TEMAS:

- ➔ HTML + CSS
  - selectores
  - .css files
  - Box model
  - Colores, Tipografía
  - CSS Grids
  - Diseño adaptativo.

OBJETIVO:

MARCO TEÓRICO

Lectura 1: CSS

MDN Web
Web: https://developer.mozilla.org/

SELECTORES

Selector Universal

El selector universal (*) es el comodín. Nos permite seleccionar todos los elementos de una página, se suele usar en combinación con otros selectores (ver Combinators más adelante).

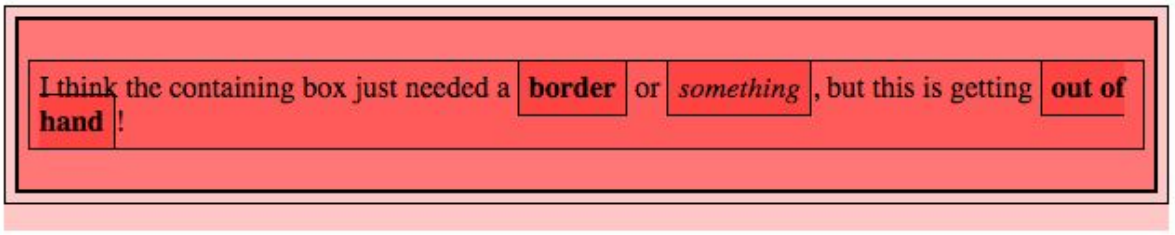
Ahora como ejemplo; dado un código HTML:

```
<div>
  <p>I think the containing box just needed
  a <strong>border</strong> or <em>something</em>,
  but this is getting <strong>out of hand</strong>!</p>
</div>
```

Y su correspondiente hoja de estilos:

```
* {
  padding: 5px;
  border: 1px solid black;
  background: rgba(255,0,0,0.25)
}
```

Juntos producirán el siguiente resultado:



Combinaciones

En CSS, podemos combinar varios selectores juntos y con ello seleccionar elementos contenidos en otros elementos, o elementos adyacentes a otros. Disponemos de cuatro tipos:

- El selector descendiente — (espacio) — permite seleccionar un elemento anidado en alguna parte dentro de otro elemento (no tiene por qué ser un hijo; puede ser un nieto, por ejemplo)
- El selector hijo — > — permite seleccionar un elemento que es hijo directo de otro elemento.
- El selector hermano — + — permite seleccionar un elemento que es hermano directo de otro elemento (a la derecha por ejemplo, en el mismo nivel jerárquico).
- El selector hermano en general — ~ — permite seleccionar cualquier elemento hermano de otro (por ejemplo en el mismo nivel jerárquico, pero no necesariamente adyacente a él).

Veamos un rápido ejemplo para ver el funcionamiento:

```
<section>
  <h2>Heading 1</h2>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
  <div>
    <h2>Heading 2</h2>
    <p>Paragraph 3</p>
    <p>Paragraph 4</p>
  </div>
</section>
```

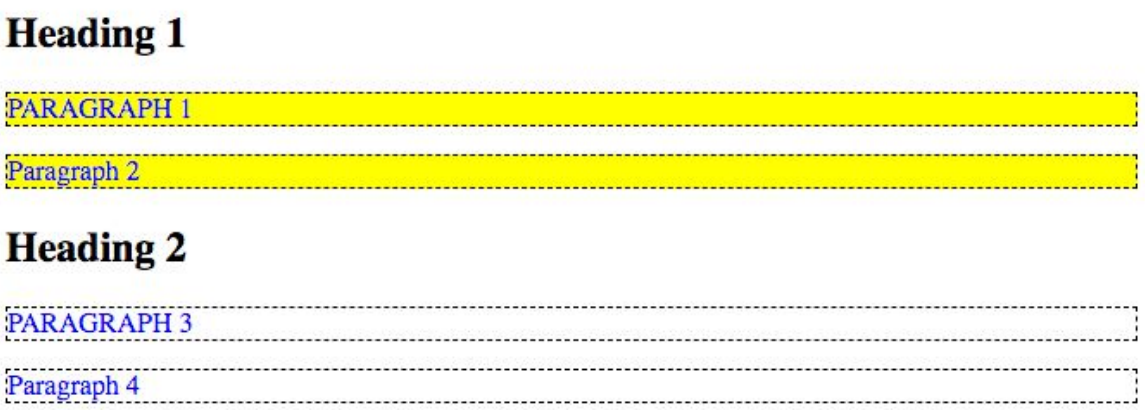
```
section p {
  color: blue;
}

section > p {
  background-color: yellow;
}

h2 + p {
  text-transform: uppercase;
}

h2 ~ p {
  border: 1px dashed black;
}
```

