

CSS 3 – Segunda parte

2º DAW
Curso 2023/2024
Mari Cruz Gómez García

1 / 59

INDICE

- 1) Metodología BEM
- 2) Estilo en cascada
- 3) Herencia
- 4) Prefijos de navegador
- 5) Variables
- 6) Unidades de medida
 - Absolutas
 - Relativas
- 1) Funciones
- 2) Colores
- 3) Transparencia y filtros
- 4) Sombras
- 11) Google Fonts
- 12) Hypens
- 13) Desbordamiento
- 14) Despliegue de textos
- 15) Bordes redondeados/con imágenes
- 16) Outline
- 17) Normalize
- 18) Barra de scroll
- 19) Cursores
- 20) Display
- 21) Posicionamiento
- 22) Float

2 / 59

1. Metodología BEM

- La metodología BEM divide la interfaz de usuario en bloques independientes para crear componentes escalables y reutilizables.
- **BEM (Block, Element, Modifier** o Bloque, Elemento, Modificador) es una metodología ágil de desarrollo basada en componentes. Fue creada por la empresa Yandex para desarrollar sitios en poco tiempo y con un soporte de largo plazo.

3 / 59

Bloques

- Es un contenedor donde se encontrarán los diferentes elementos. Por ejemplo, un encabezado (header), una barra lateral (sidebar/aside) un área de contenido principal (main) y un pie de página (footer), se consideraría cada uno como un bloque.
- El bloque de elementos corresponde a la raíz de la clase y deberá ir siempre primero. Cuando ya está definido es posible nombrar los elementos que compondrán la estructura final y su contenido.

4 / 59

Bloques

HTML

```
11 <section>
12   <article class="noticia">
13     <!--Bloque contenedor-->
14   </article>
15 </section>
```

CSS

```
1 .noticia{
2   background: lightgray;
3 }
```

5 / 59

Elementos

- El elemento es una de las variadas piezas que compondrán la estructura de un bloque. El bloque es el todo y los elementos son las piezas de este bloque.
- De acuerdo a la metodología BEM, cada elemento se escribe después del bloque padre, usando dos guiones bajos:

6 / 59

Elementos

Al convertirse en el centro de los nombres de las clases, los elementos ayudan a entender cómo estructurar y manejar las hojas de estilos una vez que se empieza a diseñar el proyecto. Por esto, es importante usar nombres claros y precisos.

HTML

```
11 <section>
12   <article class="noticia">
13     <h1 class="noticia__titulo">Título de la noticia</h1>
14   </article>
15 </section>
```

CSS

```
5 .noticia__titulo{
6   font-family: 'Times New Roman', serif;
7 }
```

7 / 59

Modificadores

- El ejercicio de nombrar las clases sirve para que los elementos puedan repetirse. Si los elementos son los mismos, no será necesario escribir nuevas clases en otras áreas del sitio.
- Para modificar el estilo de un elemento específico, existen los modificadores.
- Estos se usan agregando un doble guión justo después del elemento o bloque que se quiere modificar:

8 / 59

Modificadores

HTML

```
11 <section>
12   <article class="noticia--destacada">
13     <h1 class="noticia__titulo--uppercase">Título de la noticia</h1>
14   </article>
15 </section>
```

CSS

```
9  .noticia--destacada{
10    background: dimgray;
11  }
12  .noticia__titulo--uppercase{
13    text-transform: uppercase;
14  }
```

9 / 59

Ventajas

- La metodología BEM se encuentra en constante evolución, pero ha demostrado su eficiencia al establecer una buena jerarquía en el desarrollo de plataformas.
- Añade especificidad: Usa un selector único para cada regla, lo que permite reducirla y hacer menos repeticiones.
- Aumenta la independencia: Los bloques se pueden mover a cualquier parte del documento, sin afectar los estilos.
- Mejora la herencia múltiple: Se puede cambiar cualquiera de las tres partidas sin afectar a las demás.

10 / 59

Ventajas

- Permite la reutilización: Es posible reciclar ciertas áreas de código desde un proyecto hacia otro, esto debido a la no existencia de dependencias mayores en cuanto a la implementación de cada uno de los bloques estructurados.
- Entrega simplicidad: Permite un fácil entendimiento una vez conocido el proceso lógico sobre el cual se basa. A su vez, las convenciones a la hora de nombrar las clases permiten tener un control absoluto al saber a qué, quién y hacia dónde hacemos referencia dentro de una estructura.

