releevant.

Juan Serrano

TEMA 1 Qué es CSS y conceptos básicos

releevant.

¿Qué vamos a ver?

Día 1 a 5

CSS mínimo para que podáis entender y maquetar el front-end de los prototipos.

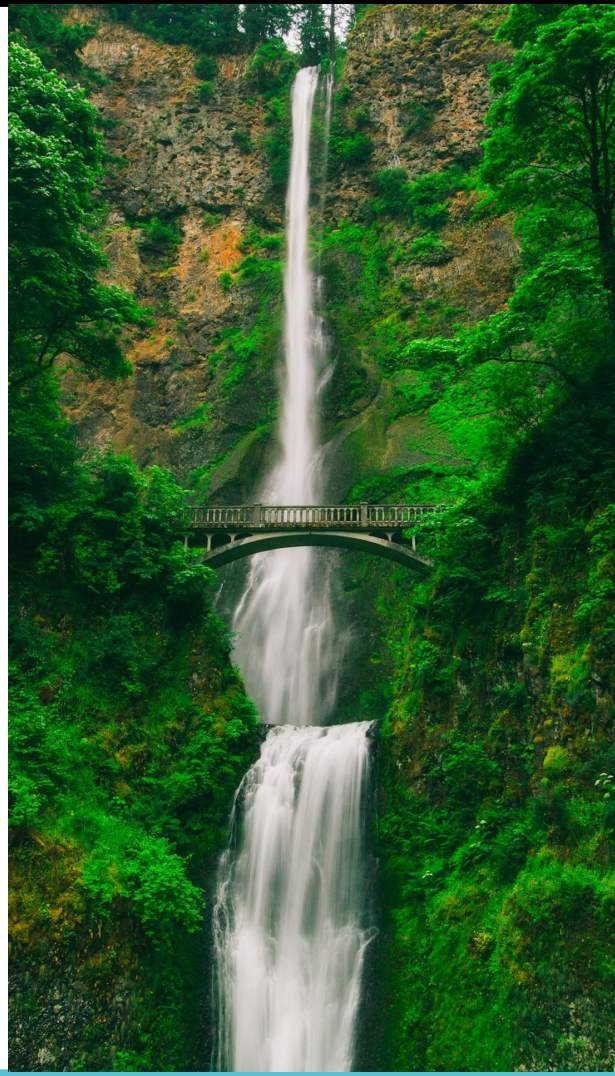
Nos centraremos en entender cómo funciona y los aspectos básicos.

Día 6

Bootstrap y sus **componentes principales**.
Os ayudará a crear **prototipos funcionales** sin necesidad de personalizarlos demasiado.

¿Qué es CSS?

CSS (siglas en inglés de Cascading Style Sheets)



La hoja de estilo en cascada o CSS, es un **lenguaje de estilos** usado para definir la presentación de un documento estructurado escrito en **HTML** y derivados.

Definición de la W3C organización internacional que desarrolla estándares para la World Wide Web

releevant.

Concepto de CSS generado por IA

Le he preguntado a DALLe si me podría representar gráficamente qué es o cómo se vería el concepto de CSS.

El resultado ha sido el siguiente.

releevant.

releevant.

releevant.

CSS es...

Lenguaje que aplica los estilos a nuestro **HTML** en forma de cascada. **(El navegador lo interpreta o "lee" de arriba hacia abajo)**.

Esta es la tecnología o estándar que vamos a utilizar para dar estilos a nuestro proyecto.

Con estilizar nos referimos a **aplicar colores**, **espacios**, **tipografías**, **disposición de elementos**, **navegación**, **estructura**, etc.

releevant.

El concepto de cascada

El concepto "cascada" en CSS significa que cuando hay varios estilos aplicados a un elemento, se da **prioridad a los estilos definidos más recientemente**.

Además, **algunos estilos se heredan** de los elementos padre a los elementos hijo, lo que facilita mantener un estilo coherente en toda la página web.

releevant.

El concepto de cascada

La lectura del archivo .css se va a realizar de arriba hacia abajo siendo los últimos estilos los que tengan prioridad.

