Hack Academy®

Curso de Front-End

Clase 02

Agenda de la clase

Agenda

- Repaso.
- HTML: Elementos Block vs. Inline.
- CSS:
  - Sintáxis.
  - CSS dentro de HTML.
  - Ejemplos de CSS: Colores y tamaño de fuente.
  - Selectores: `element`, `class` e `id`.
  - Jerarquía de selectores.
- Ejercicios.

Repaso

Tips generales

¡Practiquen mucho!

¿Cómo se aprende
a tocar la guitarra?

Recuerden que no mandamos deberes, exámenes, pruebas, entregas, etc.
Pero eso no quiere decir que no practiquen por su cuenta. Lo ideal es conseguirse un proyecto propio (un sitio web para ustedes o para alguien más). Es inevitable notar la diferencia entre los que practican y los que no.

Tips generales

Por las dudas lo repetimos...

¡Practiquen mucho!

👉 Sugerimos que practiquen en sus casas aprox. **6 horas** semanales.

REPASO

Tips generales

Si su teclado físico tiene las teclas en inglés 🇺🇸...

Configurarlo como **U.S. International**, que sirve tanto para escribir en Inglés como en Español. Así no tienen que estar cambiando el idioma ni acordarse de memoria donde está cada carácter en el teclado 🤦.

👉 Video de configuración en Windows 10: https://youtu.be/OF5pjbDldMU.

REPASO

Estructura de una página web

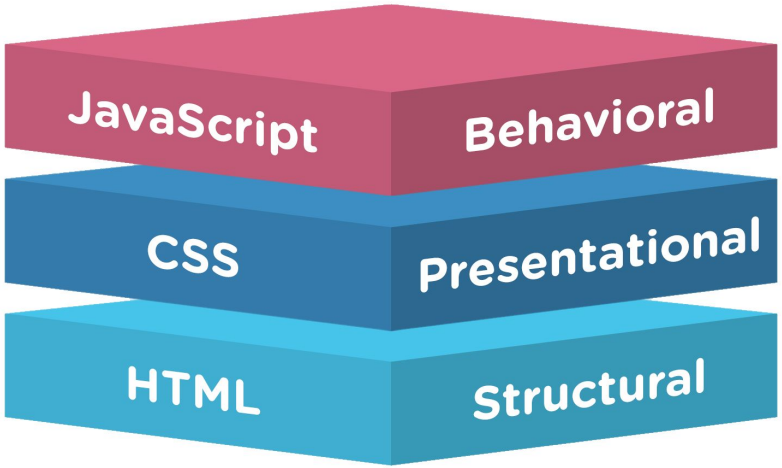
Contenido: Texto, imágenes, videos.

\+ **HTML**: Estructura + Semántica

\+ **CSS**: Presentación + Diseño

\+ **JS**: Interacción

= Página web



Nota: Esta es la estructura básica de una página desde el punto de vista de un desarrollador Front-End. Un desarrollador Back-End también consideraría, por ejemplo, el guardado (persistencia) de los datos.

REPASO

Estructura de una página web

Analogía con la construcción de una casa:

HTML

CSS

JavaScript

HTML → Qué

CSS → Cómo

“Sólo con el Bloc de Notas podríamos hacer un sitio web completo”

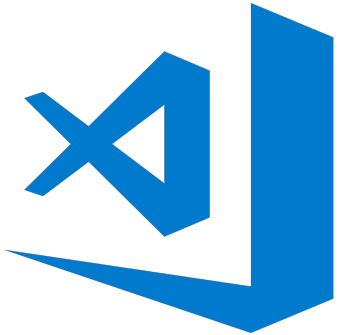
Pero hay que estar loco para hacerlo.

Bloc de Notas

+

Esteroides

=



Visual Studio Code

REPASO

¿Qué es una etiqueta (tag) HTML?

Son palabras clave escritas entre < y >.

