Conceptos Básicos De frontend

SINCOSOFT

HTML (HyperText Markup Language)

1

Es un lenguaje que nos permite definir la estructura de las páginas Web, se compone de elementos como párrafos, encabezados, imágenes y vínculos (links), entre muchos otros.

La mayoría de etiquetas necesitan una etiqueta de cierre que es muy similar a una etiqueta pero tiene un slash (/) antes de la palabra. Por ejemplo, para mostrar un párrafo se utilizan la etiqueta y la etiqueta de cierre .

SINCOSOFT

Diferencias entre

Versiones de HTML

2

HTML 2: Es considerado la primera versión del HTML, el cual no soportaba tablas y donde la declaración explícita de los elementos body, html y head es opcional.

HTML 3.2: W3C (World Wide Web Consortium) incorporó los últimos avances de las páginas web como applets de Java y texto que fluye alrededor de las imágenes.

HTML 4.0: Llegan las hojas de estilos CSS, la posibilidad de incluir pequeños programas o scripts en las páginas web, mejora de la accesibilidad de las páginas diseñadas, tablas complejas y mejoras en los formularios.

HTML 4.01: Actualización de la versión HTML 4.0, por lo que no incluye novedades significativas.

HTML 5: Es la quinta revisión importante del lenguaje básico HTML.

Novedades que trae

HTML5

3

- HTML5 define una sintaxis que es compatible con HTML4 y XHTML 1.0. Por tanto, un salto de línea se puede escribir como
 (HTML4) o
 (XHTML 1.0).

- Para definir el juego de caracteres se introduce un nuevo atributo para la etiqueta <meta>:
  <meta charset=”UTF-8">

  aunque todavía es posible utilizar el método tradicional:
  <meta http-equiv=”Content-Type” content=”text/html; charset=UTF-8">

4

- Se simplica el DOCTYPE:

  <!DOCTYPE html>

- HTML5 permite incluir elementos de SVG y MathML.
  Se introducen nuevos elementos, como: section, article, aside, header, footer, etc.
  Se introducen nuevos atributos, como: media, charset, autofocus, placeholder, etc.
  Algunos elementos cambian, como: a, b, i, menu, etc.
  Algunos atributos cambian, como: type, name, summary, etc.
  Algunos elementos desaparecen, como: basefont, big, center, etc.
  Algunos atributos desaparecen, como: align, background, bgcolor, etc.

  Mejora de las API, como: getElementsByClassName() y innerHTML.

SINCOSOFT

Estándar W3C

W3C maneja diferentes estándares de acuerdo al tipo de desarrollo a realizar.
Para aplicaciones web:

5

- Es fundamental conservar la estructura semántica de cada elemento, por accesibilidad y legibilidad: *por ejemplo un link dentro de un texto largo siempre deberá agregarse dentro de una etiqueta p, la cual conformará un párrafo de texto.*

- Para la reutilización de estilos, se deberá hacer uso de archivos externos para que se compartan las reglas definidas como globales para la aplicación.

- Webfonts: La utilización de fuentes tipográficas sin la instalación de las mismas en el sistema operativo.

- Actualmente los íconos si son de tipo flat, pueden ser utilizados como fuente tipográfica creada a partir de elementos SVG, para disminutir el consumo de recursos.

https://www.w3.org/standards/webdesign/htmlcss

SINCOSOFT

6

Aprendiendo

CSS

(Cascading Style Sheets)

SINCOSOFT

¿Cómo funcionan las hojas de estilo en cascada?

Las hojas de estilo CSS son un conjunto reglas que enumeran en un archivo .css y que describen el aspecto que deben tener los diferentes elementos HTML de una página.