11 / 59

Desventajas

- Las convenciones pueden ser muy largas.
- A algunas personas les puede tomar tiempo aprender la metodología.
- El sistema de organización puede ser difícil de implementar en proyectos pequeños.

12 / 59

Aplicaciones

- Cabe destacar que es posible aplicar esta metodología en diferentes lenguajes y contextos. Por ejemplo, destacando la estructuración en HTML5, en CSS y sus preprocesadores (como SASS o LESS) y en programación con JavaScript, considerando su utilización en frameworks como AngularJS o ReactJS.
- Además, se puede usar en la organización a nivel de archivos y cualquier otro sistema en el que sea necesario mantener un orden modular.
- Sitio oficial <https://getbem.com/>
- Ejemplos
<https://victorroblesweb.es/2022/08/03/que-es-bem-y-como-utilizarlo-en-css/>

13 / 59

2. Estilos en cascada

- ¿De qué color es el párrafo en cada caso?

```
<style>
  p {
    color: blue;
  }
  p {
    color: red;
  }
</style>
</head>
<body>

  <p class="miparrafo">Esto es un párrafo </p>
</body>
```

```
<style>
  p.miparrafo{
    color: blue;
  }
  p {
    color: red;
  }
</style>
</head>
<body>

  <p class="miparrafo">Esto es un párrafo </p>
</body>
```

```
<style>
  p{
    color: blue !important;
  }
  p {
    color: red;
  }
</style>
</head>
<body>

  <p class="miparrafo">Esto es un párrafo </p>
```

3. Herencia

- Inherit → Hereda el valor que tiene la misma propiedad CSS en su elemento padre
- Initial → Establece el valor inicial que tenía la propiedad CSS inicialmente
- Ver ejemplo inheritInitial.html

15 / 59

4. Prefijos de navegador

- CSS-3 está en continuo cambio y cuando aparecen propiedades nuevas, no siempre las soportan todos los navegadores.
- Los prefijos de navegador se ponen para asegurarnos que funcionan en los distintos navegadores.

```
<style>
div {
  transform: rotate(25deg); /* Navegadores que implementan especificación oficial */
  -webkit-transform: rotate(25deg); /* Versiones antiguas de Chrome (Motor WebKit) */
  -moz-transform: rotate(25deg); /* Versiones antiguas de Firefox (Motor Gecko) */
  -ms-transform: rotate(25deg); /* Versiones antiguas de IE (Motor Trident) */
  -o-transform: rotate(25deg); /* Versiones antiguas de Opera (Motor Presto) */
}
```

16 / 59

5. Variables en CSS

- si queremos que una variable sea global (que sea accesible desde cualquier lugar del proyecto) la declaramos en la pseudoclase :root que apunta al elemento más alto del DOM <html></html> pero tiene más especificidad que el selector de etiqueta.

```
1  /* Las variables deben declararse dentro de un selector
2     Se acostumbra usar :root para que la variable sea global
3     Se declaran con dos guiones y el nombre
4     Para utilizarlas se usa la funcion var() y entre
5     los paréntesis se escribe el nombre de la variable
6
7  */
8
9  :root {
10     --my-color: red
11 }
12
13 body {
14     color: var(--my-color)
15 }
16
```

