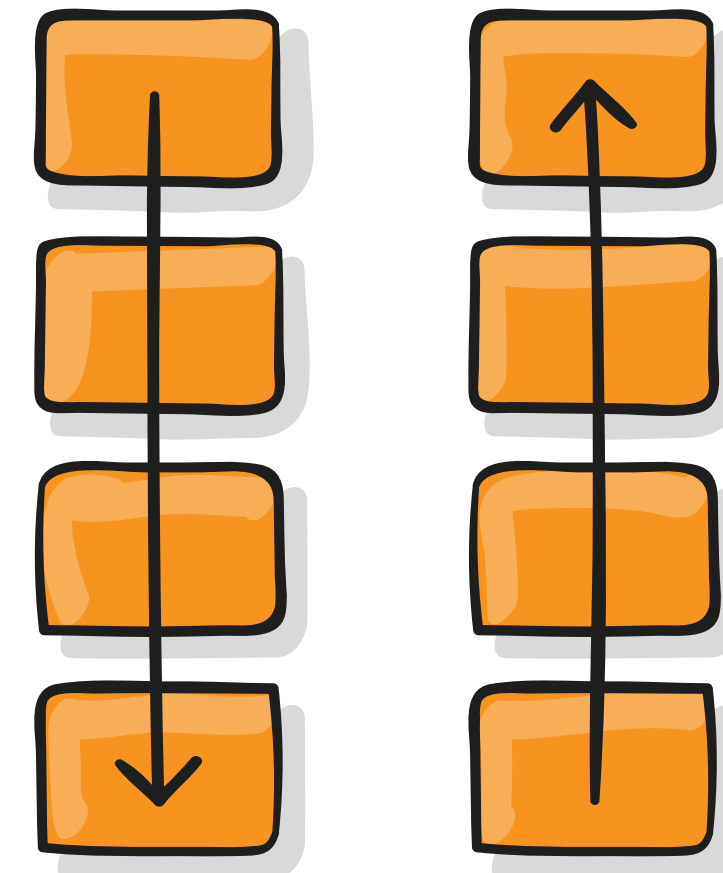
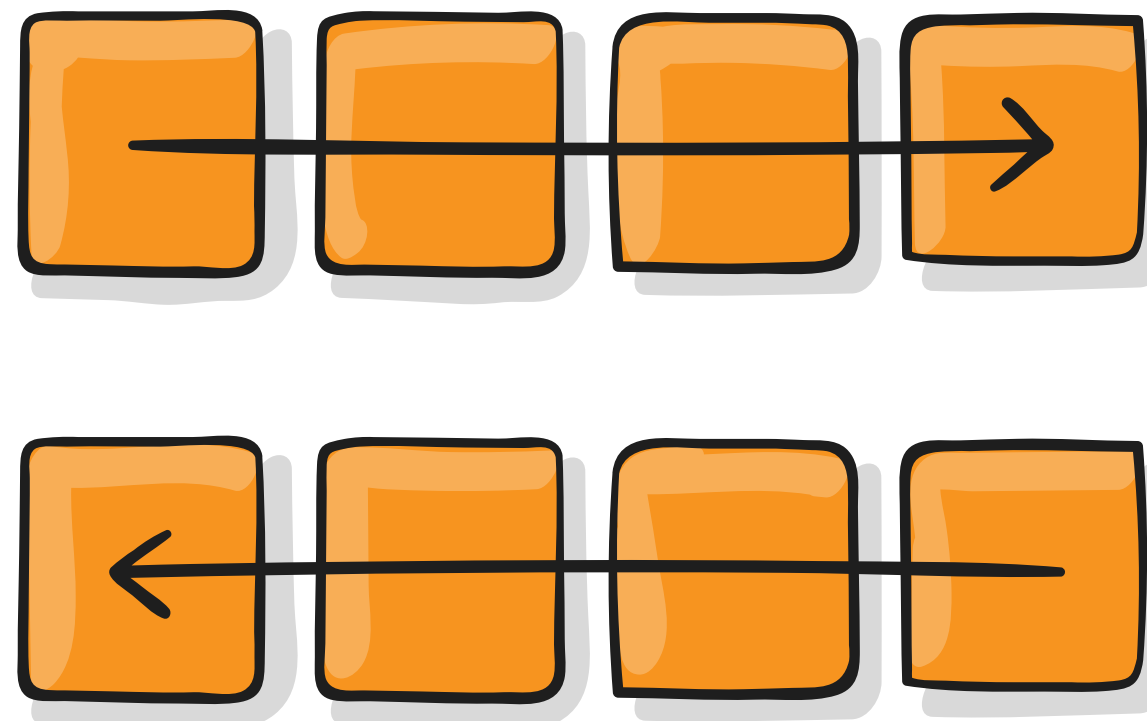
# flex-direction

- Esto establece el eje principal, definiendo así la dirección en que se colocan los elementos flexibles en el contenedor flexible.
- Flexbox es (aparte de la envoltura opcional) un concepto de diseño de una sola dirección. Piense en los elementos flexibles como distribuidos principalmente en filas horizontales o columnas verticales.