¡Pero ojo!
Esto funciona solo con **selectores** que tengan la misma **especificidad**.

¡Vamos a verlo!

releevant.

El concepto de cascada

```css
/* Declaramos el estilo de un p */
p {
  color: red;
}

/* Sobreescribimos el estilo del p */
p {
  color: blue;
}
```

releevant.

⚒️ Herramientas

https://carbon.now.sh/

(Para hacer capturas de código y descargarlas)

https://codepen.io/

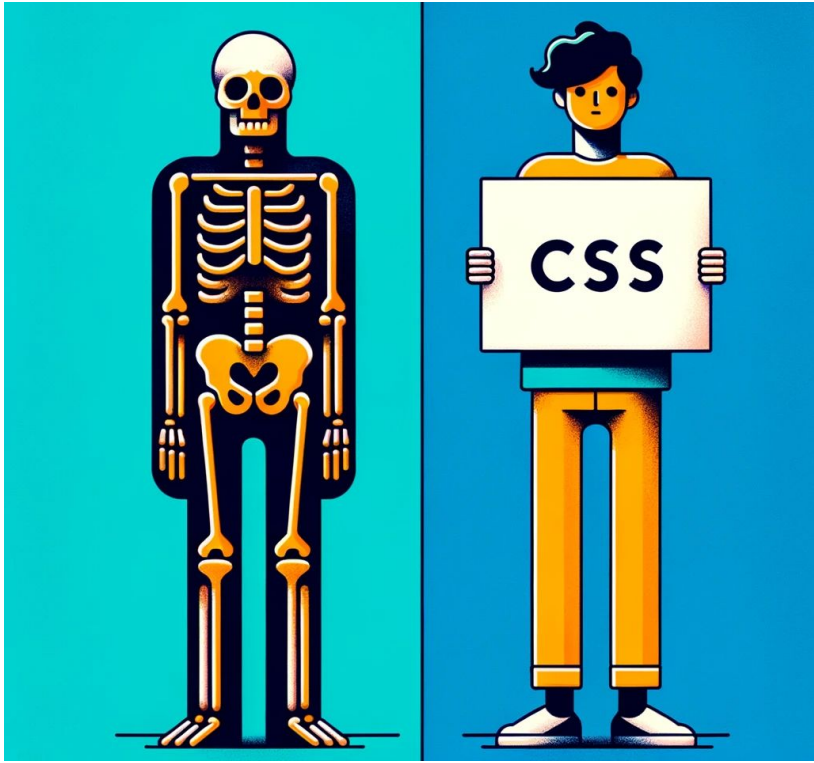
(Para probar código en línea y publicar ideas y proyectos básicos)

releevant.

HTML y CSS

Si hacemos una comparativa, podemos decir que **HTML es la estructura básica** de contenido de nuestro proyecto y **CSS viste al contenido** con un estilo visual concreto.

CSS aporta carácter e identidad a la interfaz y “la hace más usable”.



releevant.

Versiones de CSS

CSS1(1996), CSS2, CSS2.1, CSS3 y **¿CSS4?**

Al principio comenzaron sacando **“actualizaciones” por paquetes** como CSS1 o CSS2.

Desde la versión 3 se dividió en módulos para permitir que las características se desarrollen y actualicen de forma independiente.

La versión CSS3 introdujo muchas nuevas capacidades, incluyendo **bordes redondeados**, **sombras**, **gradientes**, **transiciones**, **animaciones**, **flexbox**, **grid**, y **media queries** para diseños responsivos.

releevant.

Versiones de CSS

CSS1(1996), CSS2, CSS2.1, CSS3 y **¿CSS4?**

Actualmente CSS4 está en progreso y se está estudiando la forma de lanzarlo.

Lo están enfocando desde un punto de vista más de marketing que de desarrollo de nuevas funcionalidades.

Las mejoras en **CSS4** están orientadas a **nuevos selectores** y **pseudoclases, variables** y **funciones**, y **mejoras en diseño responsivo**.

releevant.