17 / 59

Cascada y herencia en las variables

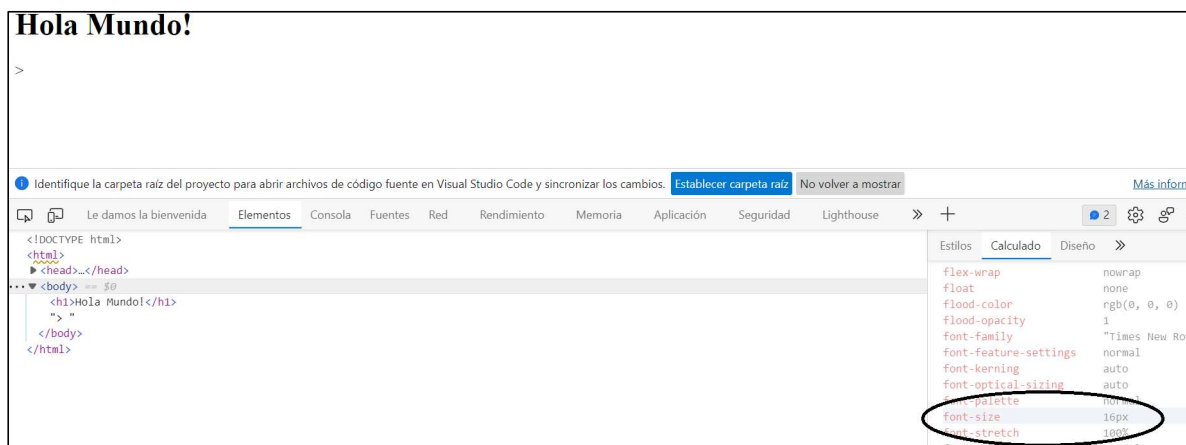
```
1  :root {
2     --my-color: red
3  }
4
5  body {
6     color: var(--my-color)
7  }
8
9  /* En este caso todos los textos (párrafos, títulos, etc.)
10     serán de color rojo por herencia, porque descienden de
11     body al resto de elementos (comportamiento propio de CSS)
12  */
13
```

```
1  :root {
2     color: red
3  }
4
5  body {
6     color: var(--color)
7  }
8
9  h1 {
10     --color: blue;
11 }
12
13 /* En este caso todos los textos (párrafos, títulos, etc.)
14     serán de color rojo por herencia, pero todos los h1 serán
15     azules porque hemos usado la cascada para redefinir el valor
16     de la propiedad personalizada --color
17  */
18
```

18 / 59

6. Unidades de medida

- En Dev Tools podemos ver en el apartado Calculado, el tamaño por defecto del navegador, así como otras propiedades.



19 / 59

Unidades de medida: FIJAS (absolutas)

No son relativas a nada más y en general se considera que siempre tienen el mismo tamaño.

- La mayoría de estos valores son más útiles cuando se usan en una salida en **formato impreso** que en la salida de pantalla. Por ejemplo, normalmente no usamos cm (centímetros) en pantalla. El único valor que usarás de forma frecuente es **px** (píxeles).

Unidad	Nombre	Equivale a
cm	Centímetros	1cm = 96px/2,54
mm	Milímetros	1mm = 1/10 de 1cm
Q	Cuartos de milímetros	1Q = 1/40 de 1cm
in	Pulgadas	1in = 2,54cm = 96px
pc	Picas	1pc = 1/6 de 1in
pt	Puntos	1pt = 1/72 de 1in
px	Píxeles	1px = 1/96 de 1in

20 / 59

Unidades de medida: RELATIVAS (flexibles)

- Ver ejemplo unidadesRelativas.html con un y font-size 2rem, ¿Qué tamaño píxeles aplicará?

Unidad	Relativa a
em	Tamaño de letra del elemento padre, en el caso de propiedades tipográficas como <code>font-size</code> , y tamaño de la fuente del propio elemento en el caso de otras propiedades, como <code>width</code> .
ex	Altura x de la fuente del elemento.
ch	La medida de avance (ancho) del glifo "0" de la letra del elemento.
rem	Tamaño de la letra del elemento raíz.
lh	Altura de la línea del elemento.
vw	1% del ancho de la ventana gráfica.
vh	1% de la altura de la ventana gráfica.
vmin	1% de la dimensión más pequeña de la ventana gráfica.
vmax	1% de la dimensión más grande de la ventana gráfica.

21 / 59

Unidades relativas

- ¿Qué pasa si aplicamos...? Compruébalo en Dev Tools

```
<style>

:root{
  font-size: 50px;
}

h1{
  font-size: 2rem;
}
```

- La pseudo-clase :root de CSS selecciona el elemento raíz de un árbol que representa el documento.
- En HTML, :root representa el elemento <html> (es el abuelo de cada contenedor del documento)

22 / 59

Unidades relativas

- ¿Cuál es la diferencia entre rem y em?
- Compruébalo poniendo un tamaño de texto en root y body y luego aplica al h1 rem y em y determina de dónde coge el tamaño en cada caso.
- vh y vw son relativas al viewport (tamaño ventana gráfica) y son números entre 1 y 100.
- ¿Cuál es la diferencia entonces entre porcentajes y viewports?