Se abren y cierran.

```
<etiqueta></etiqueta>
```

Opcionalmente, pueden tener contenido y atributos.

```
<etiqueta atributo="valor">contenido</etiqueta>
```

REPASO

Estructura básica de una página HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Título de la página</title>
  </head>
  <body>
    <h1>Mi primer título</h1>
    <p>Mi primer párrafo.</p>
  </body>
</html>
```

Recordar que **VSC** incluye la funcionalidad de insertar una estructura HTML básica.

Sólo hay que escribir "!" y apretar Enter o Tab.

Componentes:

- **DOCTYPE**
  Indica la versión del HTML que se utilizará. En este caso se indica que se está usando HTML versión 5.
- **HTML**
- **HEAD**
  Información y atributos de la página.
- **BODY**
  Contenido "visible" de la página.

REPASO

Algunos HTML tags que vimos

```
<div id="mi-seccion">
    <h1>Este es un título importante</h1>
    <h2>Este es un subtítulo</h2>
    <p>
        Este es un <strong>párrafo</strong>.<br />
        Que tiene tres renglones.<br />
        <a href="pagina2.html">Este es un link a la Página 2</a>
    </p>
    <img src="image.jpg" alt="Descripción de la imagen" />
</div>
```

REPASO

HTML – Semántico, Accesible y Organizado

El HTML que escribe un desarrollador debería ser:

- Semántico: Debe tener significado. HTML se debe ocupar del *qué*, no del *cómo*.
- Accesible: Para que el sitio web pueda ser accedido por la mayor cantidad de personas. Particularmente hay que pensar en las personas no-videntes.
- Organizado: El código debe ser prolijo, organizado, fácil de leer y entender.

Ejemplo de semántica: con HTML se puede crear un párrafo `<p>` y luego con CSS cambiar el tipo y tamaño de letra para que *parezca* un título. Pero por más de que se parezca a un título, semánticamente hablando, no deja de ser un párrafo y, por lo tanto, no es correcto hacer esto. Este concepto es muy importante en HTML.

De hecho, sucede exactamente lo mismo en un editor de textos como Microsoft Word. La forma de crear un título en Word no es cambiándole el tamaño de letra a un párrafo sino creando un Encabezado (*Heading*).

REPASO

HTML Tips

- Cerrar todos los *tags* (excepto `<br>`, `<img>`, `<meta>` y algunos otros).
- Usar indentación correctamente (En VSC: *click* derecho y dar formato).
- Usar las minúsculas en los *tags*.
- Evitar espacios, tildes y eñes en los nombres de los archivos.

```
<DIV><ul><li>Contenido 1
</li><LI>Contenido 2
</LI></UL></div>
```

PÉSIMA PRÁCTICA

```
<div>
    <ul>
        <li>Contenido 1</li>
        <li>Contenido 2</li>
    </ul>
</div>
```

BUENA PRÁCTICA

Un código prolijo habla muy bien de un programador.

Por las dudas repetimos...

Cuiden la indentación de sus archivos.

IMPORTANTE

Uso de Slack / Microsoft Teams

- Usar su nombre completo + foto de perfil.
- Escribir mensajes en los canales correspondientes.
- Compartir código usando los botones adecuados. En algunos casos puede ser más útil compartir un ZIP con todo el código de un proyecto.
- Escribir las consultas en un único mensaje. Evitar escribir varios mensajes para una misma consulta. Ej: No escribir *“Hola”*, *“¿Cómo están?”*, *“Tengo una consulta”*, *“No logré hacer funcionar…”*, en 4 mensajes diferentes.
- Respetar los hilos (*threads*) de comunicación al responder un mensaje.

HTML: Elementos Block vs. Inline

Elementos de tipo Block vs. Inline

Verlo con las **Developer Tools** en ha.dev.

Por defecto, todo elemento en HTML tiene un tipo de visualización (*display*) predeterminado. Los dos más comunes son:

Block: es un elemento que ocupa todo el ancho de su elemento padre (contenedor), generando un salto de línea antes y después de sí mismo.

Elementos *block* más usados: `<h1>`, `<h2>`, `<div>`, `<p>`, `<ul>`, `<ol>`, `<form>`.