Versiones de CSS

CSS1(1996), CSS2, CSS2.1, CSS3 y **¿CSS4?**

Para estar actualizados y tener referencias, contenidos…

- MDN Web Docs
- CSS-Tricks
- Smashing Magazine
- CSS Weekly

releevant.

Cómo dar estilos a nuestro proyecto

Hay tres formas de dar estilos a nuestro **HTML**.

1. Dentro de la etiqueta **<style></style>** en el HTML.
2. En la misma etiqueta declarando un atributo style
   ** **
3. En un archivo externo **style.css**

releevant.

Casos de uso de <style>

1. **Aplicación Local:** Ideal para estilos que son específicos para una sola página.
2. **Evitar Solicitudes HTTP Adicionales:** En páginas muy pequeñas o con muy poco CSS, puede ser más eficiente incluir el CSS directamente para evitar solicitudes HTTP adicionales.
3. **Pruebas y Desarrollo:** Útil para probar rápidamente estilos antes de moverlos a una hoja de estilo externa.
4. **Templates** o **Plantillas de Correo Electrónico**.

releevant.

Casos de uso de atributo style="" en HTML

1. **Estilos únicos y específicos:** Cuando necesitas aplicar estilos muy específicos a un solo elemento. *(Recomendable para pruebas)*
2. **Prioridad y Sobrescritura:** Los estilos en línea tienen la mayor prioridad y pueden sobrescribir otros estilos CSS.*(Recomendable para pruebas)*
3. **Dinamismo con JavaScript:** Útil para cambiar estilos directamente desde JavaScript, por ejemplo, para efectos interactivos o animaciones.
4. **Rendimiento en Elementos Críticos:** Algunas veces se usa para estilos críticos que deben cargarse inmediatamente. (Una modal de loading)

releevant.

Casos de uso de archivo externo .css

1. **Mantenibilidad y organización:** Facilita el mantenimiento y la organización de los estilos, especialmente en proyectos grandes.
2. **Reusabilidad:** Los estilos definidos en un archivo externo pueden ser reutilizados en varias páginas.
3. **Rendimiento y caché:** Los navegadores pueden cachear archivos CSS externos, lo que mejora los tiempos de carga en visitas sucesivas.
4. **Separación de preocupaciones:** Mantiene una clara separación entre la estructura del documento (HTML), la presentación (CSS) y el comportamiento (JavaScript).
5. **Colaboración y escalabilidad:** Facilita la colaboración en equipos grandes y hace que el código sea más escalable.

releevant.

💻 Vamos a probar

releevant.

Anatomía de una regla

Selector del elemento que vamos a estilizar y modificar

Abrimos brackets

Declaración que se compone de: propiedad y valor

Propiedad que vamos a modificar.

Separamos propiedad de valor con “:”

Asignamos el valor que damos a la propiedad.

Cerramos declaración con “;”

Cerramos brackets

releevant.

Anatomía de una regla

```
p {
    color: red;
    font-size: 12px;
    width: 200px;
    height: 200px;
}
```

Las declaraciones se apilan una debajo de otra.

releevant.

Selectores básicos

Un **selector en CSS** es una forma de **identificar** a qué elementos de la página HTML se les aplicarán ciertas reglas de estilo. Los selectores **determinan el alcance de las reglas** de CSS, permitiendo a los desarrolladores especificar qué elementos deben ser estilizados y cómo.

¿Cómo vamos a seleccionar nuestros elementos html para darles formato?

1. Selector **Tipo**
2. Selector **clase**
3. Selector **Id**

releevant.

Selectores básicos

SELECTOR TIPO

```
/*  Selector tipo */
p{
    color: red;
}
```

- La usamos cuando queremos **dar estilo a todas las etiquetas** de un **documento** o un **elemento padre**.
- Son **menos específicos que los ID o las clases**.
- Ideales para **estilos generales** que se aplican a través de todo el sitio.
- **Ayuda a mantener la consistencia** en los elementos HTML estándar.
- **Útil para establecer estilos base** o predeterminados. **CSS Reset**

Por ejemplo: quiero seleccionar todos los párrafos o listas de un documento para darles el formato “p”

releevant.