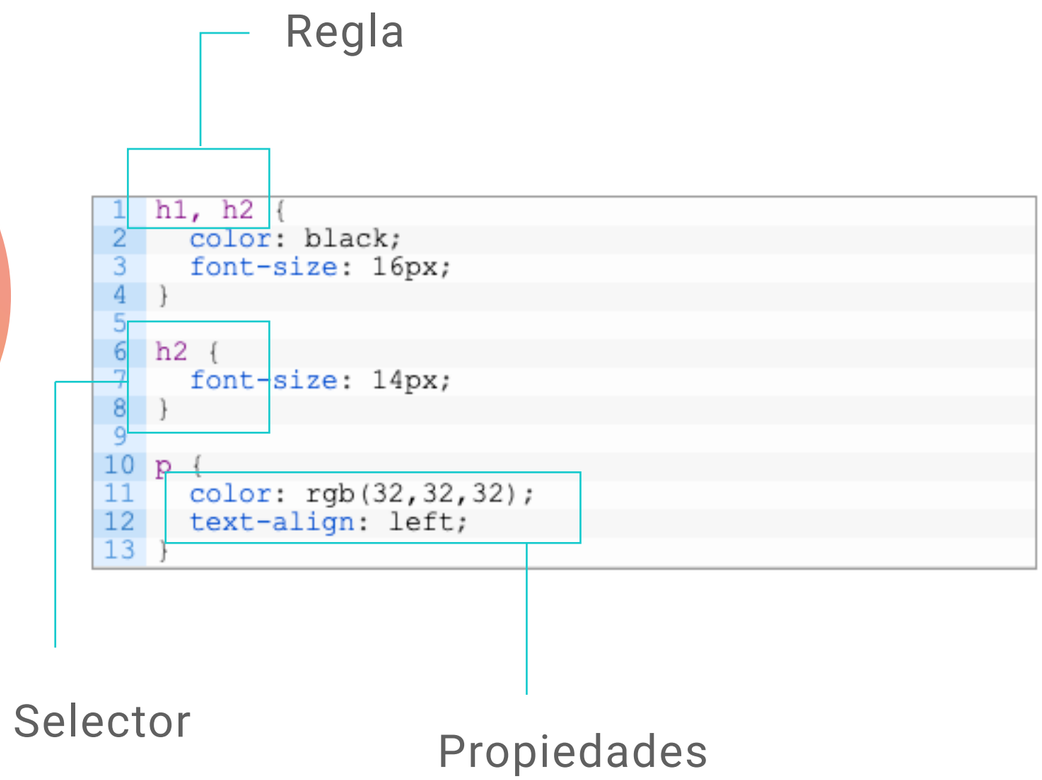
Lo interesante de esto es que funcionan con una filosofía de patrones o plantillas, es decir, no es necesario especificar cada uno de los elementos, sino que se pueden definir reglas como estas dos:

“Los títulos de nivel 1 y 2 han de ser de color negro y un tamaño de fuente de 16 y 14 pixeles respetivamente.”

“El texto de los párrafos están alineados a la izquierda, tienen un tamaño de fuente de 12 pixeles y un color gris oscuro.”

Reglas, selectores y propiedades

8



En el ejemplo anterior tenemos un total de 3 reglas, cada una se compone de un selector que indican a qué elementos HTML aplica la regla que es el texto que precede a las llaves de apertura y cierre de la regla. En la primera regla, el selector es “h1, h2”, en la segunda “h2” y en la tercera “p”.

Cada regla se compone a su vez de propiedades que especifican qué exactamente se hará con esos los elementos HTML a los que aplica.

SINCOSOFT

Herencia, sobreescritura y conflictos de estilos

El nombre de hojas de estilo en “cascada” no es casual, expresa que los estilos que especifican con reglas se pueden heredar de una manera jerárquica.

La etiqueta <body> que es la que envuelve el contenido de cualquier página web tenga un tipo de letra “Arial”. Esto no tendría mucho sentido si no fuera porque gracias a la capacidad de herencia de las reglas de este modo por defecto cualquier elemento hijo como un título o un párrafo va a “heredar” ese estilo.

```
4   body {
5     font-family: Arial;
6   }
7
8   p {
9     font-family: Verdana;
10  }
```

En esta última regla aplica el principio de “especificidad”. En principio, se plantearía un conflicto entre la regla general de <body> con lo que dice la regla de <p>, pero se resuelve fácilmente puesto que se aplica la regla más específica y referirse a un párrafo es más específico que referirse a “los elementos hijos que pueden dentro de <body>”.

Propiedades básicas

que deberías conocer

Maquetación básica

10

- width: Ancho de un elemento.
- height: Alto de un elemento.
- vertical-align: Alineamiento vertical dentro de un elemento.
- margin: Espacio que se añade entre el elemento y sus vecinos. Se puede diferencia por lado (arriba, abajo, izquierda, derecha).
- padding: Relleno interior que se añade en los bordes del A diferencia de margin, cuenta para el tamaño del elemento.
- float: Mueve el elemento todo lo posible hacia el lado indicado. Esta propiedad se usa en el posicionamiento flotante de CSS. El tema del posicionamiento en CSS no es trivial y conviene estudiar cómo funciona antes de usar esta propiedad.