# flex-direction

- container {
- flex-direction: row | row-reverse | column | column-reverse;
- }



# flex-wrap

- De forma predeterminada, todos los elementos flexibles intentarán encajar en una línea. Puede cambiar eso y permitir que los elementos se ajusten según sea necesario con esta propiedad según:
- nowrap (predeterminado): todos los elementos flexibles estarán en una línea
- envolver: los elementos flexibles se envolverán en varias líneas, de arriba a abajo.
- wrap-reverse: los elementos flexibles se envolverán en varias líneas de abajo hacia arriba.



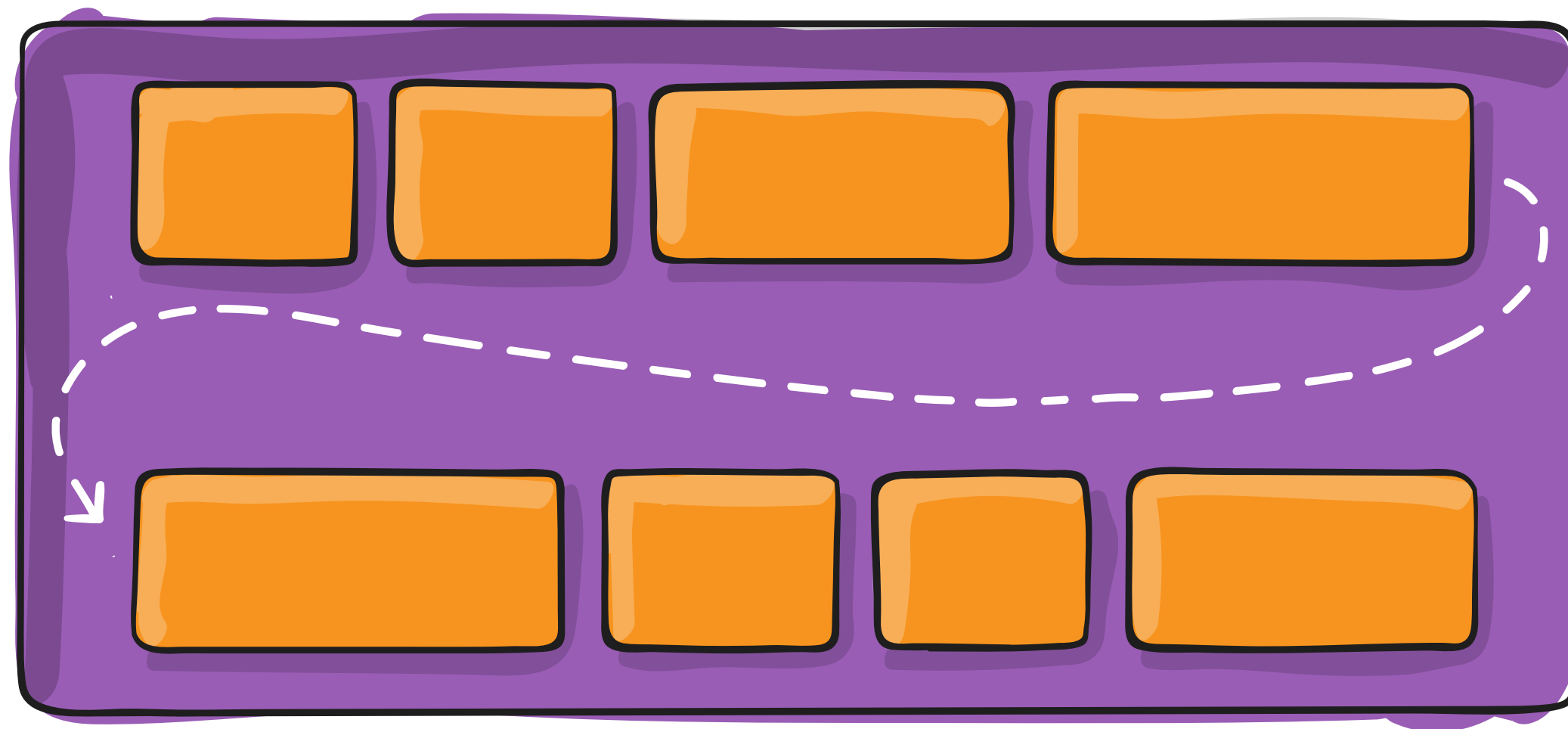
# flex-wrap

- De forma predeterminada, todos los elementos flexibles intentarán encajar en una línea. Puede cambiar eso y permitir que los elementos se ajusten según sea necesario con esta propiedad según:
  - nowrap (predeterminado): todos los elementos flexibles estarán en una línea
  - envolver: los elementos flexibles se envolverán en varias líneas, de arriba a abajo.
  - wrap-reverse: los elementos flexibles se envolverán en varias líneas de abajo hacia arriba.