Selectores básicos

SELECTOR ID

```
/*  Selector ID */
#texto{
    color: blue;
}
```

- **Nunca lo usaremos para dar estilos**. Normalmente se deja para utilizarlo con JS.
- En **ocasiones muy especiales si es muy preciso**.
- Debe ser un **ID único por cada elemento**. No puede repetirse.
- Tienen una alta **especificidad**, lo que significa que sobreescribirá los estilos de clases y etiquetas en caso de conflicto.
- **Se suele utilizar para anclajes de navegación interna**.

releevant.

Selectores básicos

SELECTOR CLASE

```
/*  Selector clase */
.contenido{
    color: green;
}
```

- Para **dar estilos a etiquetas o a varios elementos** de la página pero no necesariamente a todos los elementos de un tipo.
- Proporcionan equilibrio entre **especificidad** y **reusabilidad**.
- Ideal para **componentes** o **elementos de diseño** que se repiten como botones o tarjetas de producto…
- Permiten mayor **modularidad** y **mantenimiento** del código con una correcta nomenclatura.

releevant.

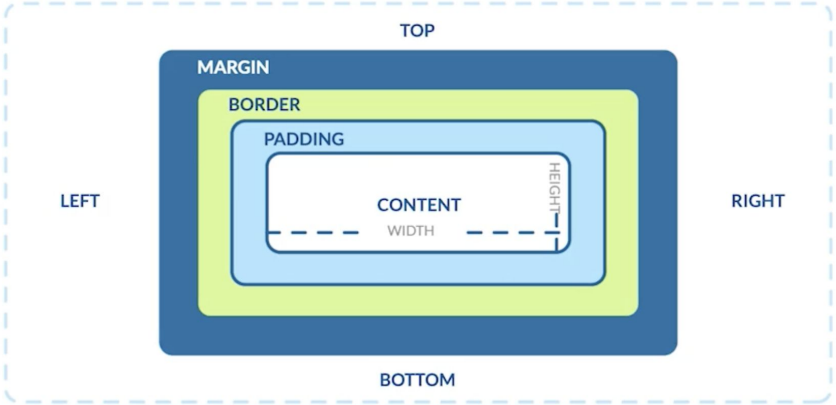
💻 Vamos a probar

releevant.

Concepto Box Model

El Modelo de Caja **(Box Model)** en CSS es un concepto fundamental que describe cómo se estructuran y se visualizan los elementos HTML.

Cada elemento se representa como una caja, y el Box Model define cómo se calcula el tamaño total de esa caja y cómo interactúa con otras cajas (elementos) **según su display** en la página.

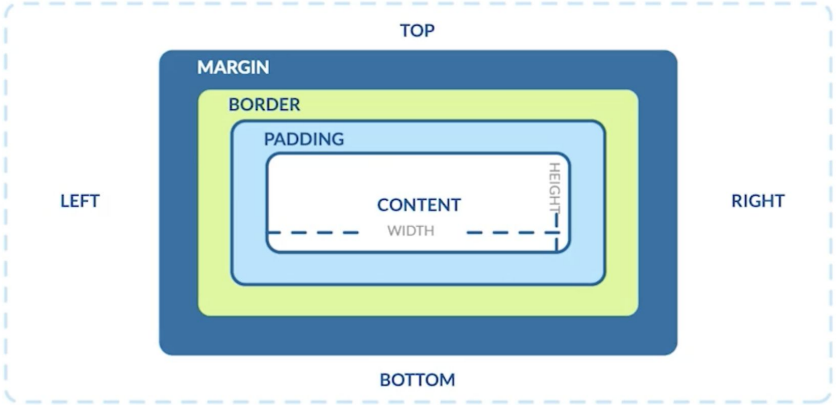


releevant.