23 / 59

Porcentajes vs Viewports

La diferencia entre las unidades porcentuales expresadas con "%" (width:100%) y las viewport-units (width:100vw) es que las primeras heredan los valores de su elemento padre, y las segundas siempre heredan los valores del bloque inicial o ventana gráfica.

```
7      <style>
8
9      body {
10         width: 500px;
11         height: 100px;
12     }
13
14     div{
15         width: 50vw;
16         height: 50vh;
17         background-color: palevioletred;
18     }
19
20
21     </style>
22 </head>
23 <body>
24
25     <div>Hola Mundo!</div>
26 </body>
```

```
7      <style>
8
9      body {
10         width: 500px;
11         height: 100px;
12     }
13
14     div{
15         width: 50%;
16         height: 50%;
17         background-color: palevioletred;
18     }
19
20
21     </style>
22 </head>
23 <body>
24
25     <div>Hola Mundo!</div>
26 </body>
```

24 / 59

Porcentajes

- El porcentaje también es una unidad de medida relativa, aunque por su importancia CSS la trata de forma separada a em, ex y px.
- El porcentaje es relativo a su elemento padre.

25 / 59

Unidades recomendadas

- En general, se recomienda el uso de unidades relativas siempre que sea posible, ya que mejora la accesibilidad de la página y permite que los documentos se adapten fácilmente a cualquier medio y dispositivo.
- El organismo W3C, recomienda el uso de la unidad em para indicar el tamaño del texto y para todas las medidas que sean posibles.
- Normalmente se utilizan píxel y porcentajes para definir el layout del documento (básicamente, la anchura de las columnas y de los elementos de las páginas) y em y porcentajes para el tamaño de letra de los textos.

26 / 59

7.Funciones

- Mira el ejemplo funciones.html.
- Fíjate que al añadir un menú arriba, se hace scroll y tendríamos que ir calculando el height del contenedor para ver el tamaño.
- En vez de eso, podemos usar la función `calc()` de forma que calcule automáticamente el espacio: `calc(100vh - 100px)`

```
8
9      /*Con esto le quitas el espacio que deja por defecto*/
10     * {
11         margin: 0;
12         padding: 0;
13     }
14     .menu {
15         width: 100%;
16         height: 100px;
17         background-color: black;
18     }
19
20     div.contenedor{
21         width: 100%;
22         height: 100vh;
23         background-color: palevioletred;
24     }
25
26
27     </style>
28 </head>
29 <body>
30     <div class="menu"></div>
31     <div class="contenedor"></div>
```

27 / 59

Funciones

- Prueba otras funciones. Por ejemplo establece para el contenedor:
- Width: `min(200px, 25%)`
- Y ahora prueba a redimensionar la página, ¿Qué sucede?

28 / 59

8. Colores

- Formas de aplicar colores:
- https://www.w3schools.com/css/css_colors_hsl.asp
 - El nombre en inglés del color
 - En hexadecimal
 - Rgb o rgba (el cuarto parámetro es un valor entre 0 o 1 que especifica la transparencia)
 - Hsl o hsla (color, saturación y brillo)
 - Linear-gradient o radial-gradient
- <https://developer.mozilla.org/en-US/docs/Web/CSS/gradient/linear-gradient>

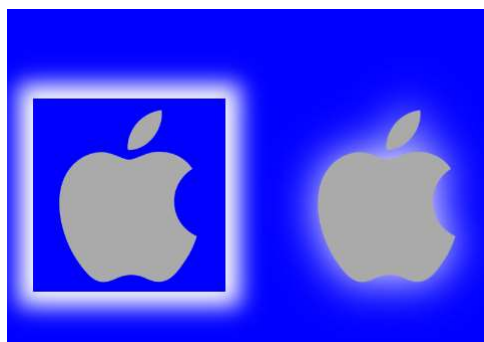
29 / 59

9. Transparencia y filtros

- ¿Cuál es la diferencia de aplicar transparencia mediante rgba o hsla y la propiedad *opacity*?
- Prueba los distintos filtros (propiedad *filter*) que se permite aplicar a una imagen. Ver ejemplo filtros.html
- Propiedad backdrop-filter (ver ejemplo backdropFilter.html).
- ¿Cuál es la diferencia entre filter y backdrop-filter?
- Filter aplica el filtro a todo el contenedor en general y backdrop-filter al fondo

10. Sombras

- Propiedad **box-shadow** para contenedores y **text-shadow** para textos.
- Ver ejemplo sombras.html
- <https://developer.mozilla.org/es/docs/Web/CSS/box-shadow>
- Ejercicio: prueba la diferencia entre aplicar a un logo un box-shadow y a otro logo un filtro drop-shadow.