# flex-wrap

- .container {
- flex-wrap: nowrap | wrap | wrap-reverse;
- }



# justify-content

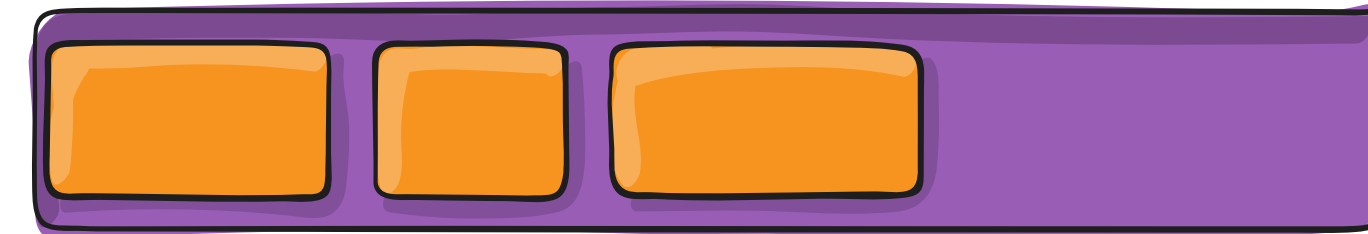
- Esto define la alineación a lo largo del eje principal.
- Ayuda a distribuir el espacio libre adicional que queda cuando todos los elementos flexibles de una línea son inflexibles o son flexibles pero han alcanzado su tamaño máximo.
- También ejerce cierto control sobre la alineación de los elementos cuando desbordan la línea.



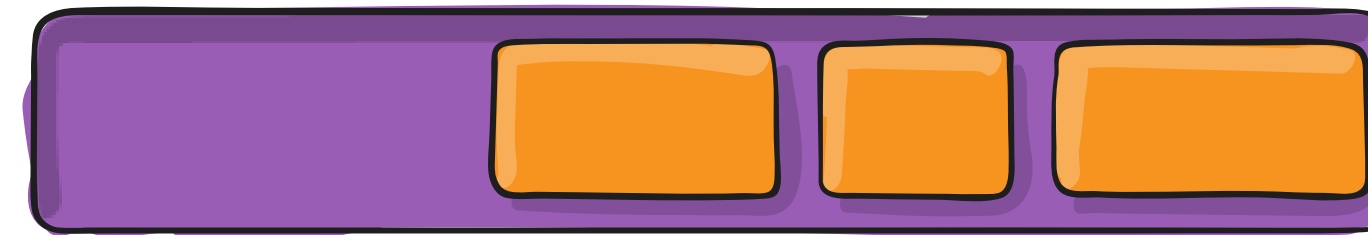
# justify-content

- .container {
- justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly | start | end | left | right ...  
+ safe | unsafe;
- }