Fuentes y texto

- font-family: Tipo de letra
- font-size: Tamaño de letra
- font-weight: Peso (normal, negrita, …)
- font-style: Estilo (normal, cursiva, …)
- text-decoration: “Decoraciones” como subrayado, tachado, etc.
- text-align: Alineación del texto (izquierda, derecha, etc.)
- text-transform: Mostrar un texto en mayúsculas, minúsculas o la primera letra de cada palabra en mayúsculas.

Color y fondos

- color: Color del elemento. Se puede especificar en diferentes formatos como palabras predefinidas (red, green, etc.) RGB o como valor hexadecimal.
- background-color: Color del fondo del elemento.
- background-image: Permite especificar una imagen de fondo.

- background-repeat: Permite usar una imagen a modo de mosaico en diferentes modalidades.
- box-shadow: Crear un efecto de sombra para un elemento.

Listas

- list-style-image: Usar la imagen especificada como viñeta para la lista.
- list-style-type: Diferentes estilos de viñetas y estilos de numeración para elementos de lista.

Bordes

- border: Añade un borde a un elemento y establece algunas propiedades (grosor, estilo de línea, etc.)
- border-color: Color del borde.
- border-style: Diferentes estilos para el borde (sólido, puntos, etc.)
- border-radius: Permite crear esquinas redondeadas para un elemento.

Posicionamiento

- **position: static;** Es la posición por defecto en la del navegador

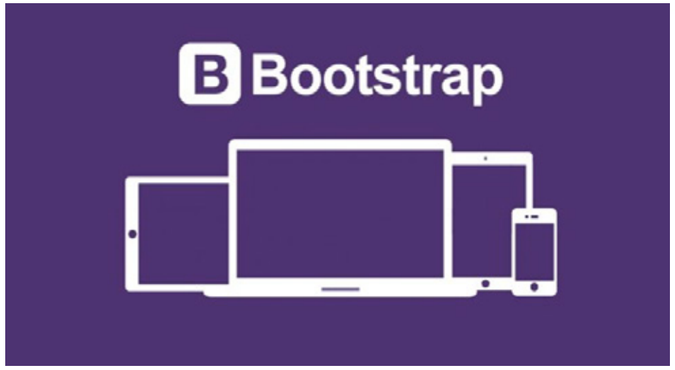
- **position: relative;** Nos permite mover el elemento a una ubicación relativa a su posición original utilizando las propiedades: left, top, right, bottom, y se utiliza más para posicionar elementos de forma absoluta dentro del elemento.

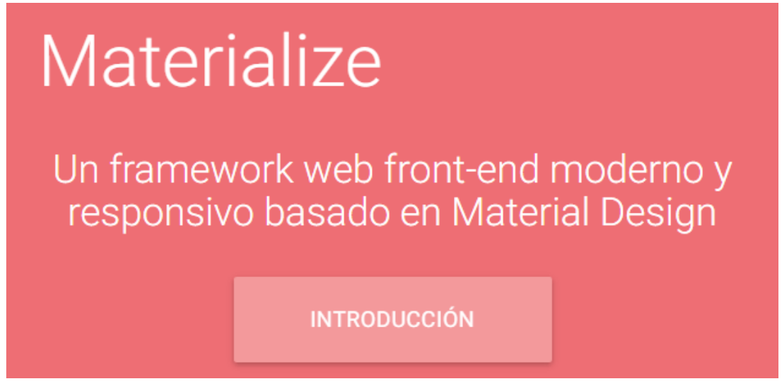
- **position: absolute;** Nos permite ubicar un elemento a una ubicación relativa al ancestro más cercano que esté posicionado con position: relative, o a la ventana del navegador en su defecto, y se utilizan las propiedades top, left, right y bottom para ubicar el elemento.

- **position: fixed;** Esta propiedad nos permite ubicar un elemento de manera fija relativamente a la ventana del navegador. No importa si se utilizan las barras de desplazamiento, el elemento va a permanecer fijo en su posición.

Frameworks (css)

Lo importante a la hora de decidir que framework usar no debe ser la cantidad de elementos, ni tampoco el diseño base de los elementos. Todo lo contrario. Debemos sentirnos cómodos con la librería, entender como funciona y sentir que es de utilidad, que nos ahorra tiempo. Es lo más importante, encontrar el vínculo que nos hace avanzar y mejorar en nuestro trabajo. Y no todos los frameworks lo consiguen, hay una framework para cada tipo de desarrollador Front-End.