31 / 59

Ejercicio sombras

- Crea una página que contenga el texto “Abierto 24 horas” y que al pasar el ratón por encima se vea así:



- PISTA: sombras encadenadas

32 / 59

11. Google Fonts

- 1) Ve a Google Fonts y descarga la fuente “Train One”.
- Descárgala en tu proyecto en la carpeta *Fonts*.
- Aplícala a un h1 de tu página.
- 2) También puedes ir a la fuente que te guste de Google Fonts y copiar el link e incluirlo en tu proyecto.
- 3) O instalar la extensión e incluir la fuente que quieras con `ctrl+shift+p` escribes Google Fonts y el nombre de la fuente.

```
@font-face {  
  font-family: 'Train One';  
  src: url('../Fonts/TrainOne-Regular.ttf');  
}  
  
h1 {  
  font-family: 'Train One';  
}  
  
</style>  
</head>  
<body>  
  <h1>Esto es un título</h1>
```

33 / 59

12. Hyphens

- Hyphens significa quiebre de palabra.
- <https://developer.mozilla.org/es/docs/Web/CSS/hyphens>

- 

34 / 59

13. Desbordamiento

- Vamos a imaginar que creamos una caja de 250x250px con texto dentro pero el texto no cabe en la caja y se desborda.
- Podemos tratar el comportamiento con la propiedad overflow.
- Prueba los distintos valores que puede tomar.

35 / 59

Desbordamiento de textos

- Imagina que tienes un encabezado de 300px de ancho y el texto que ocupa es muy largo.
- ¿Cómo hacemos para colocar puntos suspensivos y que el usuario pueda hacer clic si quiere para ver el texto completo?
 - **White-space: nowrap;** //Para colocarlo todo en una línea y que no quiebre el texto
 - **Overflow: hidden;** //Oculta el texto en cuanto llegue al corte de los 300 píxeles
 - **Text-overflow: ellipsis;** //Indica el tipo de desbordamiento y coloca los ...

36 / 59

14. Despliegue de textos

- Propiedad **column-count** para especificar que el texto se divida en columnas.
- Con **column-gap** podemos especificar la separación entre columnas.
- **Column-rule** permite indicar una línea divisoria entre columnas.

37 / 59

15. Bordes redondeados

- Border-radius
- https://www.w3schools.com/CSSref/CSS3_pr_border-radius.php

```
/* Border redondeados */  
border-radius: 10px;  
  
/* En direccion X. Primer valor Arriba izquierda y abajo derecha. Segundo valor arriba derecha y abajo izquierda */  
border-radius: 20px 50px;  
  
/* 1er valor: Arriba izquierda. 2do valor: arriba derecha + abajo izquierda y 3er valor: abajo derecha */  
border-radius: 20px 50px 80px;  
  
/* Sentido agujas del reloj */  
border-radius: 20px 50px 80px 40px;
```

38 / 59

Bordes con imágenes

- Crea un div con un alto y ancho y borde.
- Aplica ahora un borde con una imagen trabajando estas 4 propiedades y ve cambiando sus valores en el inspector del documento para que veas el resultado:
 - Border-image-width
 - Border-image-source
 - Border-image-slice
 - Border-image-repeat

39 / 59

16. Propiedad outline

- Borde o contorno que pone en el navegador en los inputs cuando situamos el cursor dentro de él:



- Propiedad outline (valores de borde como 1px solid blue o bien none si queremos que no lo aplique)

40 / 59

17. Normalize

- Para tener un control total de nuestro espacio, siempre incluimos la regla
* { margin:0; padding:0; box-sizing:border-box; }
- Si queremos proyectos más profesionales:
- <https://necolas.github.io/normalize.css/>
- Copiar código y guardarlo como normalize.css y enlazarlo a todos nuestros proyectos.