El resultado será:



Los selectores funcionarán como sigue:

- section p selecciona todos los elementos <p> — los dos primeros son hijos directos del elemento <section>, y los dos siguientes son nietos del elemento <section> (ya que también están dentro de). Todo el texto del párrafo será de color azul.
- section > p selecciona solo los dos primeros elementos <p>, que son hijos directos del elemento <section> (pero no los dos siguientes, pues no son hijos directos). Por tanto, solo los dos primeros párrafos tendrán realzado amarillo.

- h2 + p seleccione sólo los elementos <p> que están directamente después del elemento <h2> en el mismo nivel jerárquico — en este caso el primer y el tercer párrafo. Así estos tendrán el texto en mayúsculas
- h2 ~ p selecciona cualquier elemento <p> en el mismo nivel jerárquico desde los elementos <h2> — en este caso todos los párrafos. Todos ellos tendrán el borde punteado.

Selectores de presencia y valor

Estos selectores de atributos afectarán a los elementos cuyo valor coincida exactamente con el valor del atributo especificado:

- **[attr] :** Este selector 'seleccionará' todos los elementos que contengan el atributo attr, sin importar el valor que tenga.
- **[attr=val]** : Este, seleccionará los elementos con el atributo attr, pero solo aquello cuyo valor coincida con val.
- **[attr~=val]:** Este selector afectará a los elementos con el atributo attr, pero solo si el valor val está contenido en la lista de valores (separados por espacios) incluidos en el valor de attr, por ejemplo una de las clases contenida en una lista de clases (separadas por espacios).

Veamos como ejemplo el siguiente fragmento de código HTML:

Ingredients for my recipe: Poulet basquaise

```
<ul>
  <li data-quantity="1kg" data-vegetable>Tomatoes</li>
  <li data-quantity="3" data-vegetable>Onions</li>
  <li data-quantity="3" data-vegetable>Garlic</li>
  <li data-quantity="700g" data-vegetable="not spicy like chili">Red
pepper</li>
  <li data-quantity="2kg" data-meat>Chicken</li>
  <li data-quantity="optional 150g" data-meat>Bacon bits</li>
  <li data-quantity="optional 10ml" data-vegetable="liquid">Olive
oil</li>
  <li data-quantity="25cl" data-vegetable="liquid">White wine</li>
</ul>
```

Y un sencillo documento de estilos CSS:

```
/* All elements with the attribute "data-vegetable" are given green
text */
[data-vegetable] {
```

```
  color: green;
}

/* All elements with the attribute "data-vegetable"
   with the exact value "liquid" are given a golden
   background color */
[data-vegetable="liquid"] {
  background-color: goldenrod;
}

/* All elements with the attribute "data-vegetable",
   containing the value "spicy", even among others,
   are given a red text color */
[data-vegetable~="spicy"] {
  color: red;
}
```

El resultado será:

Selector de atributos por valor textual

También conocidos como "RegExp-like selectors", pues proporcionan una forma de selección similar a las expresiones normales regular expression (aunque siendo estrictos, estos selectores no son verdaderas expresiones normales):

- [attr|=val] : Este selector elegirá todos los elementos con el atributo attr cuyo valor sea exactamente val o empieza por val- (nota: el guion no es un error, se usa para manejar códigos de lenguaje de programación).
-
- [attr^=val] : Seleccionará todos los elementos cuyo atributo attr comienza por el valor val.
- [attr$=val] : Este selector elegirá todos los elementos cuyo atributo attr termina por el valor val.

- [attr*=val] : Este seleccionará todos los elementos cuyo atributo attr contiene la cadena val (al contrario que [attr~=val], este selector no considera los espacios como separador de valores sino como parte del valor del atributo).