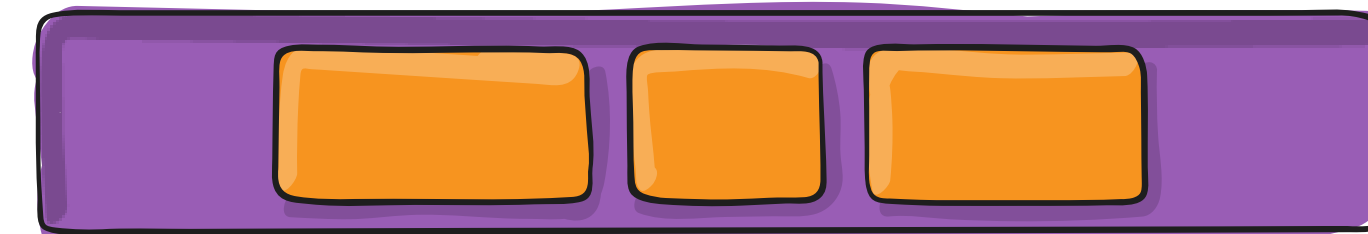
flex-start



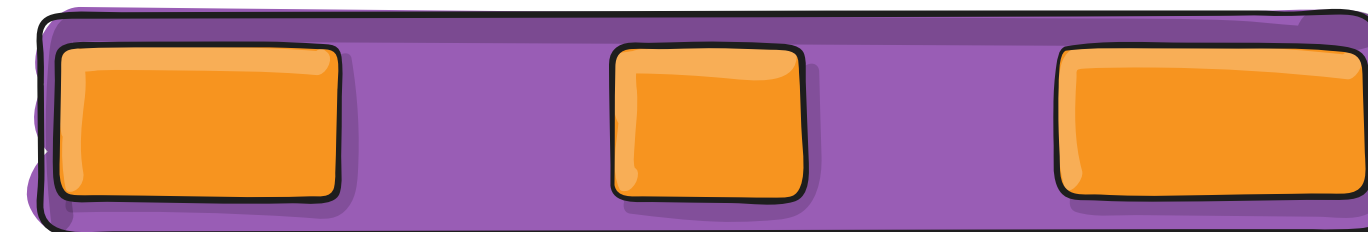
flex-end



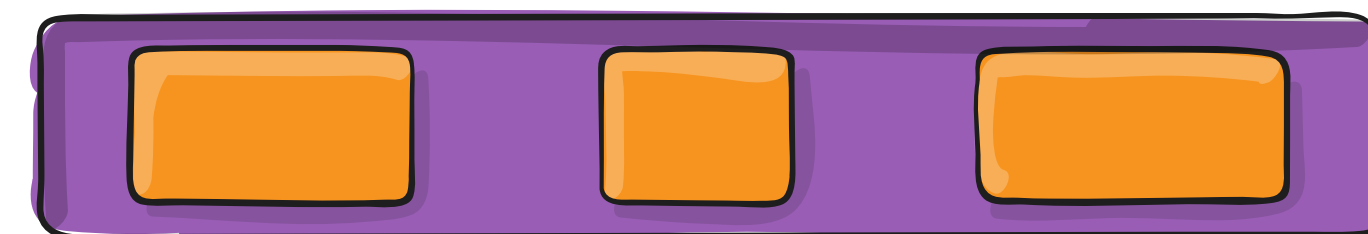
center



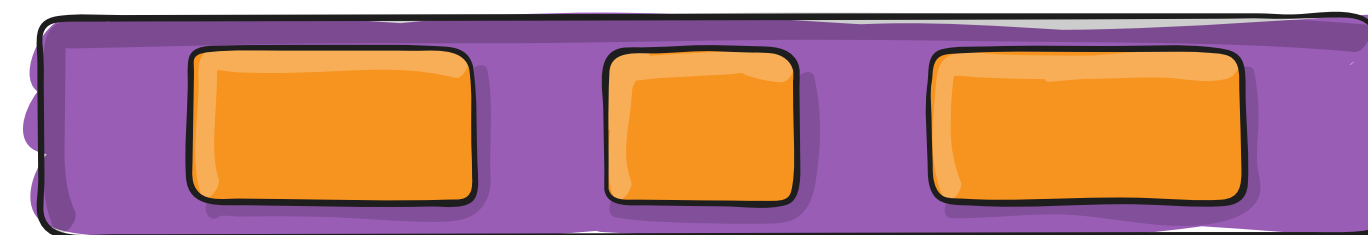
space-between



space-around



space-evenly





# align-items

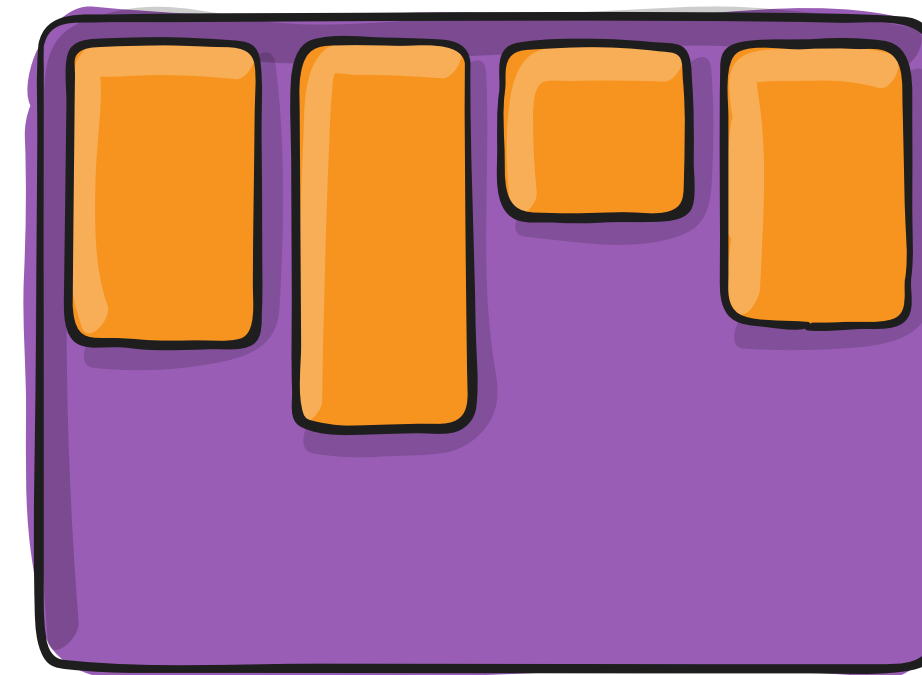
- Esto define el comportamiento predeterminado de cómo se distribuyen los elementos flexibles a lo largo del eje transversal de la línea actual.
- Piense en ello como la versión de justificación de contenido para el eje transversal (perpendicular al eje principal).



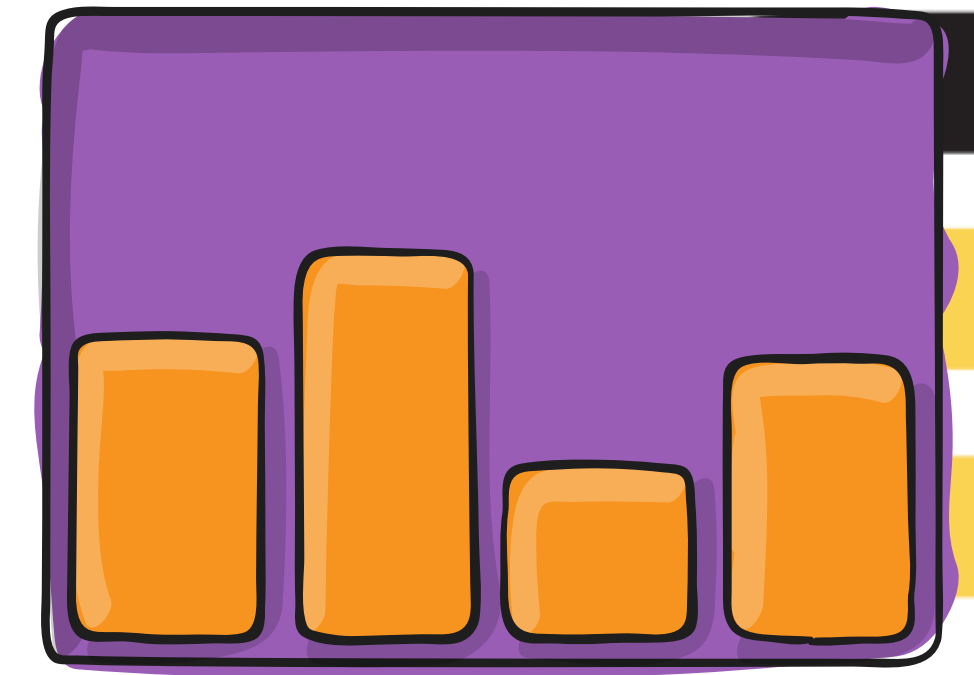
# justify-content

- .container {
- align-items: stretch | flex-start | flex-end | center | baseline | first baseline | last baseline | start | end | self-start | self-end + ... safe | unsafe;
- }