Concepto Box Model

ÁREAS BOX MODEL

1. Content Box (Contenido)
2. Padding Box (Relleno)
3. Border Box (Borde)
4. Margin Box (Margen)



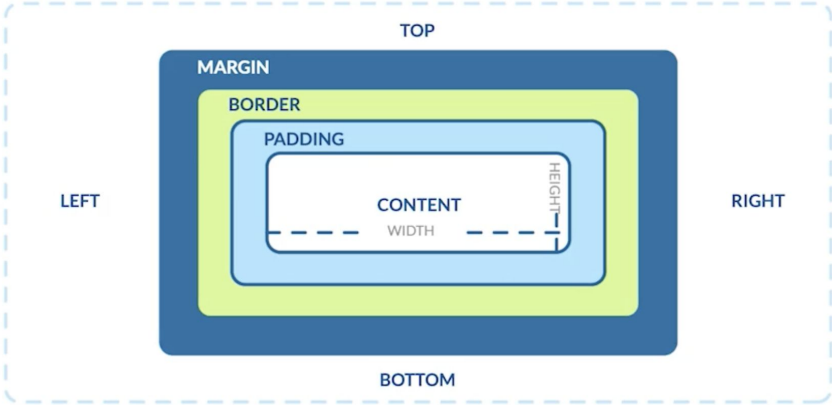
releevant.

Concepto Box Model

CONTENT BOX

Es la **parte central donde se muestra el contenido** del elemento, como texto o imágenes.

El **tamaño del área de contenido** se puede definir con las propiedades **width** (ancho) y **height** (alto).



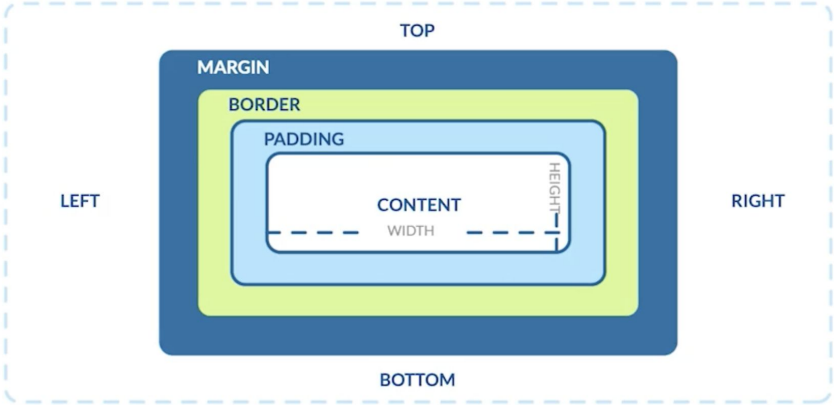
releevant.

Concepto Box Model

PADDING BOX

Es el **espacio entre el contenido y el borde del elemento**.

El **padding aumenta el tamaño** total del elemento **pero es “transparente”**; es decir, **muestra el fondo del elemento**.



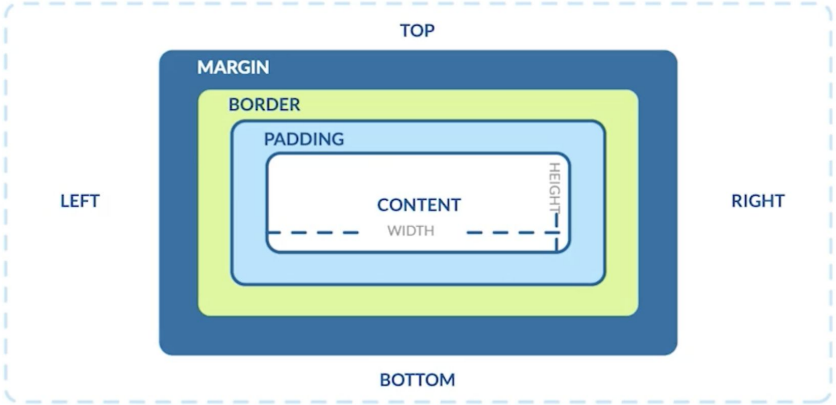
releevant.

Concepto Box Model

BORDER BOX

Rodea el padding y el contenido.

Puede tener **diferentes estilos**, **colores** y **anchos**.



releevant.