Continuemos con el ejemplo previo y añadámosle las siguientes reglas CSS:

```
/* Classic usage for language selection */
[lang|=fr] {
  font-weight: bold;
}

/* All elements with the attribute "data-quantity", for which
   the value starts with "optional" */
[data-quantity^="optional"] {
  opacity: 0.5;
}

/* All elements with the attribute "data-quantity", for which
   the value ends with "kg" */
[data-quantity$="kg"] {
  font-weight: bold;
}

/* All elements with the attribute "data-vegetable" containing
   the value "not spicy" are turned back to green */
[data-vegetable*="not spicy"] {
  color: green;
}
```

Con las nuevas reglas el resultado será:

Ingredients for my recipe: ***Poulet basquaise***

- **Tomatoes**
- Onions
- Garlic
- Red pepper
- **Chicken**
- Bacon bits
- Olive oil
- White wine

Pseudo-clases

Una pseudo-clase CSS consta de una clave precedida de dos puntos (:) que añadiremos al final del selector para indicar que daremos estilo a los elementos seleccionados solo cuando estos se encuentren en un estado determinado. Por ejemplo podríamos querer dar estilo a un elemento cuando este se muestre al pasarle el puntero del ratón, o una caja de selección al estar habilitada o deshabilitada o cuando un elemento es hijo directo de su padre en el árbol DOM.

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

No profundizaremos en cada una de las pseudo-clases — no es objetivo del Area de Aprendizaje presentarlas todas exhaustivamente, y se irán viendo con más detalle a lo largo del curso en el momento oportuno.

Ejemplo de pseudo-clase

Por ahora, veamos un ejemplo de cómo usarlas. Primero, un fragmento de HTML:

```
<a href="https://developer.mozilla.org/" target="_blank">Mozilla
Developer Network</a>
```

Y las reglas CSS:

```
/* These styles will style our link
```

```
   in all states */
a {
  color: blue;
  font-weight: bold;
}

/* We want visited links to be the same color
   as non visited links */
a:visited {
  color: blue;
}

/* We highlight the link when it is
   hovered (mouse), activated
   or focused (keyboard) */
a:hover,
a:active,
a:focus {
  color: darkred;
  text-decoration: none;
}
```

Pseudo-elementos

Los pseudo-elementos son parecidos a las pseudo-clases, con alguna diferencia. Estos son claves — ahora precedidas por (::) — que se añaden al final del selector para elegir cierta parte de un elemento.

- ::after
- ::before
- ::first-letter
- ::first-line
- ::selection
- ::backdrop

Todos disponen de comportamientos específicos e interesantes características que escapan a nuestros objetivos de aprendizaje por el momento.

Ejemplo de pseudo-elemento

Mostramos a continuación un ejemplo sencillo de CSS que selecciona los espacios situados justo después de todos los enlaces absolutos y en su lugar añade una flecha:

```
<ul>
  <li><a
href="https://developer.mozilla.org/en-US/docs/Glossary/CSS">CSS</a>
defined in the MDN glossary.</li>
  <li><a
href="https://developer.mozilla.org/en-US/docs/Glossary/HTML">HTML</a>
defined in the MDN glossary.</li>
</ul>
```

Añadamos ahora la regla CSS:

```
/* All elements with an attribute "href", which values
   start with "http", will be added an arrow after its
   content (to indicate it's an external link) */
[href^=http]::after {
  content: '↗';
}
```

para obtener el siguiente resultado:

- CSS ↗ defined in the MDN glossary.
- HTML ↗ defined in the MDN glossary.

VALORES Y UNIDADES EN CSS

Hay gran variedad de valores CSS, algunos de ellos muy usuales y otros que rara vez nos encontraremos. En este artículo no los vamos a estudiar todos de manera exhaustiva; solo aquellos que nos servirán en nuestro camino de aprendizaje de CSS por ser los más usuales. Vamos a ver los siguientes valores CSS:

Valores numéricos: Valores de longitud para expresar el ancho de elementos, de bordes, o tamaño de fuentes; también enteros sin unidades para expresar anchos de línea o veces que se repite una animación.

Porcentajes: Se usan para expresar tamaño o longitud — relativos por ejemplo al ancho y alto del contenedor previo o al tamaño de fuente predeterminado.
Colores: Para expresar colores de fondo, de texto, etc.

Coordenadas: Para expresar la posición relativa de un elemento desde, por ejemplo, el margen superior izquierdo de la pantalla.