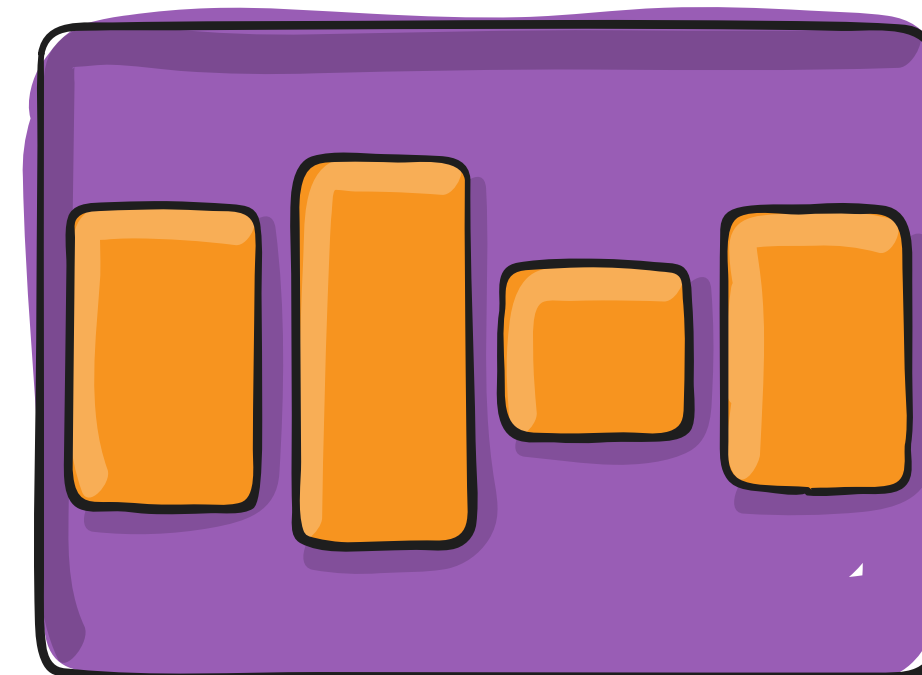
flex-start



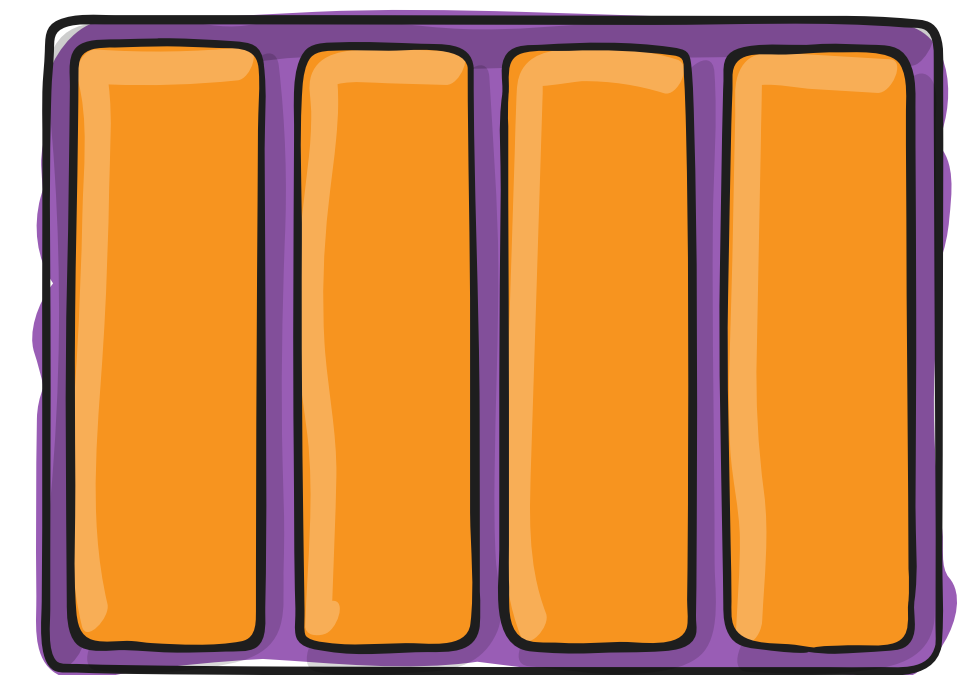
flex-end



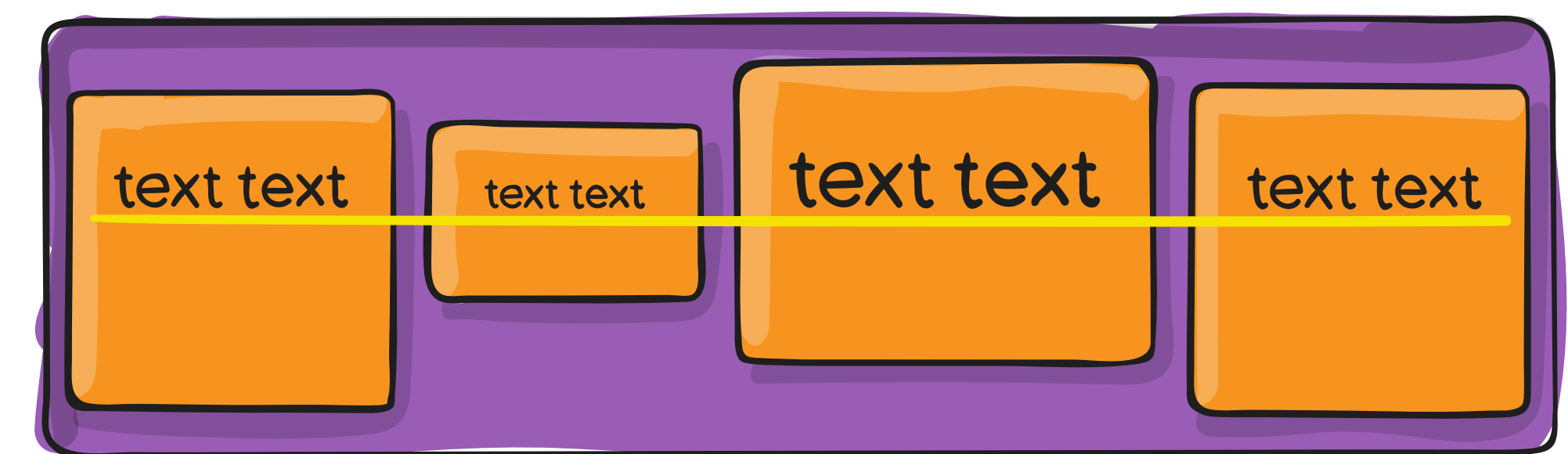
center



stretch



baseline

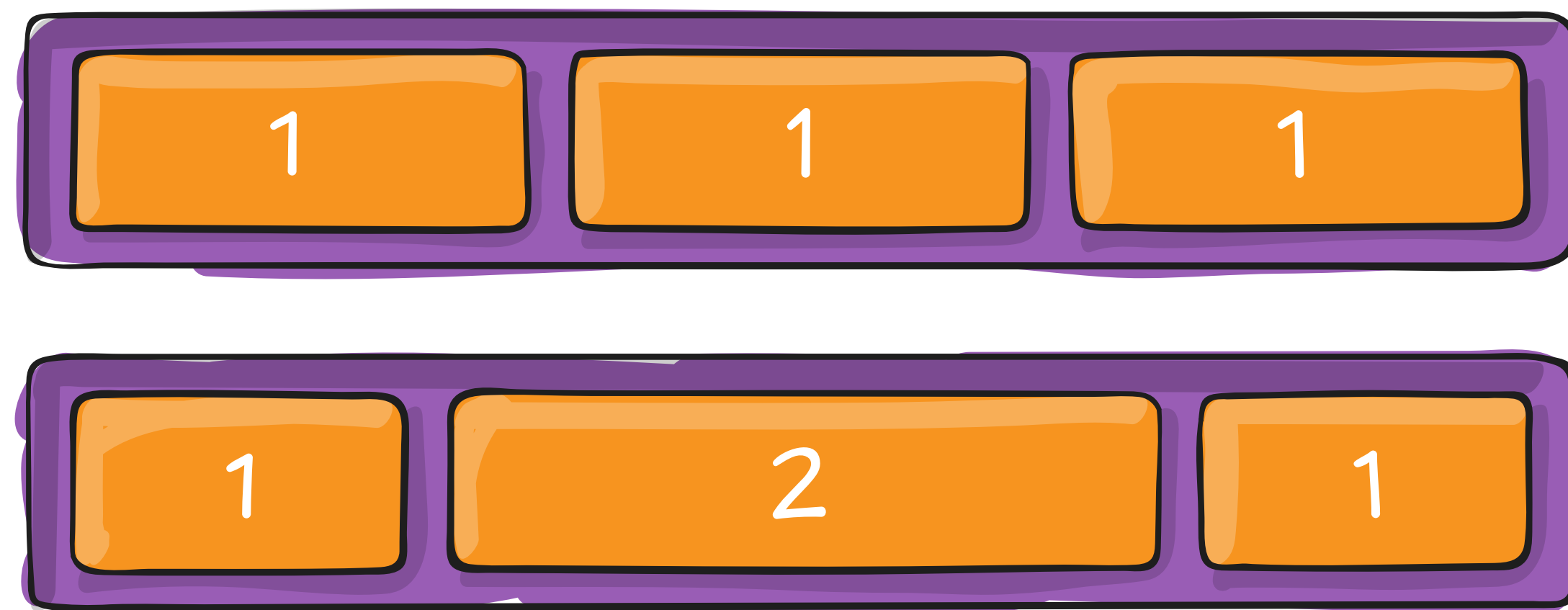


# Propiedades flex-items



# order

- De forma predeterminada, los elementos flexibles se distribuyen en el orden de origen. Sin embargo, la propiedad `order` controla el orden en que aparecen en el contenedor flexible.