```
Este es un elemento <div>.
```

Inline: es un elemento que ocupa sólo el ancho que precisa y no genera saltos de línea.

Elementos *inline* más usados: `<a>`, `<em>`, `<strong>`, `<button>`, `<span>`.

```
Este es un elemento <span> dentro de un párrafo.
```

CSS

¿Qué es CSS?

- CSS = Cascading Style Sheets.

- CSS es un lenguaje de hojas de estilo que permite darle **estilo** a los elementos de una página. Fue creado en 1996.

- CSS se encuentra embebido en el HTML, pero no es HTML. La gente confunde esto habitualmente. CSS no es HTML. Repitan: CSS no es HTML.

Entonces, ¿qué podemos hacer con CSS?

Con CSS se puede hacer *de todo*

Subrayar un texto.

Darle color a un texto, **hacerlo negrita,** *hacerlo cursiva.*

Tamaño y la tipografía (Recomendación: nunca usar comic sans)

Posicionar un texto hacia la derecha

Cambiar el color de fondo

Cambiar el borde de un elemento (color, ancho, estilo)

CSS – Documentación

- MDN: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps
- MDN (español): https://developer.mozilla.org/es/docs/Learn/CSS/First_steps
- W3 Schools: http://www.w3schools.com/css

Empiecen a acostumbrarse a consultar la documentación de cada tecnología.

En general, suelen ser mejores las versiones en **inglés**.

CSS – Links interesantes

Con CSS versión 3 se pueden hacer varias cosas “locas”. Si bien no entran en el temario del curso, aquí se pueden ver algunos ejemplos:

- Flip Card: https://desandro.github.io/3dtransforms/examples/card-01.html.
- Girar Cubo: https://desandro.github.io/3dtransforms/examples/cube-02-show-sides.html.
- Gallo que camina: https://codepen.io/judag/pen/zxKVQR.

Conviene señalar que muchas veces este tipo de efectos requieren utilizar propiedades de CSS y HTML que no son soportadas por todos los navegadores.

Ejemplo de código CSS

```css
body {
    color: blue;
    background-color: grey;
}
```

Este código le indica al `body` que el color de texto debe ser azul y que el color de fondo debe ser gris.

Sintaxis de CSS

CSS consiste de una serie de reglas de estilo. Cada regla está conformada por un **selector** y declaraciones de pares "**propiedad-valor**" (o clave-valor).

```
selector {
    property: value;
}
```

Sintaxis:

El selector encierra las propiedades entre llaves `{` y `}`.

Los pares de propiedad-valor se separan por `:` y terminan con un `;`.

```
body {
    color: blue;
    background-color: grey;
}
```

En este ejemplo el selector es el *tag* `body` y se le asignaron 2 propiedades CSS.

La primera establece el color del texto en el `body`.

La segunda establece el color de fondo del `body`.

CSS dentro de HTML

CSS se puede *embeber* dentro del HTML de **diversas formas**.
Hoy vamos a aprender dos:

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

```
<body style="color:red;">
    <h1>
        Esto es un título
    </h1>
    <p>
        Esto es un párrafo.
    </p>
</body>
```

Ambas producen el mismo resultado.

1. CSS embebido en tags `<style>`

Una forma de incorporar CSS en una página web es poniéndolo dentro de los *tags* `<style>`.
Los *tags* `<style>` se suelen colocar dentro del *tag* `<head>`.