Funciones: Para expresar imágenes de fondo o el degradado de la imagen de fondo.
Veremos ejemplos de unidades que encontraremos en el resto del tema CSS, y nos servirán para ¡cualquier CSS que nos encontremos! Nos acostumbraremos con rapidez.

Valores numéricos

Veremos los números utilizados en muchos sitios en unidades CSS. Esta sección trata acerca de los tipos más utilizados de valores numéricos.

Longitud y tamaño
Usamos unidades de longitud/tamaño (ver <length> como referencia) muy a menudo en diseños CSS, tipografía y otros. Veamos un ejemplo — primero el HTML:

```
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
```

Y el CSS:

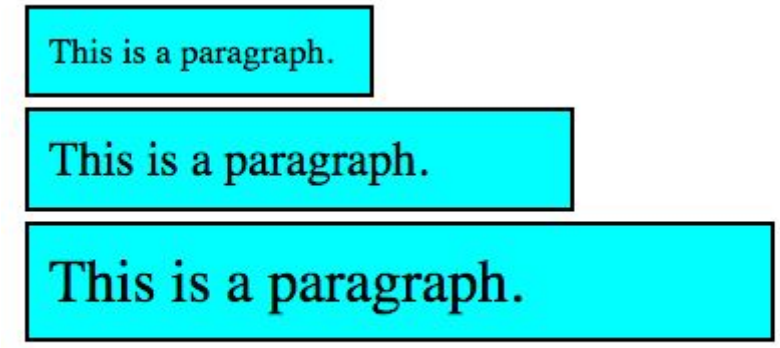
```
p {
  margin: 5px;
  padding: 10px;
  border: 2px solid black;
  background-color: cyan;
}

p:nth-child(1) {
  width: 150px;
  font-size: 18px;
}

p:nth-child(2) {
  width: 250px;
  font-size: 24px;
}

p:nth-child(3) {
  width: 350px;
  font-size: 30px;
}
```

El resultado será:



En este código estamos haciendo lo siguiente:

- Establecemos margin, padding (relleno) y border-width de los párrafos en 5, 10 y 2 pixeles respectivamente. Al haber un solo valor para el margen/relleno, los cuatro lados tendrán el mismo valor. El ancho del borde se establece mediante el valor de border.
-
- Establecemos el ancho width de los tres párrafos a valores cada vez con más píxeles, así conseguimos que las cajas sean más grandes a medida que bajamos.
- Establecemos font-size en los tres párrafos valores cada vez más grandes, así conseguimos que el texto se haga más grande a medida que bajamos. font-size hace referencia a la altura de cada carácter.

Nos referimos a píxeles (px) como **unidades absolutas** pues siempre tienen el mismo tamaño independientemente de cualquier otra medida. Otras unidades absolutas:

- mm, cm, in: Milímetros, centímetros, o pulgadas.
- pt, pc: Puntos ($1/_{72}$ de una pulgada) or picas (12 puntos.)

Probablemente sólo usaremos los píxeles.

También disponemos de unidades relativas, que lo son respecto al tamaño de fuente font-size o a la ventana actual:

- em: 1em es el tamaño de fuente del elemento actual (es el ancho de la letra M mayúscula). El tamaño de fuente por defecto que los navegadores usan antes de aplicar CSS es de 16 píxeles, lo que significa que este es el valor asignado por defecto a un elemento (1em). Ojo — los tamaños de fuente de los elementos se heredan de los padres, por lo que si a los padres se les aplica otros tamaños de fuente, la equivalencia en pixel de un em puede complicarse. No nos vamos a ocupar de esto

ahora — las herencias y los tamaños de fuentes los veremos en posteriores artículos y módulos. ems son las unidades relativas más usadas en el desarrollo web.

- ex, ch: Son respectivamente la altura de la x minúscula, y el ancho del número 0. Aunque no son tan soportadas por los navegadores como los ems.
- rem: (em raíz) funciona igual que em, excepto que está siempre igualará el tamaño del tamaño de fuente por defecto; los tamaños de fuente heredados no afectan, por lo que parece mejor solución que ems, rem no funciona en versiones antiguas de Internet Explorer.
- vw, vh: Estas son respectivamente $1/_{100}$ del ancho de la ventana, y $1/_{100}$ de la altura de la ventana. Tampoco son tan soportadas como los rems.