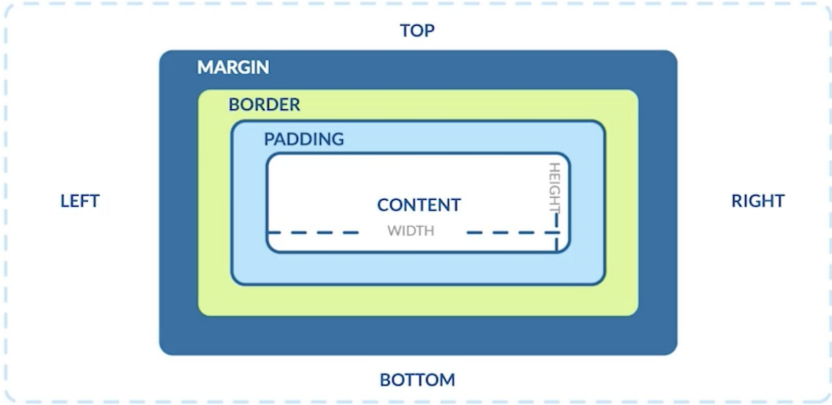
Concepto Box Model

MARGIN BOX

Es el **espacio exterior** al borde.

El margen **separa el elemento de otros** elementos en la página.

A diferencia del padding, el margen no muestra el fondo del elemento y es transparente.



releevant.

💻 Vamos a probar

releevant.

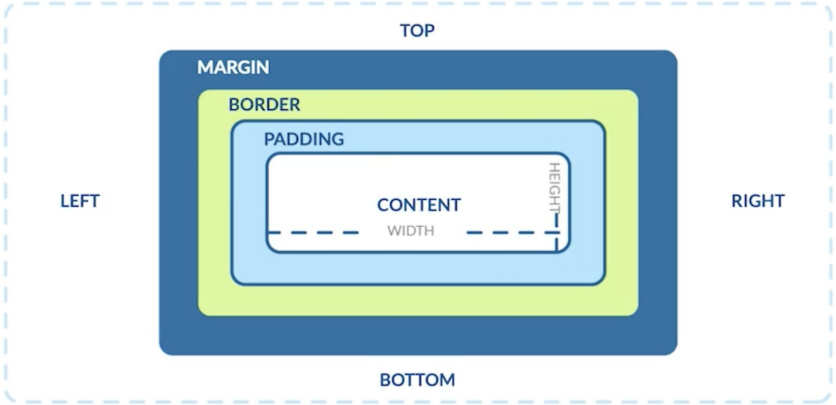
Propiedades básicas de la caja

MARGIN Y PADDING

Como hemos visto en el gráfico del concepto box model las cajas poseen un eje de 4 coordenadas: **top**, **right**, **bottom** y **left**.

En este orden.
En el sentido de las agujas del reloj.

Los valores del **padding** y el **margin** los podemos establecer de varias formas.



releevant.

Propiedades básicas de la caja

MARGIN Y PADDING

Para asignar estilos al **margin** o **padding** podemos aplicar:

- Un **único valor a todos los ejes** o lados.
- Un valor al **top y el bottom**, y otro al **right y al left,**
- Un valor **diferente a cada eje.**

```css
/* Asignamos un margin y padding de 10px a todos sus lados */
margin: 10px;
padding: 10px;

/* Asignamos un margin de 10px al top y al bottom */
/* Asignamos un margin de 32px al right y al left */
margin: 10px 32px;
/* Asignamos un padding de 10px al top y al bottom */
/* Asignamos un padding de 32px al right y al left */
padding: 10px 32px;

/* Asignamos un margin de 10px al top */
/* Asignamos un margin de 32px al left */
/* Asignamos un margin de 15px al bottom */
/* Asignamos un margin de 42px al left */
margin: 10px 32px 15px 42px;
/* Asignamos un padding de 10px al top */
/* Asignamos un padding de 32px al left */
/* Asignamos un padding de 15px al bottom */
/* Asignamos un padding de 42px al left */
padding: 10px 32px 15px 42px;
```

releevant.