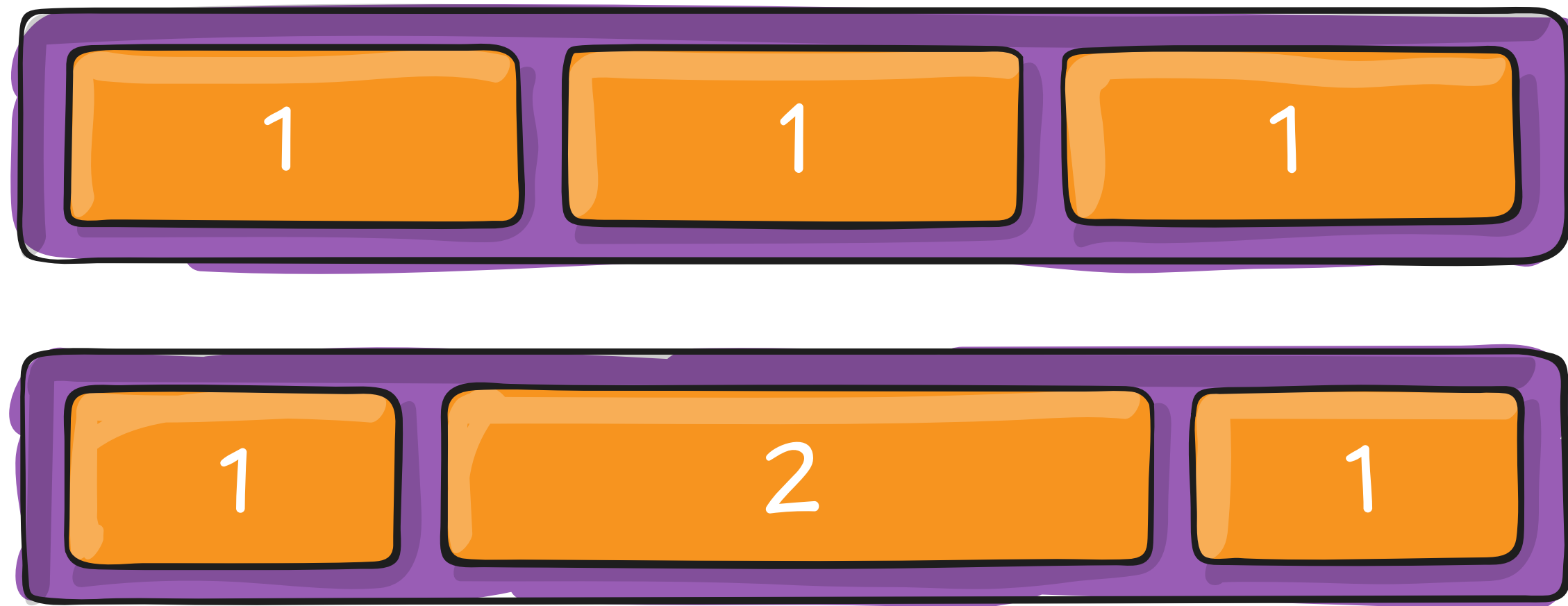


# order

- .item {
- order: 5; /\* default is 0 \*/
- }

# flex-grow

- Esto define la capacidad de un artículo flexible para crecer si es necesario. Acepta un valor sin unidades que sirve como proporción. Dicta la cantidad de espacio disponible dentro del contenedor flexible que debe ocupar el artículo.
- Si todos los elementos tienen flex-grow establecido en 1, el espacio restante en el contenedor se distribuirá por igual a todos los niños. Si uno de los niños tiene un valor de 2, el espacio restante ocuparía el doble de espacio que los demás (o lo intentará, al menos).



# flex-grow

- .item {
- flex-grow: 4; /\* default 0 \*/
- }

# flex-shrink

- Esto define la capacidad de un artículo flexible de encogerse si es necesario.
- `.item {`
- `flex-shrink: 3; /* default 1 */`
- `}`
- Numero negativos son inválidos



# align-self

- Esto permite que se anule la alineación predeterminada (o la especificada por align-items) para elementos flexibles individuales.
- .item {
- align-self: auto | flex-start | flex-end | center | baseline | stretch;
- }

CSS



Position

# Z-index

- Cuando se colocan elementos, pueden superponerse a otros elementos.
- La propiedad z-index especifica el orden de pila de un elemento (qué elemento debe colocarse delante o detrás de los demás).
- Mientras mas alto el numero mas cerca de la pantalla
- Mientras mas bajo el numero más alejado de la pantalla
- Ejemplos:
  - z-index: -1
  - z-index: 9999



# Z-index

- La propiedad z-index **solo se aplica a elementos posicionados.**
- Los elementos no posicionados siempre se van a colocar uno en encima de otro en el orden en el que aparecen en el código.
- La propiedad z-index no es heredable.

- Especifica el tipo de método de posicionamiento utilizado para un elemento.
- La propiedad position de CSS especifica cómo un elemento es posicionado en el documento.
- Posee 5 posibles valores:
  - static
  - relative
  - fixed
  - absolute
  - sticky
- Las propiedades top, right, bottom, y left determinan la ubicación final de los elementos posicionados.