```
<!DOCTYPE html>
<html>
    <head>
        ...
        <style>
            body {
                background-color: lightgrey;
            }
            h1 {
                color: red;
            }
        </style>
    </head>
    <body>
        ...
    </body>
</html>
```

2. CSS embebido directo en tags HTML

Otra forma es incluir el CSS directamente dentro de los *tags* HTML usando el atributo `style`:

```
<body style="background-color:lightgrey;">
    <h1 style="color:red;">
        Esto es un título
    </h1>
    <p style="color:yellow;">
        Esto es un párrafo.
    </p>
</body>
```

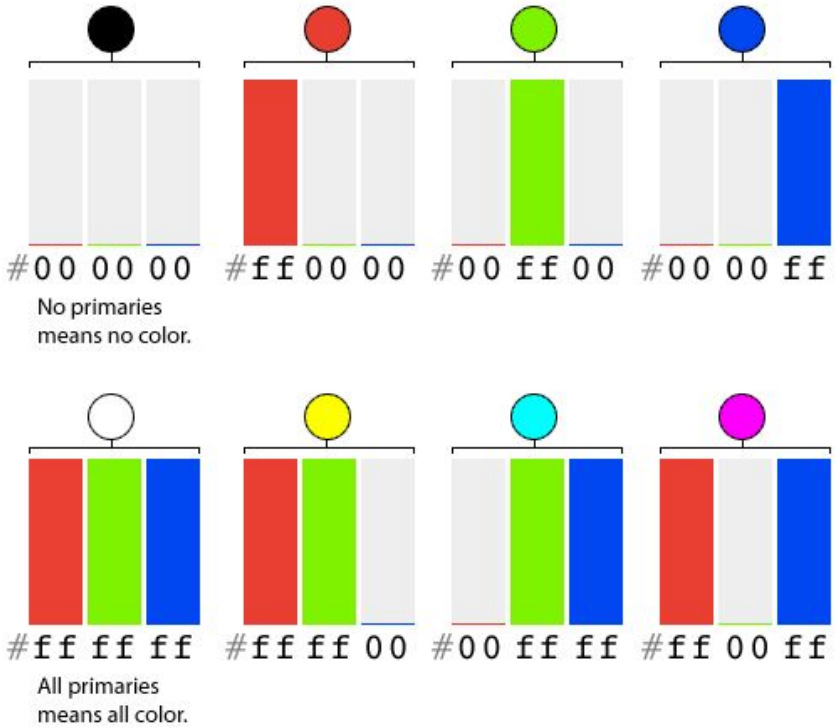
CSS – Colores

Propiedades CSS – Colores (1/2)

A veces, las propiedades pueden tener distintos tipos de valores. Por ejemplo, el color puede tomar distintos tipos de valores.

```
<body style="background-color:lightgrey">

    <h2 style="color:blue; background-color:pink;">Esto es un título</h2>

    <h2 style="color:#AA22FF">Otro título</h2>

    <h2 style="color:rgb(0,255,255)">¿Y esto qué es?</h2>

</body>
```

Propiedades CSS – Colores (2/2)



¿Cómo se forman los colores hexadecimales?

Probar este color picker: http://htmlcolorcodes.com/color-picker/.

CSS – Tamaño de fuente

Propiedades CSS – Tamaño de fuente (1/3)

`font-size` (usando `px`)

La unidad `px` permite establecer el tamaño del texto en **píxeles** (la misma unidad que se usa para medir el tamaño de imágenes y otros elementos).

Al usar la unidad `px`, el tamaño del texto queda fijo (independiente del tamaño de los demás textos de la página o del tamaño por defecto del *browser*). Es la unidad más fácil de usar pero no siempre la más recomendada. De hecho, en general, por un tema de accesibilidad no aconsejamos usar `px` para `font-size`.

```
<p style="font-size:20px;">

    Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget
    dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes,
    nascetur ridiculus mus.