Propiedades básicas de la caja

MARGIN Y PADDING

También podemos asignar sólo a un eje o lado con las propiedades:

- **padding-top**
- **padding-right**
- **padding-bottom**
- **padding-left**
- **margin-top**
- **margin-right**
- **margin-bottom**
- **margin-left**

```
/* Sólo asignamos el valor al margin o padding top */
margin-top: 10px;
padding-top: 10px;

/* Sólo asignamos el valor al margin o padding right */
margin-right: 10px;
padding-right: 10px;

/* Sólo asignamos el valor al margin o padding bottom */
margin-bottom: 10px;
padding-bottom: 10px;

/* Sólo asignamos el valor al margin o padding left */
margin-left: 10px;
padding-left: 10px;
```

releevant.

Propiedades básicas de la caja

BORDER

Para declarar la **propiedad border** debemos asignar **3 valores**:

1. Ancho
2. Estilo
3. Color

En caso de necesitar aplicar el estilo sólo a un eje, también podemos hacerlo llamando a la propiedad **border-[eje/lado]**:

```
/* Valores [ancho] [estilo] [color] */
border: 5px solid black;

/* Propiedades border por separado */
border-top: 5px solid black;
border-right: 5px solid red;
border-bottom: 5px solid blue;
border-right: 5px solid green;
```

releevant.

Propiedades básicas de la caja

BORDER-RADIUS

El **border radius** es una propiedad muy utilizada y la podemos aplicar a casi cualquier elemento para redondear sus vértices.

En este caso los ejes o **vértices serían**:

- **top-left**
- **top-right**
- **bottom-left**
- **bottom-right**

```
/* Aplicamos border radius a todos los vértices */
border-radius: 10px;

/* Aplicamos border radius de 10px a top-left & botom-right */
/* Aplicamos border radius de 20px a top-right & botom-left */
border-radius: 10px 20px;

/* Aplicamos border radius
10px a top-left
20px a top right
 5px a bottom right
15px a bottom left */
border-radius: 10px 20px 5px 15px;

/* Aplicamos border radius a cada vértice de forma independiente */
border-top-left-radius: 30px;
border-top-right-radius: 30px;
border-bottom-right-radius: 30px;
border-bottom-left-radius: 30px;
```

releevant.

Propiedades básicas de la caja

BORDER-RADIUS

También podemos asignar la propiedad a un único vértice declarando:

- **border-top-left-radius**
- **border-top-right-radius**
- **border-bottom-right-radius**
- **border-bottom-left-radius**

```
/* Aplicamos border radius a todos los vértices */
border-radius: 10px;

/* Aplicamos border radius de 10px a top-left & botom-right */
/* Aplicamos border radius de 20px a top-right & botom-left */
border-radius: 10px 20px;

/* Aplicamos border radius
10px a top-left
20px a top right
 5px a bottom right
15px a bottom left */
border-radius: 10px 20px 5px 15px;

/* Aplicamos border radius a cada vértice de forma independiente */
border-top-left-radius: 30px;
border-top-right-radius: 30px;
border-bottom-right-radius: 30px;
border-bottom-left-radius: 30px;
```

releevant.

Propiedades básicas de la caja

BACKGROUND-COLOR

Para darle color de fondo a nuestra caja o elemento html debemos declarar la propiedad **background-color** y asignarle un color.