# static

- El elemento es posicionado de acuerdo al flujo normal del documento. Las propiedades top, right, bottom, left, and z-index no tienen efecto. Este es el valor por defecto.



# relative

- El elemento es posicionado de acuerdo al flujo normal del documento, y luego es desplazado con relación a sí mismo, con base en los valores de top, right, bottom, and left.
- El desplazamiento no afecta la posición de ningún otro elemento; por lo que, el espacio que se le da al elemento en el esquema de la página es el mismo como si la posición fuera static.
- Este valor crea un nuevo contexto de apilamiento, donde el valor de z-index no es auto.



# absolute

- El elemento es removido del flujo normal del documento, sin crearse espacio alguno para el elemento en el esquema de la página. Es posicionado relativo a su ancestro posicionado más cercano, si lo hay; de lo contrario, se ubica relativo al bloque contenedor inicial.
- Este valor crea un nuevo contexto de apilamiento, donde el valor de z-index no es auto.





# fixed

- El elemento es removido del flujo normal del documento, sin crearse espacio alguno para el elemento en el esquema de la página. Es posicionado con relación al bloque contenedor inicial establecido por el viewport,

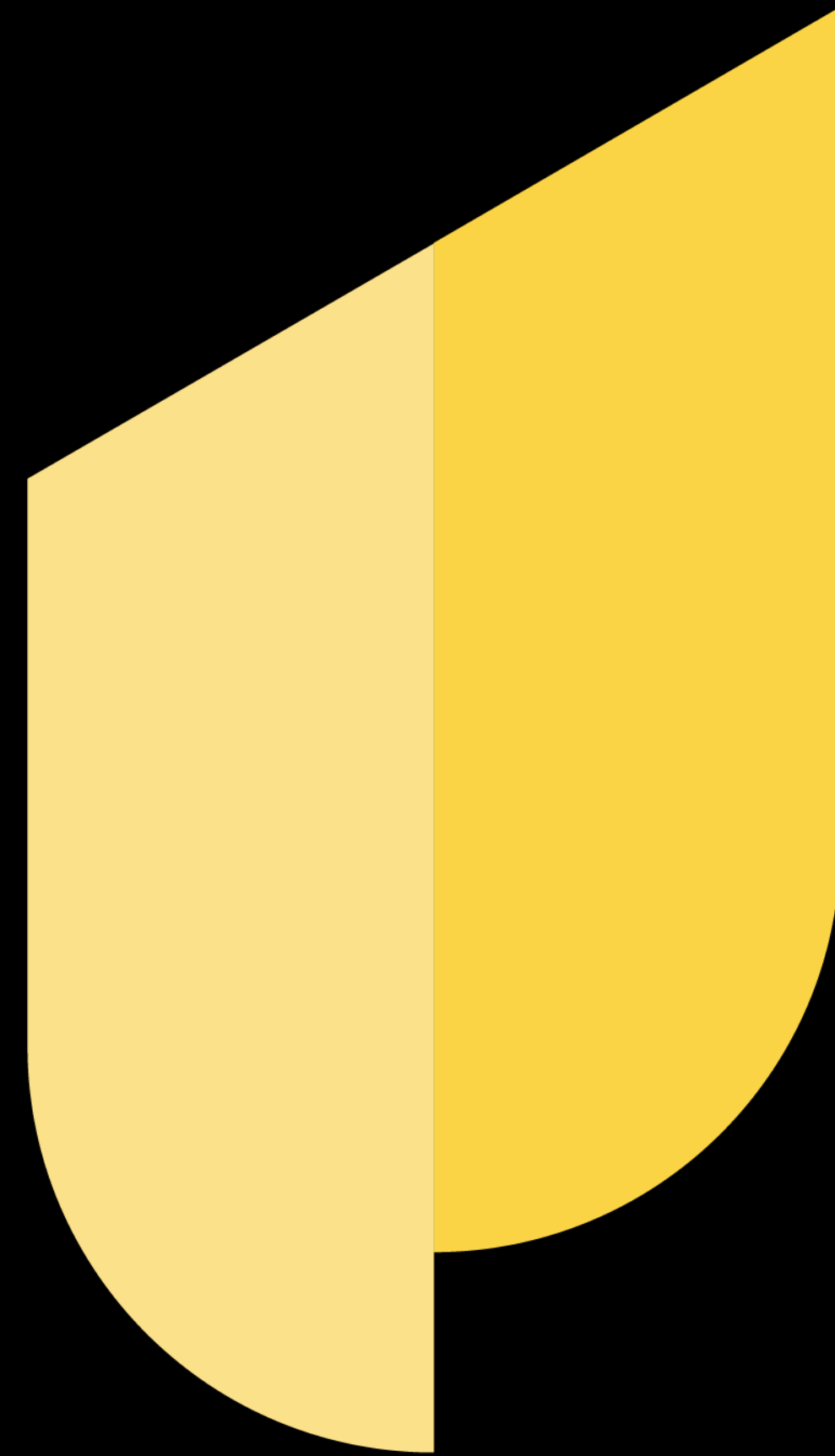


# sticky

- El elemento es posicionado de acuerdo al flujo normal del documento, y luego es desplazado con relación a su ancestro que se desplace más cercano y su bloque contenedor



MUCHAS



GRACIAS