Ventajas de usar un framework(css)

- Ahorrar tiempo: Escribir una y otra vez el mismo código es repetitivo y no te lleva a ninguna parte. Es uno de los principios que comentábamos en nuestro listado de buenas prácticas para escribir CSS. Con un framework tendrás ya una base sobre la que empezar a maquetar el diseño.Escalabilidad: Usar un Framework CSS permite a tu diseño tener la posibilidad de crecer en el futuro. Si añadimos una nueva sección, una nueva página o un nuevo elemento no necesitaremos escribir nuevas lineas de CSS, podemos reutilizar los estilos ya predefinidos, centrándonos en el HTML y en los detalles que hagas a esa página singular respecto al resto.

- Testeado en navegadores: la mayoría de los frameworks son usados por muchos usuarios y los mismos usuarios reportan los fallos o errores que encuentran en los distintos navegadores, para que puedan ser arreglados. Esto hace que a medida que pase el tiempo, el frameworks vaya mejorando su compatibilidad y no tengas que preocuparte de si tu diseño se ve bien en Safari, Chrome, Firefox…

- Mobile first: La mayoría de los frameworks son responsive design, y además están construidos basándose en la metodología Mobile First, por lo que nos aseguramos una óptima visualización en los dispositivos de menor resolución sin sobrecargar con estilos y propiedades innecesarias dichas resoluciones.

- Escalabilidad: Usar un Framework CSS permite a tu diseño tener la posibilidad de crecer en el futuro. Si añadimos una nueva sección, una nueva página o un nuevo elemento no necesitaremos escribir nuevas lineas de CSS, podemos reutilizar los estilos ya predefinidos, centrándonos en el HTML y en los detalles que hagas a esa página singular respecto al resto.

Desventajas de usar un framework (css)

- Puede limitar la creatividad de tu diseño. Partir de una estructura de rejilla o de unos márgenes y formas preestablecidos puede hacer que tu diseño se vea condicionado por el mismo y el nivel de creatividad de tu proyecto no sea el mismo que si partes de una hoja en blanco.

- Líneas de código innecesarias. Seguramente no harás uso del 50% de las especificaciones y clases escritas en el CSS. Es lo normal, sin embargo los usuarios se descargan los archivos completos, por lo que el peso de la web será más alto, y por lo tanto la carga más lenta.

- Tiempo de aprendizaje: Aunque la curva de aprendizaje suele ser rápida, es cierto que familiarizarnos con el framework nos llevará tiempo y hasta pasados un par de proyectos no conoceremos la mayor parte del código.

Preprocesadores
de CSS

18

En términos simples, mediante un lenguaje de script escribimos código parecido al que usamos en CSS (esto dependerá del preprocesador que estemos usando), luego este código será compilado y como resultado de esta compilación tendremos un archivo CSS.

Diferencias entre preprocesadores
de CSS

Los preprocesadores más utilizados son Less, Saas, Stylus. Ninguno es mejor que el otro, cada uno tiene sus ventajas y desventajas.

Less: compacto y rápido, apropiado para proyectos pequeños.

Sass: proyecto Open Source, una comunidad muy grande y ampliamente usado.
Stylus: bastante avanzado y completo, es el más cercano a un lenguaje de programación.

Todos difieren en su sintaxis, pero tienen fundamentos parecidos y un mismo objetivo, que es hacer más organizado y eficiente el código CSS durante el desarrollo.

20

Practicando

CSS

SINCOSOFT

21

Intenta construir la estructura

de ésta página

Te enseño cómo

agregar una tipografía personalizada.