41 / 59

18. Personalizar barra de scroll

- Podemos personalizar la barra de scroll de la página o de los contenedores.
- Tan solo tenemos que usar los pseudoelementos **::-webkit-scrollbar** y **::-webkit-scrollbar-thumb**.
- Ver ejemplo scrollbar.html
- También se puede personalizar su comportamiento con la propiedad **scroll-behavior: smooth;**
- Ver ejemplo scrollbehavior.html

42 / 59

19. Cursores

- La propiedad **cursor** especifica el tipo de cursor que se mostrara cuando este se encuentre sobre un elemento.
- <https://developer.mozilla.org/es/docs/Web/CSS/cursor>
- Crea una caja y prueba los distintos valores de cursor.

43 / 59

20. Display

- Ver ejemplo display.html
- **display: block;** Por defecto - Se aplica en vertical
- **display: inline;** En linea - Horizontal ignora width y height
- **display: inline-block;** En linea - Horizontal NO ignora width y height
- **display: list-item;** Actúa como lista
- **display: table-cell;** Actúa como la celda de una tabla. Es el comportamiento de etiquetas como <th> o <td>.
- **display: table-row;** Actúa como la fila de una tabla. Es el comportamiento de etiquetas como <tr>.
- **display: none; visibility: hidden;** Ocultar

44 / 59

21. Posicionamiento

- **position: static;** Posicionamiento estático. Utiliza el orden natural de los elementos HTML.
- **position: relative;** Posicionamiento relativo. Los elementos se mueven ligeramente en base a su posición estática
 - top: 100px;
 - left: 200px;
- Ver ejemplo `static_position.html`

45 / 59

Posicionamiento absoluto

- El elemento se coloca en base al contenedor padre.
- Para ello, es IMPRESCINDIBLE que el padre tenga un posicionamiento RELATIVE.
- Ver ejemplo `absolute.html`

46 / 59

Posicionamiento fixed

- Le indicamos un lugar al contenedor y siempre va a estar fijo, aunque hagamos scroll.
- Es decir, siempre va a ser relativo al viewport.
- Ver ejemplo [fixed.html](#)

47 / 59

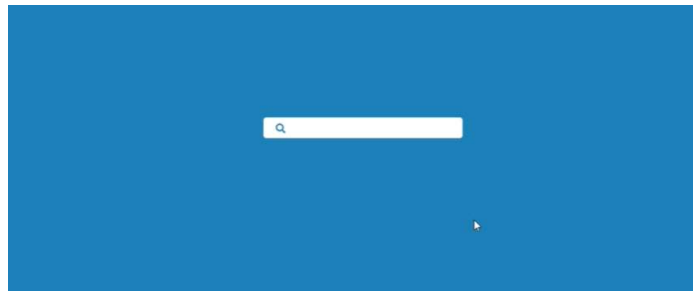
Posicionamiento sticky

- Es como el fixed pero dentro del contenedor padre. Si luego abajo hay otro tipo de contenido, se para.
- El contenedor padre debe tener un ancho o un alto predeterminado o contenido para que pueda hacer scroll y ver cómo se queda “pegado” en su posición.
- Ver ejemplo [sticky.html](#)

48 / 59

Práctica: Input con icono

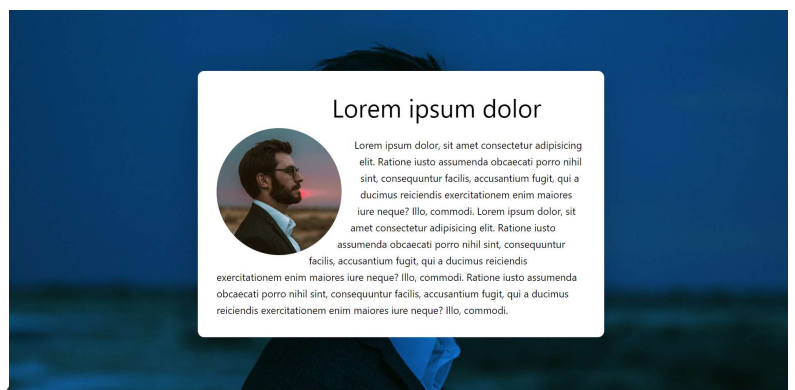
- Ir a fontawesome.com, descarga carpeta fonts (<https://fontawesome.com/download> Free for Web), descomprime en tu proyecto y agregas en tu archivola hoja de estilo all.css.
- Busca el icono de buscar <https://fontawesome.com/icons>



49 / 59

22. Float

- Ver ejemplo practica_float.html
- El body es una imagen de fondo.
- Ponemos contenedor blanco con bordes redondeados.
- Dentro del contenedor: imagen, título y párrafo.
- La imagen está también en un contenedor con la imagen de fondo a la que le hemos dado forma redondeada.