**Nota: Más adelante veremos los códigos de color.*

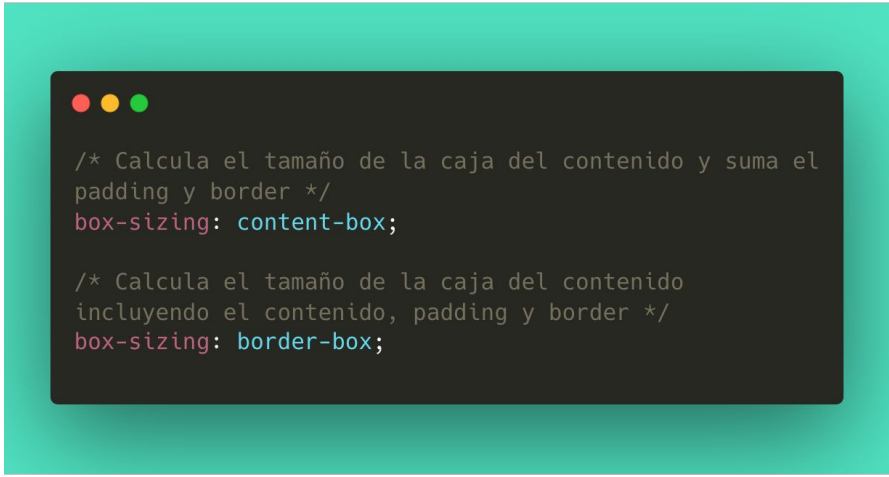
releevant.

Box sizing

En CSS, la propiedad **box-sizing** permite **controlar** cómo se **calcula el tamaño de las cajas**:

Content-box (valor predeterminado): El tamaño declarado con width y height se aplica solo al área de contenido. El **padding** y el **borde** se **añaden al tamaño total**.

Border-box: El tamaño declarado con **width** y **height incluye el contenido**, el **padding** y el **borde**. Esto hace que sea más fácil calcular y controlar el tamaño total de un elemento.



releevant.

Todo son cajas

En nuestros proyectos vamos a trabajar constantemente con este concepto.
TODO son cajas, y estas tienen ciertas propiedades y estilos que podemos customizar.

releevant.

¿Qué hemos aprendido hasta ahora?

1. Qué es **CSS**
2. Concepto de cascada
3. Formas de dar estilos a nuestro proyecto
4. Anatomía de una regla
5. Selectores básicos
6. Concepto Box Model
7. Margins, paddings, border & border-radius y background-color.
8. Propiedad box-sizing

releevant.

Deberes 🤓

releevant.

Deberes

- Crea una carpeta **ejercicioTema1**.
- Dentro de la carpeta crea un archivo **index.html** y genera el código HTML5 básico para un documento.
- Crea un archivo **styles.css** dentro de la carpeta **ejercicioTema1.**
- Añade la hoja de estilos en el **<head>** con la etiqueta **<link>**
- Declara un id **#cuerpo** y una clase **.home** a la etiqueta **body**.
- Crea un **<section></section>** en el **<body>** del **HTML** con la clase **.container** e id **#contenido**
- Declara **5** etiquetas ** ** dentro del **<section></section>** e inserta contenido **Lorem Ipsum** a cada uno.
- Asigna una clase con nombre **.texto** a cada etiqueta ** **
- Declara un id único para cada etiqueta ** ** ej: **#textoUno #textoDos** …
- Debajo del **<section>** Crea una lista ordenada con **5 ** que contengan un **lorem ipsum** cada uno de ellos.
- Asigna la clase **.listaOrdenada** al elemento **<ol>**
- Asigna la clase **.item** a cada elemento **<li>**
- En el archivo **styles.css** añade la regla **p{color:red;}**
- Abre una etiqueta **<style>** en el head del documento html y declara la regla **ol{color:blue;}**
- Asigna el atributo **style** y la declaración **color:blue;** en la etiqueta **<section>**

releevant.

Deberes

- Declara un **border** de **1px sólido de color negro** al elemento con clase **.container**.
- Declara un **padding** de **40px** al elemento con clase **.container**
- Declara un **color de fondo** con código **#f2f2f2** al elemento con clase **.container**.
- Declara un **border radius** de **10px** al elemento con clase **.container**
- Declara la propiedad **font-size: 21px;** a todos los elementos **<p>** con clase **.texto**.
- Aplica un color de **fondo diferente** a cada uno de los elementos **<p>** accediendo a ellos a **través de su ID**.
- Declara un **padding-bottom de 10px** a todos los elementos con **clase .item**
- Crea un **<button>** con clase **.btn** dentro del body.
- Dale formato de botón a tu gusto. Debes darle un **padding**, **color**, **border**, **border-radius** y **background-color**.