22

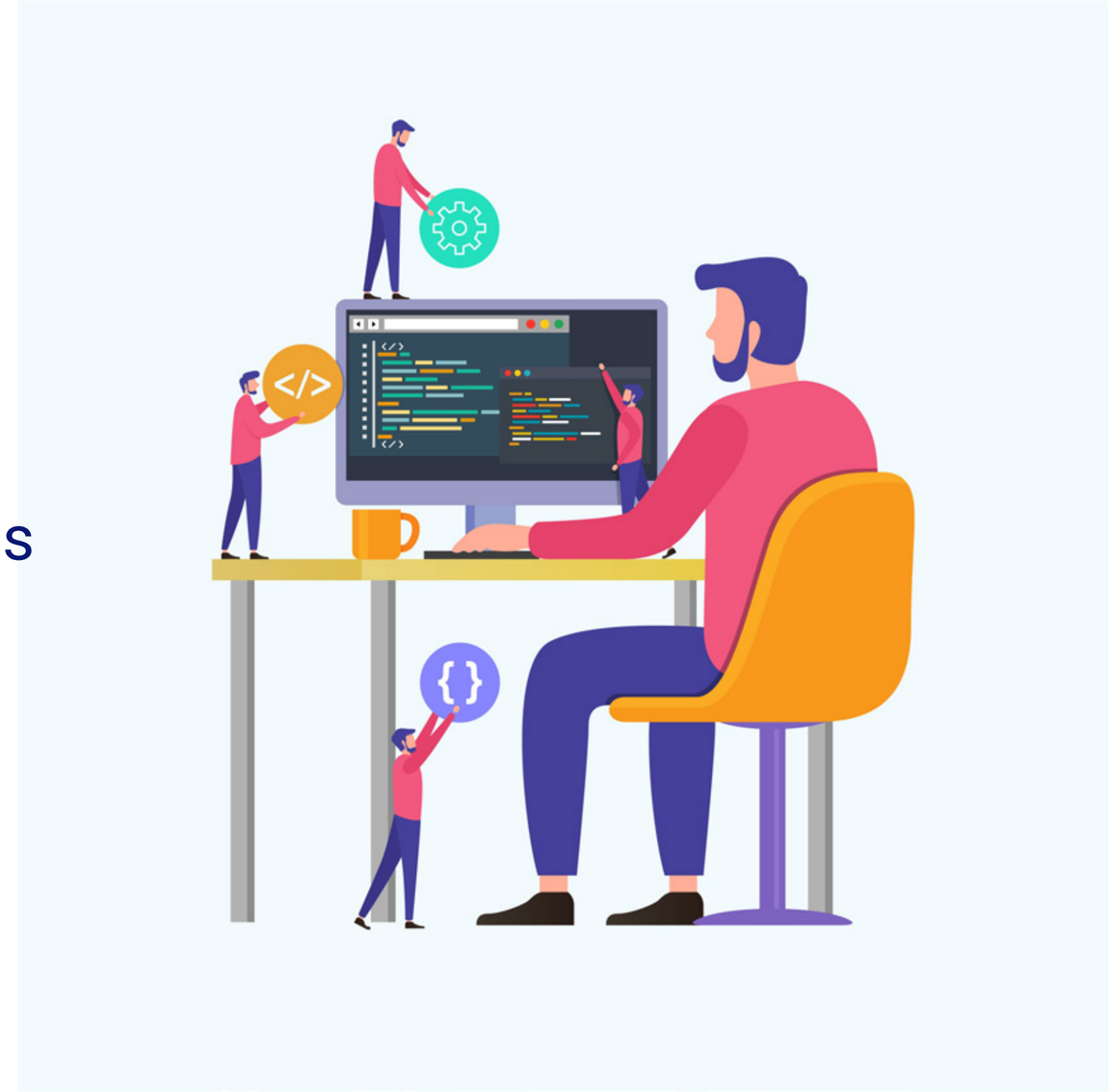
Google Fonts

transfonter

23

Trucos divertidos

animaciones sencillas, cambiar textos
con css, y librerías que puedes
usar para esto

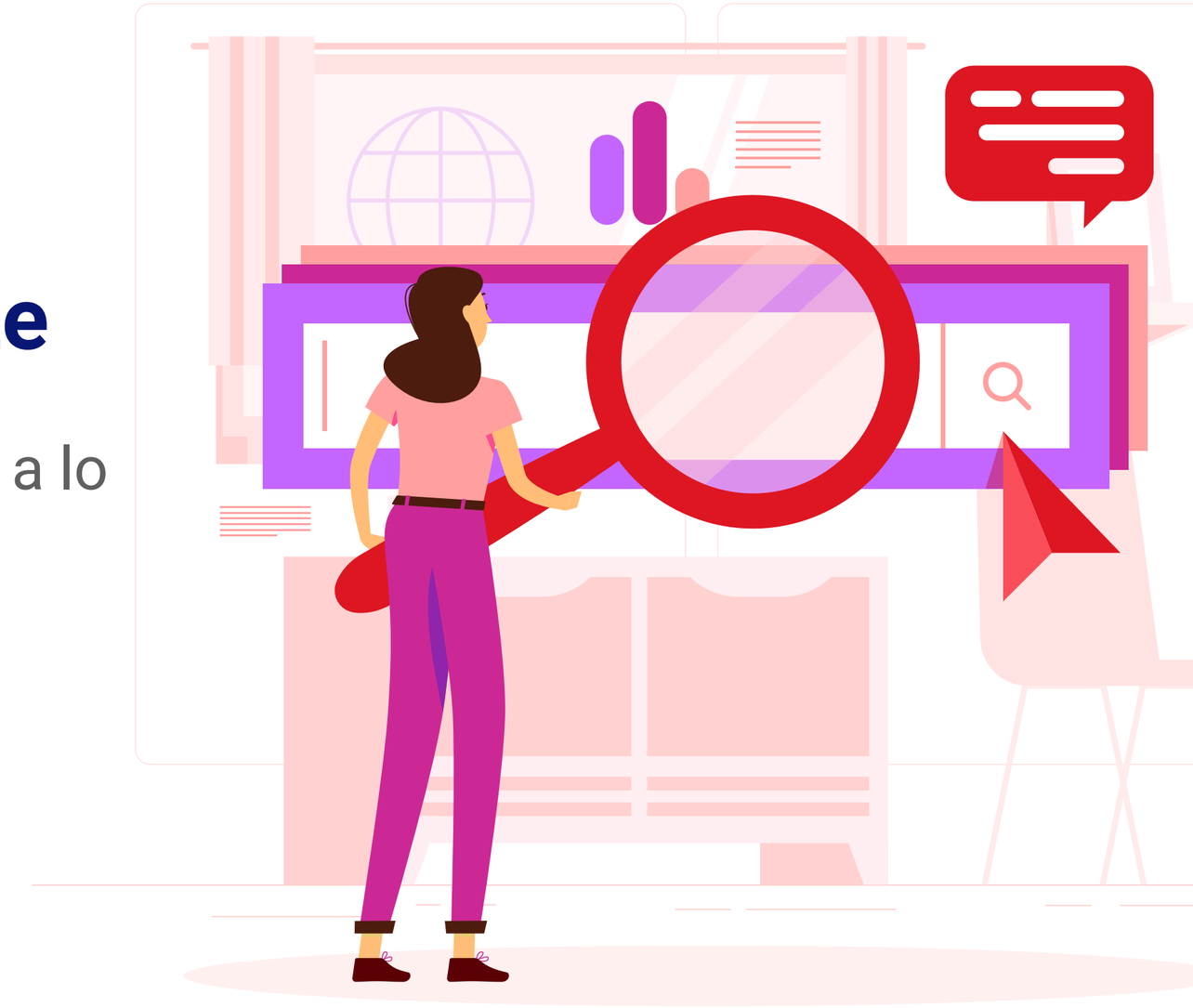


24

Revisemos si entendiste

Juntos encontremos la respuesta a lo que sucede con estos estílos,

¡Es hora de analizar!



25

Si tengo este html

```html
<body>
    <div class="container">
        <div class="links">
            <a href="" class="link"> primer link</a>
            <a href="" class="link" id="link"> segundo link</a>
            <a href="" class="link"> tercer link</a>
        </div>
    </div>
</body>
```

y tengo estos estilos

```css
a{
    color: red;
    display: block;
}
#link{
    color: purple;
}
.container a{
    color: blue;
}
.container a.link{
    color: green;
}
```

¿Qué color tiene la segunda a?

¿Por qué mis estilos no funcionan en IE?

26

```
.container{
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
}
.container .links{
    flex-basis: 50%;
}
```

La respuesta es tan sencilla que no necesito revisar el diseño en el navegador

¿qué texto tiene al final mi botón si le agregué la clase active?

27

```html
<button class="active">
    ver más opciones
</button>
```

```css
button{
    -webkit-appearance: none;
    border: 0;
    background-color: #77bcdd;
    color: white;
    padding: 8px 11px;
    font-size: 13px;
    border-radius: 30px;
}
button.active{
    background-color: red;
    font-size: 0;
    text-indent: 0;
}
button.active::before{
    content: 'Cerrar';
    font-size: 13px;
}
```

La respuesta es tan sencilla que no necesito revisar el diseño en el navegador

Fuentes

https://guias.makeitreal.camp/html-y-css/posicionamiento

https://luisforgiariniblog.com/cual-diferencia-entre-html-html5/

https://www.ciudadano2cero.com/aprender-css-basico-desde-cero/#Como_funcionan_las_hojas_de_estilo_en_cascada

https://www.silocreativo.com/mejores-frameworks-css/

https://devcode.la/blog/preprocesador-de-css/