50 / 59

23. Transiciones

- Para pasar de un estado a otro de forma menos abrupta. Ejemplo, dos cajas de un color y al pasar el ratón por encima cambia de color.
- Propiedades de transición:

transition-property: background; (a quién le va a realizar el cambio)

transition-duration: 1s;

transition-timing-function: linear; (movimiento o velocidad)

transition-delay: 1s; (retraso)

- `/* Atajo */`

transition: all 1s linear 0.2s;

51 / 59

Transiciones

- Ver ejemplo transiciones.html
- https://www.w3schools.com/css/css3_transitions.asp
- Ver ejemplo timing-function.html
- https://www.w3schools.com/cssref/css3_pr_transition-timing-function.php

52 / 59

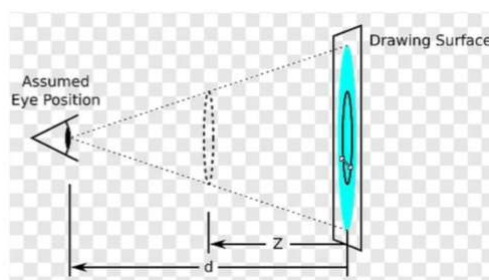
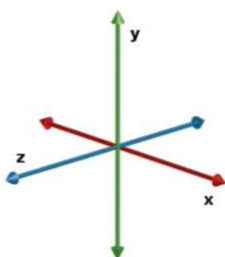
24. Transformaciones

- Consiste en mover elementos en el eje X y/o el eje Y, escalar objetos, rotarlos o deformarlos. Ver ejemplo [transform.html](#)
 - **transform: translate(ejeX, ejeY);**
 - **transform: scale(1.5);**
 - **transform: rotate(150deg);**
 - **transform: skew(50deg);**Podemos indicar el origen de la transformación:
 - **transform-origin: left;** Ver ejemplo [transformorigin.html](#)

53 / 59

Transformaciones 3D

- Necesitamos aplicar perspectiva para que se pueda apreciar el efecto tridimensional. Para ello, hay que aplicárselo al padre: con la propiedad **perspective** ponemos un tamaño que va a significar el espacio que se aleja el hijo.
- Con perspective-origin podemos indicar el foco de visión.



54 / 59

Transformaciones 3D

- https://www.w3schools.com/css/css3_3dtransforms.asp
- <https://developer.mozilla.org/en-US/docs/Web/CSS/transform>

55 / 59

25. Clip-path

- Permite crear figuras.
- <https://bennettfeely.com/clippy/>
- Te permite crear la figura y copias y pegas en tu proyecto.

56 / 59

26. Animaciones

- https://developer.mozilla.org/es/docs/Web/CSS/CSS_Animations/Using_CSS_animations
- https://www.w3schools.com/css/css3_animations.asp
- **PASO 1 - PROPIEDADES DE ANIMACION**

- animation-name -- Nombre a la animacion
- animation-duration -- Duracion
- animation-timing-function -- Movimiento
- animation-delay -- Retraso
- animation-iteration-count -- Cantidad de veces que se repite
- animation-direction -- Direccion
- animation-fill-mode -- Que mostrara cuando termine
- animation-play-state -- Estado de reproduccion

57 / 59

Animaciones

- Atajo de animacion

animation: changeColor 5s linear infinite alternate running;

PASO 2 - CREAR LA ANIMACION

```
@keyframes changeColor {  
  0%{  
    transform: translateX(0);  
    background: blue;  
  }  
  100%{  
    transform: translateX(1000px);  
    background: #000;  
  }  
}
```

58 / 59

Animaciones

- Crea el efecto de carga:

