

Agencia de
Aprendizaje
a lo largo
de la vida

FULL STACK JAVA Clase:

CSS

Les damos la bienvenida

Vamos a comenzar a grabar la clase

Temas a tratar:

- Introducción a CSS
- Integración con HTML
- Propiedades (Atributo y valor)
 - Color, Tipografía, Tamaño de fuente
- Agrupamientos
 - Div, Span, Atributos globales
- Especificidad (Etiquetas semánticas)
- Iconos

Introducción a CSS

¿Qué es CSS?

CSS (Cascading Style Sheets) es un lenguaje de maquetación esencial para el desarrollo Frontend.

Se encarga del diseño visual del sitio, manipulando desde el color o tipo de letra hasta la organización visual en pantalla y animaciones.

Con el pasar de los años el concepto audiovisual se ha revolucionado, al punto de llevar el formato de la pantalla vertical como un estandarte de la industria.

¿Qué es CSS?

Ahora tenemos múltiples dispositivos, diversos formatos, podemos ver reproducciones en celulares, televisores, heladeras, relojes; dispositivos con formatos clásicos (4:3), horizontales (16:10, 16:9), verticales (9:16), cuadrados, esta variedad llevó a replantear la forma de trabajar y facilitar las labores que veníamos haciendo, el mercado extendió sus posibilidades.

¿Qué es CSS?

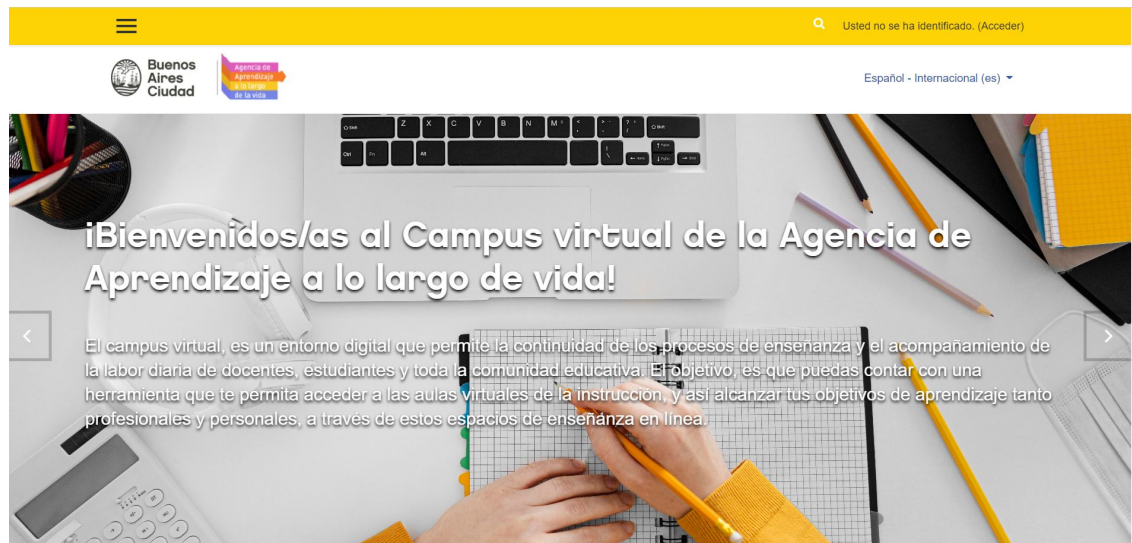
CSS también nos facilita la forma en que podemos manejar nuestro producto entre diferentes dispositivos, ofreciéndonos herramientas para simplificar la labor y darle compatibilidad.

Este proceso de adaptar nuestro sitio se denomina “responsive”.

Cuando una página es Responsive nos referimos a que se ajusta al lugar donde se ejecuta, modificando (si es necesario) su estructura visual para ofrecer una experiencia óptima al usuario.

¿Qué es CSS?

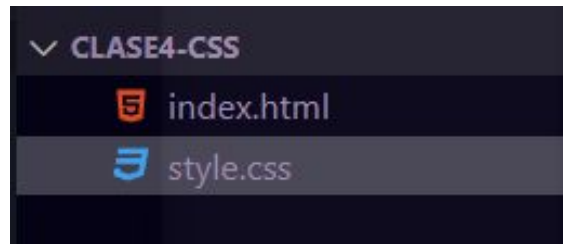
Con CSS nuestra página toma un aspecto visual totalmente personalizado, y se puede adaptar al dispositivo que lo ejecute, sea vertical u horizontal.



Integración con HTML

Integración con HTML

Llegó el momento de crear nuestra primera hoja de estilos, para ello debemos crear un archivo de extensión “.css” .



Por lo general usaremos una sola hoja de estilos (a menos que la forma de afrontar el proyecto amerite otra estrategia), usaremos como nombre “style” y lo ingresamos como atributo **href** de la etiqueta **<link>**.

Integración con HTML

Tenemos tres formas de integrar elementos CSS:

Integrando hojas de estilo de fuentes externas:

Es la forma que venimos viendo en los ejemplos, mediante el tag *link*, es la mejor forma de integrar código CSS a nuestro sitio, además de mantener nuestro código ordenado.

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

Integración con HTML

Mediante incrustación con la etiqueta style:

Podemos hacer uso de un tag especial, llamado *style*. Todo lo que esté dentro de él deberá ser código CSS.

Esto se suele usar sólo si se hará un uso mínimo de css, de hacerlo por orden visual deberíamos colocarlo en la cabecera de la página (head) ya que es código que no se interpretará visualmente (aunque sí lo afectará).

```
<title>Clase 4</title>
<style>
  h2 {
    color: #4dc04d;
    font-family: Verdana, C
  }
</style>
/head>
```

Integración con HTML

Incrustación por atributo (en línea):

El tercer tipo de integración es mediante un atributo a la etiqueta que queremos afectar. La etiqueta **style** nos permite ingresar todo tipo de código css a dicho tag, pero también es el menos recomendado. Su jerarquía es mayor a las otras integraciones, por ende puede anular los otros valores cargados por el tag style o link , además de entorpecer la lectura del código nos puede generar más problemas que soluciones.

```
<div style="color: hsl(3, 75%, 61%)">Este texto se verá afectado por el atributo style</div>
```



**Propiedades:
Atributo y valor**

Propiedades: Atributo y valor

Como venimos viendo unas clases atrás, una idea que nos rondará en el mundo de la programación (a partir de ahora) es la representación que haremos de atributo y valor.

Como cuando encontramos un término en el *diccionario* (e inclusive en algunos lenguajes se lo llama así) tenemos la palabra y a su lado (dos puntos mediante) su significado; nosotros manejaremos con gran similitud esta interpretación.

Ejemplos de atributo: valor

Como podemos ver: en varios lenguajes tenemos el concepto atributo: valor, en algunos a esta asociación la llamamos clave: valor (key: value)

```
index.html > ...  
30 <div class="clase">Tex  
31
```

```
[...} archivo.json > ...  
1 {  
2   "nombres": "Pepe Luis",  
3   "apellidos": "Biondi Sangango",  
4   "edad": 25  
5 }  
6
```

```
style.css > ...  
1 .clase {  
2   color: blue;  
3   font-size: 2em;  
4 }  
5
```

```
archivo.py > ...  
1 diccionario = {  
2   "nombres": "Pepe Luis",  
3   "apellidos": "Biondi Sangango",  
4   "edad": 25  
5 }  
6
```

```
JS archivo.js > ...  
1 const objeto = {  
2   "nombres": "Pepe Luis",  
3   "apellidos": "Biondi Sangango",  
4   "edad": 25  
5 }  
6
```


CSS: Color, tipografía, tamaño de fuente

Un atributo de una clase CSS puede tener más de un valor, un atributo en particular también se puede descomponer en varios atributos... No todos, sino algunos particulares (que manejaremos con más precisión a medida que tengamos más práctica con el código), este es un ejemplo de lo que hablamos, ambas clases manifiestan lo mismo en la etiqueta que les sea asignada:

```
.borde-un-atributo {  
  border: 1px solid red;  
}
```

```
.borde-varios-atributos {  
  border-width: 1px;  
  border-style: solid;  
  border-color: red;  
}
```

CSS: Color, tipografía, tamaño de fuente

A lo largo de nuestros proyectos vamos a usar muchos atributos, y entre tantas variantes algunas podrán modificar colores y letras. Hoy nos centraremos en estas cualidades.

Color: Los colores transmiten muchas cosas, incluso hay un análisis de la psicología de los colores en el diseño; los mismos se componen de manera aditiva (sumando intensidades de luz) y sustractiva (absorbiendo las ondas que percibimos). Al usar un medio de luz (el monitor genera luces, de manera aditiva; y con ella compone los colores que vemos en pantalla) usaremos determinados valores para componerlos.

Colores

Este ejemplo será inválido en un estilo CSS, pero nos servirá de referencia.

Tendremos algunos colores básicos a los que podremos acceder con su nombre en inglés (en este caso *aqua*).

Luego vemos la composición de colores mediante **rgb** y **rgba**.

```
.color{  
  color: ■ aqua;  
  color: □ rgb(0,0,0);  
  color: □ rgba(0, 0, 0, 50%);  
  color: □ #000000;  
  color: □ #00000085;  
  color: □ hsl(0, 0%, 0%);  
  color: □ hwb(0 0% 100%)  
}
```

Colores

El valor rgb hace referencia a los colores **r**ed, **g**reen y **b**lue, los mismos se miden desde 0 (sin intensidad, nulo) hasta el 255 (en su máxima intensidad).

En el ejemplo vemos que están en cero, por lo que los tres canales están anulados, sin intensidad, logrando el color negro.

Si tuvieran de valor 255 el color logrado sería el blanco.

```
.color{  
  color: aqua;  
  color: rgb(0,0,0);  
  color: rgba(0, 0, 0, 50%);  
  color: #000000;  
  color: #00000085;  
  color: hsl(0, 0%, 0%);  
  color: hwb(0 0% 100%)  
}
```

Colores

El valor rgba incluye un cuarto canal que representa al **alfa** (la transparencia), ésta puede variar del 0 al 1 (cuantificado en décimas, por lo que si deseamos que sea exactamente medio transparente pondremos un valor de 0.5 o 0.50). También puede usarse un valor de porcentaje, el que nos resulte más cómodo.

```
.color{  
  color: ■ aqua;  
  color: ■ rgb(0,0,0);  
  color: ■ rgba(0, 0, 0, 50%);  
  color: ■ #000000;  
  color: ■ #00000085;  
  color: ■ hsl(0, 0%, 0%);  
  color: ■ hwb(0 0% 100%)  
}
```

Colores

Luego podemos ver valores **hexadecimales** (acompañados de un símbolo numeral a la izquierda). cada par de números representan un canal r, g y b.

El valor de cada unidad va del 0 al f, por lo que el código #ff0000 representaría al rojo, el #00ff00 al verde y el #0000ff al azul.

Debajo tenemos un valor con dos unidades más, esas dos nuevas unidades representan al canal alfa.

```
.color{  
  color: ■ aqua;  
  color: □ rgb(0,0,0);  
  color: □ rgba(0, 0, 0, 50%);  
  color: □ #000000;  
  color: □ #00000085;  
  color: □ hsl(0, 0%, 0%);  
  color: □ hwb(0 0% 100%)  
}
```

Colores

Los siguientes canales son los menos utilizados en css, constan de otras maneras de componer colores, el superior es **hsl**: **h**ue, **s**aturation y **l**ightness y representan la matiz/tono de 0 a 360, saturación y luminosidad del objeto, y **hwb**: **h**ue, **w**hiteness, **b**lackness (matiz/tono de 0 a 360, intensidad de blancos, intensidad de negros).

```
.color{  
  color: ■ aqua;  
  color: □ rgb(0,0,0);  
  color: □ rgba(0, 0, 0, 50%);  
  color: □ #000000;  
  color: □ #00000085;  
  color: □ hsl(0, 0%, 0%);  
  color: □ hwb(0 0% 100%)  
}
```

Colores

Cuando se repite un atributo el sistema tendrá en cuenta la última escrita en la clase.

El atributo color modificará el color del texto, pero hay muchos atributos que tienen como componente de sus valores al color, como el background, los bordes, las sombras.

```
.color{  
  color: aqua;  
  color: rgb(0,0,0);  
  color: rgba(0, 0, 0, 50%);  
  color: #000000;  
  color: #00000085;  
  color: hsl(0, 0%, 0%);  
  color: hwb(0 0% 100%)  
}
```


Tipografía

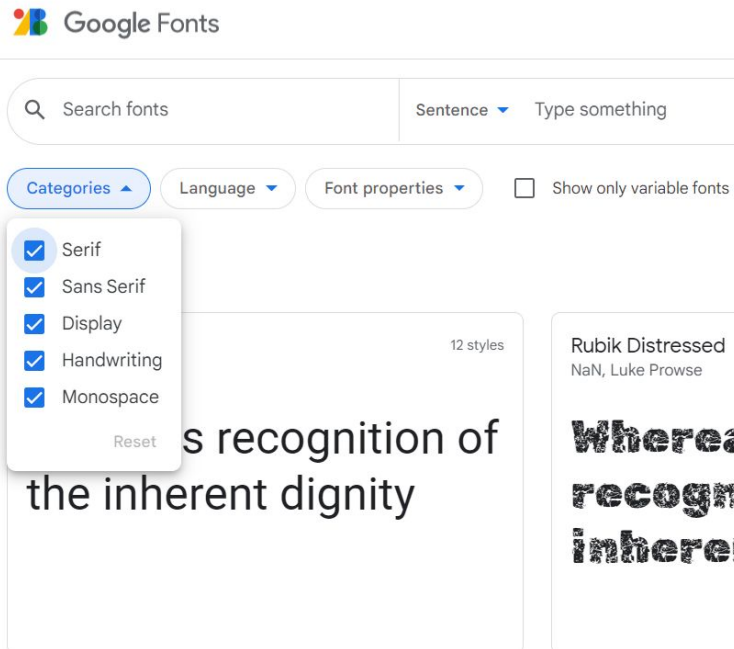
Podremos poner diferentes tipos de letras, desde las básicas que nos ofrece el sistema hasta agregar nuestras propias fuentes o importar desde otros sitios particulares como Google Fonts.

El atributo que usaremos para asignar nuestra “font” (fuente) es **font-family**, y a continuación veremos la forma de importarlas o instalarlas en el proyecto.

```
.fonts{  
  font-family: 'Times New Roman', Times, serif;  
}
```

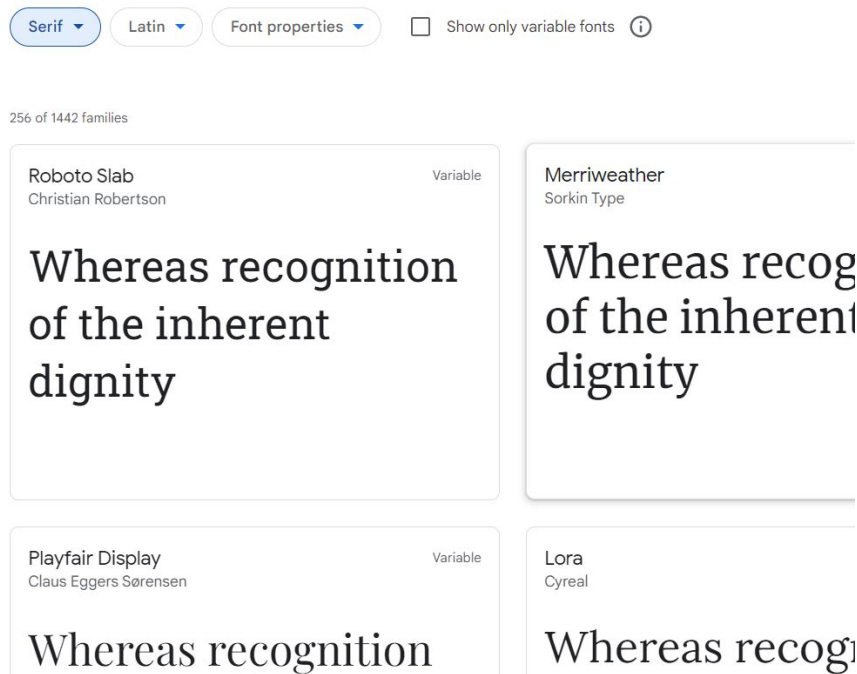
Tipografía: importación

Vamos a [Google Fonts](#) y podemos ver que (además de buscar una fuente en particular) podremos filtrar por categorías.



Tipografía: importación

Serif (Serifa): Son aquellas que tienen sus terminaciones estilizadas como la Times New Roman o las que vemos en este ejemplo.



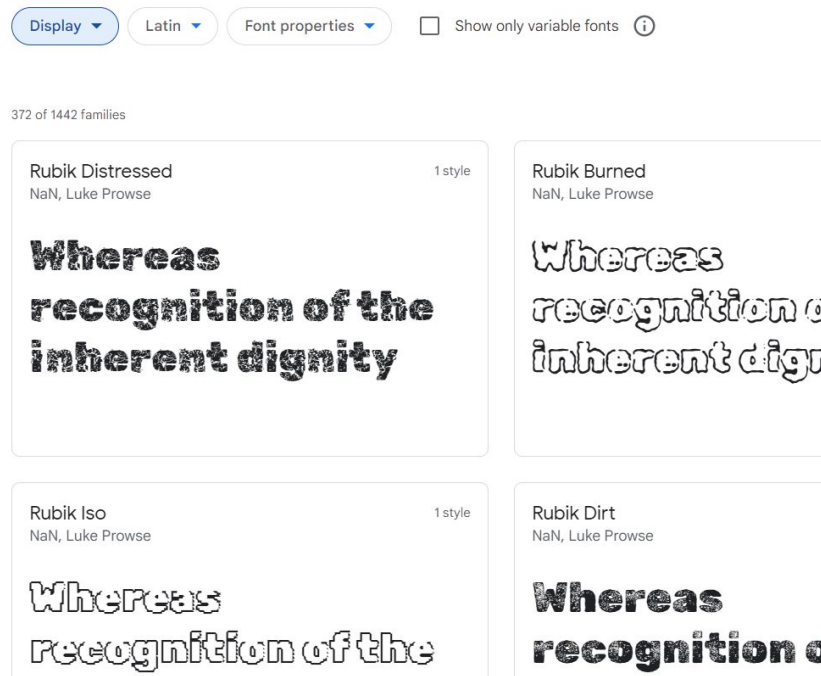
Tipografía: importación

Sans-Serif (Sin serifa): A diferencia de las anteriores no tienen adornos en sus terminaciones, como la Arial.



Tipografía: importación

Display: Tienen un diseño artístico, no se recomienda su uso para textos largos, sí para títulos o de presencia ocasional.



Tipografía: importación

Monospace: Cada letra tiene un espacio proporcional en dimensiones, donde cada carácter ocupará el mismo espacio que cualquier otro carácter.

Es el que usamos en nuestros editores de código.

Monospace ▾ Latin ▾ Font properties ▾ ☐ Show only variable fonts ⓘ

29 of 1442 families

Roboto Mono
Christian Robertson

Variable

Whereas
recognition of
the inherent
dignity

Inconsolata
Raph Levien

Whereas re
of the inh
dignity

Tipografía: importación

Una vez que seleccionamos una letra Google Fonts nos llevará a una página, si vamos a la parte inferior de ella encontraremos una extensa variedad de la fuente seleccionada, entre lo que veremos cambios como diferentes grosores y tamaños base.

n of the inherent dign Thin 100 ⊕

n of the inherent dig Thin 100 Italic ⊕

n of the inherent dig ExtraLight 200 ⊕

n of the inherent d ExtraLight 200 Italic ⊕

Tipografía: importación

Cuando seleccionamos una se nos abrirá una barra lateral, ahí seleccionaremos la segunda opción **@import** y usaremos los valores que se ven debajo:

To embed a font, copy the code into the `<head>` of your html



`<link>`



`@import`

```
<style>
@import url('https://fonts.googlea
pis.com/css2?family=Poppins&displa
y=swap');
</style>
```

CSS rules to specify families

```
font-family: 'Poppins', sans-seri
f;
```


Tipografía: importación

Seleccionamos todo lo que está dentro de la etiqueta *style* y lo pegamos en la parte superior de nuestra hoja de estilos.

To embed a font, copy the code into the `<head>` of your html



`<link>`



`@import`

```
<style>
```

```
@import url('https://fonts.googlea  
pis.com/css2?family=Poppins&displa  
y=swap');
```

```
</style>
```

CSS rules to specify families

```
font-family: 'Poppins', sans-seri  
f;
```

Tipografía: importación

Ahora copiaremos el atributo font-family a la clase que deseemos asignar esa fuente. Nos quedaría algo similar a esto:

```
le.css >  .clase
@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
.fuente-importada{
  font-family: 'Poppins', sans-serif;
}
```

To embed a font, copy the code into the <head> of your html

☐ <link> ☒ @import

```
<style>
@import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
</style>
```

CSS rules to specify families

```
font-family: 'Poppins', sans-serif;
```

Tipografía: importación en nuestro server

Para insertar una tipografía que tengamos en nuestros archivos haremos uso de @font-face en nuestro CSS, de esta manera:

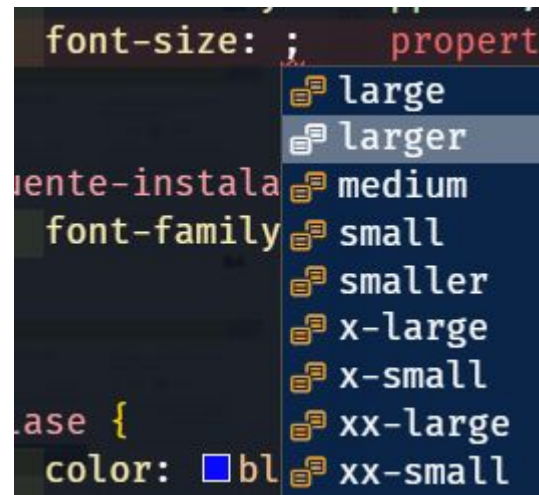
En **src:** insertamos la ruta del archivo font que tengamos en nuestro proyecto, y en **font-family** el nombre que deseamos darle, para luego llamarla así en la clase que deseamos tenerla.

```
@font-face {  
  font-family: nofex;  
  src: url('./src/fonts/Nofex.otf');  
}  
  
.fuente-instalada {  
  font-family: nofex;  
}
```

Tamaño de fuente

Usaremos el atributo font-size, tenemos varias formas de pasarles valor a la clase:

Podemos elegir tamaños predeterminados, como los que vemos en el ejemplo.



Tamaño de fuente

Tendremos dos tipos de medidas para asignarles, los clasificaremos en:

Medidas absolutas: El más popular es el px (píxel), también tenemos cm, pulgadas... Son medidas que siempre tendrán el mismo valor.

Medidas relativas: Son aquellas cuyas medidas varían según el elemento en que se encuentran, las más usadas son em, rem, vh y vw.

Tamaño de fuente

Las medidas **em** se verán afectadas por el font size del elemento donde se encuentran, si el tamaño del font de su contenedor es de 10px y proporcionamos a nuestro ítem un valor de 2em entonces su medida será equivalente a 20px ($2 * 10\text{px}$).

En el caso de las medidas **rem** el procedimiento es similar, con un pequeño cambio: El elemento de referencia no será su contenedor, sino el root(<html>), si este tiene un valor de 15px entonces la font con un tamaño de 2rem equivaldrá a 30px ($2 * 15\text{px}$).

Tamaño de fuente

Los valores **vh** y **vw** hacen referencia a viewport height y viewport width respectivamente, los mismos representan a la 1/100 parte del alto o ancho del dispositivo (según corresponda).

100vh es el alto total de la pantalla, y 100vw es el ancho total de ella.



Agrupamientos

Agrupamientos: Div y Span

Las etiquetas **<div>** y **** hacen referencia a agrupamientos de elementos HTML, son contenedores, pueden contener muchas etiquetas y proporcionar valor de clases a su contenido.

<div> Hace referencia a división, maneja sus unidades por bloque, esto quiere decir que de tener dos div consecutivos uno ocupará una 'línea' y el otro otra.

**** Esta etiqueta agrupa elementos y no divide por bloques (vertical), sino por lo que abarca en la línea (horizontal), si tenemos dos span consecutivos uno estará al lado del otro.

Atributos globales

Son esos atributos que podremos usar en cualquier etiqueta, sin ser exclusivo de alguna de ellas.

Unos ejemplos de ellos son el atributo class, hidden, id, ,draggable, contenteditable, accesskey, style, title.

**Especificidad
(Etiquetas semánticas)**

Etiquetas semánticas

De a poco estamos conociendo el alcance del poder que tiene CSS en nuestro navegador, y vemos que hay varios valores que ofrece HTML y que pueden ser sustituidos o incluso superados por nuestras clases.

Por este motivo nos enfocaremos en determinadas etiquetas que se destacarán en la semántica de nuestra página.

Etiquetas semánticas

Las etiquetas semánticas más fuertes que usaremos a partir de ahora serán:

<header> Para definir la cabecera de nuestro sitio, quizás el menú.

<nav> En nuestras secciones de enlaces, por lo general un menú.

<main> Donde se encontrará el cuerpo y toda la información principal del proyecto.

<section> Con la que manejaremos las secciones que puede tener nuestro sitio.

<article> Cuando hagamos referencias a artículos, sean informativos, productos, etc.

<aside> Cuando marquemos elementos que no están involucrados con la idea general de nuestro producto, como una publicidad

<footer> Para marcar al pie de página, o quizás también se puede encontrar dentro de una sección.

Etiquetas semánticas

Como podemos ver, estas etiquetas en HTML no proporcionan valor alguno a su contenido como lo hacen las etiquetas de estilos (negrita, cursiva) o de recursos (como `img`, `a`, `video`) sin embargo ayudan a tener un código fácil de leer (en caso de que debamos recurrir a él en un futuro), o que algún evento nos genere un error, ayudará a localizar el problema sin afectar o empeorar el mismo.

Esto no significa que algunas etiquetas dejarán de ser usadas por su equivalente CSS, a veces son un recurso más efectivo que estas últimas.



Iconos

Iconos

Ya hemos visto cómo poner el ícono de nuestra página web, sabemos que manejamos el formato .ico para ella, pero en el resto de nuestra web utilizaremos otros formatos que ya conocemos.

Entendemos por icono a todo signo (o en nuestro caso imagen) que nos referencie “algo” en nuestra mente, que nos represente un producto, un servicio, un objeto, algo.



Iconos

Cuando hagamos uso de ellos podremos usar cualquier formato de imagen, aunque en la actualidad hay un favoritismo por el formato SVG.

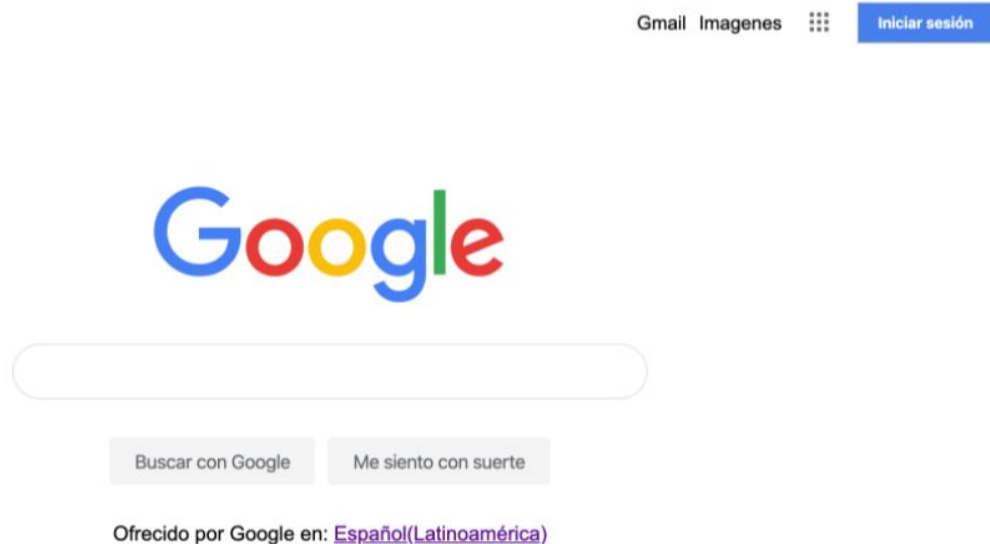
SVG (Scalable Vector Graphics) como su nombre lo indica son gráficos vectoriales, por lo que pueden ser redimensionados sin pérdida alguna de calidad.

Iconos

Podemos usar la etiqueta `img`, `iframe`, `embed`, `object`. Con `img` tenemos más libertades para manipular su tamaño.

Podemos crear un `svg` desde cero, usando la etiqueta `<svg>` e incluyendo dentro de ella otros tags como `line`, `circle`, `polygon`, aunque lo recomendable es usar un programa de diseño vectorial como Adobe Illustrator para acelerar nuestra productividad.

Ejercicio: Clonar la página de inicio de Google



Trata de replicar esta página inicial de Google!

Datos: Puedes usar imágenes para el logo, o buscar su brandbook y obtener su fuente, los colores que usan en valor hexadecimal, usar la consola del desarrollador para obtener ciertos colores de la pantalla!

Colombia

[Publicidad](#) [Negocios](#) [Sobre Google](#) [Cómo Funciona la Búsqueda](#)

[Privacidad](#) [Condiciones](#) [Preferencias](#)

Links de interés

Recursos de fuentes:

[Google Fonts](#)
[Fontdesk](#)

Herramienta para tipografías:

[FontJoy](#)

Colores:

[Adobe Color](#)

Construcción de paletas de colores:

[Paletton](#)

Recordá:

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**

Todo en el Aula Virtual.