Usar unidades relativas es bastante práctico — se puede redimensionar los elementos HTML respecto al tamaño de la fuente o de la ventana, haciendo que la apariencia permanezca correcta si por ejemplo el tamaño del texto se duplica en el sitio web al ser utilizado por un usuario con deficiencia visual.

Valores sin unidades

En algunas ocasiones encontramos en CSS valores sin unidades — no siempre es un error, de hecho, es algo permitido bajo determinadas circunstancias. Imaginemos que queremos eliminar el margen o el relleno de un elemento, simplemente usaremos el 0, ya que 0 es 0 cualesquiera que fueran las unidades anteriores.

```
margin: 0;
```

Altura de línea sin unidades

Otro ejemplo es line-height, que establece la altura de cada línea de texto en un elemento. Podemos usar unidades para establecer la altura de la línea, pero normalmente es más fácil usar simplemente un valor que actúe como factor multiplicador. Supongamos el siguiente código HTML:

```
<p>Blue ocean silo royal baby space glocal evergreen relationship
housekeeping native advertising diversify ideation session.
Soup-to-nuts herding cats resolutionary virtuoso granularity
catalyst wow factor loop back brainstorm. Core competency baked in
push back silo irrational exuberance circle back roll-up.</p>
```

Y el siguiente CSS:

```
p {
  line-height: 1.5;
}
```

El resultado será:

> Blue ocean silo royal baby space glocal evergreen relationship housekeeping native advertising diversify ideation session. Soup-to-nuts herding cats resolutionary virtuoso granularity catalyst wow factor loop back brainstorm. Core competency baked in push back silo irrational exuberance circle back roll-up.

El (tamaño de fuente) font-size son 16px; la altura de línea 1.5 veces esta, o sea 24px.

Porcentajes

También podemos usar valores porcentuales para expresar la mayoría de cosas que requieran de valores numéricos, lo que nos permite crear, por ejemplo, cajas cuya anchura siempre cambie según el ancho del contenedor padre. En contraposición a las cajas cuya anchura este definida por un cierto valor (en px o en ems), que siempre serán de la misma longitud, incluso aunque cambie el ancho de los contenedores padres.
Mejor verlo con un ejemplo:

Primero dos cajas parecidas, expresadas en HTML:

```
<div>Fixed width layout with pixels</div>
<div>Liquid layout with percentages</div>
```

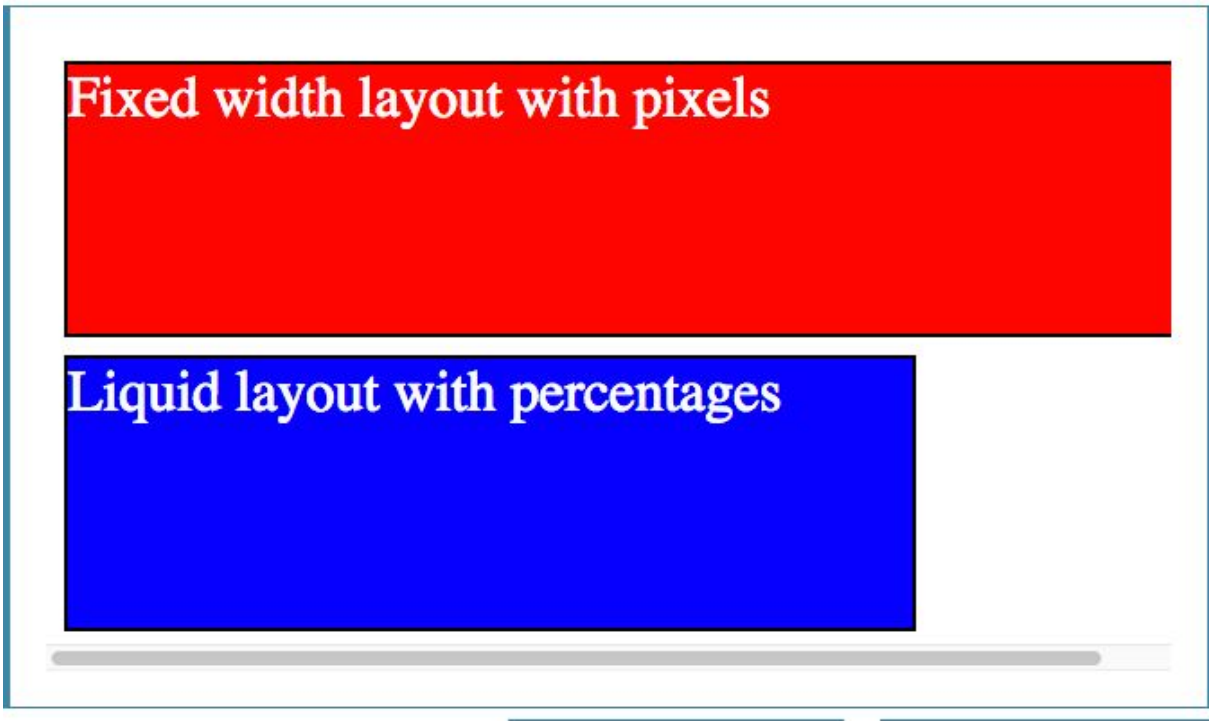
Y el CSS de estilos:

```
div {
  margin: 10px;
  font-size: 200%;
  color: white;
  height: 150px;
  border: 2px solid black;
}

div:nth-child(1) {
  background-color: red;
  width: 650px;
}

div:nth-child(2) {
  background-color: blue;
  width: 75%;
}
```

Nos proporciona el siguiente resultado:



Primero les hemos dado a ambos divs (margen, altura, tamaño de fuente, marco y color) margin, height, font-size, border y color. Luego hemos cambiado los colores del primer y el segundo div background-color para diferenciarlos. Hemos establecido el ancho del primer div con width en 650px, y del segundo al 75%, por lo que el primer div siempre tendrá el mismo ancho, aunque se redimensione la ventana (desaparecerá de la pantalla si la ventana es más estrecha que la pantalla), mientras que el segundo irá cambiando a medida que cambie su padre (en este caso el elemento <body>, que por defecto es el 100% del ancho de la ventana. También hemos establecido el font-size como un valor porcentual: 200%. Esto funciona diferente de como podríamos esperar, pero tiene sentido — nuevamente, esta nueva medida es relativa a la del predecesor, pues la hemos expresado con ems. En este caso, el tamaño de fuente del predecesor es de 16px — por defecto, entonces el valor calculado será el 200% de este, o sea, 32px. En realidad funciona de manera muy parecida a ems — pues el 200% es el equivalente a 2ems.

Estas dos maneras diferentes de formar las cajas se conocen como formato líquido (se redimensiona con la ventana del navegador), y formato de ancho fijo (permanece invariable). Ambos tienen distintos usos, por ejemplo:

- Un formato líquido podríamos usarlo para asegurarnos que un documento estándar cabrá siempre en la pantalla y se verá bien independientemente del tamaño de la pantalla del móvil.

- Un formato de ancho fijo lo usaremos para mantener el tamaño de un mapa online; la ventana del navegador podrá moverse por el mapa, viendo solo la parte del mapa escogida. La parte que veremos dependerá del tamaño de la ventana

Aprenderemos mucho más sobre formatos más adelante en el curso, en los módulos de Formato CSS y Diseño a medida (TBD).

Colores

Hay varias formas de referenciar colores en CSS, algunas más recientes que otras. Los mismos valores de color pueden utilizarse en todos los sitios en CSS, si nos referimos a color de texto, de fondo o cualquier otro.

El sistema de color estándar en los ordenadores actuales es el de 24 bits, que permite representar unos 16,7 millones de colores diferentes por canal (256 x 256 x 256 = 16.777.216).

Veamos los valores de los distintos tipos de sistemas de colores.

```
<p>This paragraph has a red background</p>

p {
  background-color: red;
}
```

Nos ofrece el resultado:

Es fácil de entender, aunque solo nos permite referenciar unos cuantos colores primitivos. Existen alrededor de 165 claves diferentes para su uso en los distintos navegadores.

Seguramente en nuestro trabajo utilizamos colores como el rojo, negro o blanco, pero además de estos necesitaremos conocer otro sistema de colores.

Valores hexadecimales

El siguiente sistema de color universal es el sistema hexadecimal, o códigos hex. Cada valor hex está compuesto por una almohadilla (#) seguida por seis números hexadecimales, cada uno de los cuales puede estar comprendido entre el 0 y la f (que representa el 15) — 0123456789abcdef (16 símbolos). Cada par de valores representa uno de los canales primarios — rojo, verde y azul — y nos permite referenciar cualquiera de los 256 valores disponibles para cada uno (16 x 16 = 256).

Por ejemplo, el código:

```
<p>This paragraph has a red background</p>
<p>This paragraph has a blue background</p>
<p>This paragraph has a kind of pinky lilac background</p>
```

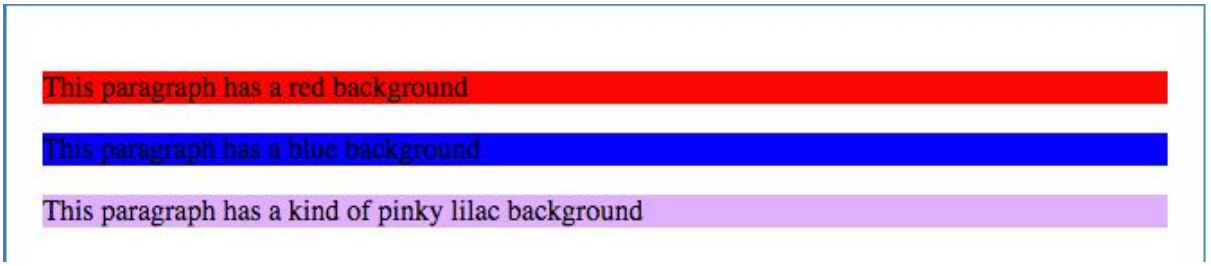
CSS

```
/* equivalent to the red keyword */
p:nth-child(1) {
  background-color: #ff0000;
}

/* equivalent to the blue keyword */
p:nth-child(2) {
  background-color: #0000ff;
}

/* has no exact keyword equivalent */
p:nth-child(3) {
  background-color: #e0b0ff;
}
```

Ofrece el siguiente resultado:



Estos valores son un poco más complejos y difíciles de entender, pero es mucho más versátil que las claves — podemos utilizar valores hex para representar cualquier color que queramos usar en nuestra paleta de colores.

RGB

El tercer sistema al que vamos a hacer referencia es RGB. Un valor RGB es una función — rgb() — a la que pasamos tres parámetros que representas los colores primarios rojo, verde y azul, parecido al sistema hex. La diferencia radica en que, en lugar de representar cada canal por un par de números, ahora lo hacemos simplemente por un número decimal entre el 0 y el 255.

Reescribamos el ejemplo anterior con el sistema RGB:

```
<p>This paragraph has a red background</p>
<p>This paragraph has a blue background</p>
<p>This paragraph has a kind of pinky lilac background</p>
```

CSS

```
/* equivalent to the red keyword */
p:nth-child(1) {
  background-color: rgb(255,0,0);
}

/* equivalent to the blue keyword */
p:nth-child(2) {
  background-color: rgb(0,0,255);
}

/* has no exact keyword equivalent */
p:nth-child(3) {
  background-color: rgb(224,176,255);
}
```

El sistema RGB es casi tan soportado como el hexadecimal — probablemente no nos encontremos en nuestro trabajo ningún navegador que no los reconozca ambos. Los valores RGB son sin duda un poco más intuitivos y fáciles de comprender que los RGB.

Opacidad

Existe otra forma de especificar la transparencia vía CSS — la propiedad opacity. En lugar de fijar la transparencia de un determinado color, aquí fijamos la transparencia de todos los elementos seleccionados y sus hijos. De nuevo, veamos un ejemplo para estudiar la diferencia:

```
<p>This paragraph is using RGBA for transparency</p>
<p>This paragraph is using opacity for transparency</p>
```

CSS

```
/* Red with RGBA */
p:nth-child(1) {
  background-color: rgba(255,0,0,0.5);
}

/* Red with opacity */
p:nth-child(2) {
```

```
  background-color: rgb(255,0,0);
  opacity: 0.5;
}
```

This paragraph is using RGBA for transparency

This paragraph is using opacity for transparency

Herramientas

Para el desarrollo de las clases recomienro usar https://codepen.io/pen/ para las demostraciones de codigo.

La confirmación de los ejercicios y seguimiento a la guía debe ser en el editor de su preferencia **Sublime Text** (https://www.sublimetext.com/)

Ejercicios

Con los conocimientas adquiridos clonar la web https://www.modulosdesk.com/, subir un screenshot o una imagen en PNG o PDF se su trabajo.