</p>
```

Propiedades CSS – Tamaño de fuente (2/3)

`font-size` (usando `em`)

Esta unidad permite establecer el tamaño de un texto de forma **relativa** al tamaño del texto del elemento actual. Ejemplo: `2em` significa que el texto será 2 veces más grande que el texto actual.

`font-size` (usando `%`)

Similares características a `em`, pero se especifica como un porcentaje. Puede ser utilizado en conjunto con otras unidades.

En general, `1em = 100%`.

⚠️ Por un tema de complejidad de uso, tampoco aconsejamos usar `em` y `%` para `font-size`. La unidad de medida sugerida es `rem`, que se explica en la siguiente diapositiva.

Propiedades CSS – Tamaño de fuente (3/3)

`font-size` (usando `rem`)

Esta unidad permite establecer el tamaño de un texto de forma **relativa** al tamaño del texto establecido en el elemento ***root***, es decir, en el elemento `<html>`, que en general es de 16px (en la mayoría de los *browsers*).

Esto hace que `rem` sea más “controlable” que `em` ya que todos los tamaños dependen únicamente del tamaño de texto definido en el elemento *root*, en lugar de depender del tamaño de texto definido en el elemento en cuestión.

Nota: `rem` sólo se puede usar a partir de Internet Explorer 9.

👉 Esta es la unidad que recomendamos usar para textos.

Otras propiedades CSS muy utilizadas

```
selector {
    line-height: 10px;
    text-align: center;
    text-decoration: underline;
    border: 5px solid red;
    background-image: url("img/fondo.gif");
    height: 30px;
    width: 100%;
}
```

Familiarizándote con CSS

- Inspeccionar los estilos de otros sitios utilizando los *Developer Tools*.
  - Habilitar/inhabilitar atributos que te resulten curiosos.
  - Modificar el valor de los atributos (o agregar nuevos) y observar lo que pasa.
  - Copiar estilos que te gusten y usarlos en tu sitio.
- ¡Experimentar, experimentar y experimentar!

1. Escribir un estilo CSS.
2. Guardar el archivo.
3. Recargar página en el *browser*.

Repetir este proceso.

Ejercicio

Ejercicio – Replicar esta página (1/2)

Explicación en la siguiente *slide*.

ejercicio.html

Ir

Primer ejercicio de CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse.

Ejercicio – Replicar esta página (2/2)

- Lo primero que aparece en la página es un título `<h1>` de color negro.
- Luego hay tres párrafos `<p>`.
- El título y los dos primeros párrafos tienen borde violeta.
- Todos los párrafos tienen tipo de letra Arial y texto centrado.
- Cada párrafo tiene un color particular.
- El color de fondo de la página es celeste.
- Por ahora, no se preocupen por los espaciados.

CSS – Selectores

Selectores CSS

```
selector {
    property: value;
}
```

Los selectores se utilizan para determinar sobre qué elemento del HTML se aplicarán los estilos.

Básicamente, hay 4 tipos de selectores:

- element
- class
- id
- pseudo classes

Tipos de selectores: `element`

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

Esto selecciona todos los elementos `p` en el documento.

```
<p>Lorem impsum</p>
```

IMPORTANTE

Tipos de selectores: `class`

Un elemento puede tener un único `id` pero múltiples `class`, lo cual permite tener gran flexibilidad si se usa correctamente.

```css
.introduccion {
    background-color: blue;
}
.descripcion {
    color: red;
}
```


```html
<p class="introduccion descripcion">Lorem impsum</p>
<p class="introduccion">Lorem impsum</p>
<p class="descripcion">Lorem impsum</p>
```

IMPORTANTE

Tipos de selectores: `id`

```css
#pepito {
    color: red;
}
```

En programación (en inglés) es común usar nombres de ejemplo como `foo` o `bar`.

Con el numeral (#) se le indica a CSS que un selector es de tipo `id`.

Esto selecciona el (único) elemento HTML que tenga `id=pepito`.

No importa qué elemento (etiqueta) HTML sea. En este ejemplo se trata de un `p`.

```html
<p id="pepito">Lorem ipsum</p>
```

Los `id` son identificadores únicos (sólo puede haber un elemento por página con determinado nombre).

Nota: Recordar que en la clase anterior vimos de usar el `id` para darle nombre a una sección de una página y luego vimos cómo crear un link a la misma.

Tipos de selectores: `class` (especificando el element)

```
div.introduccion {
    background-color: blue;
}
```

Esto selecciona sólo los elementos de tipo `div` con `class=introduccion`.

Tipos de selectores: `id` (especificando el element)

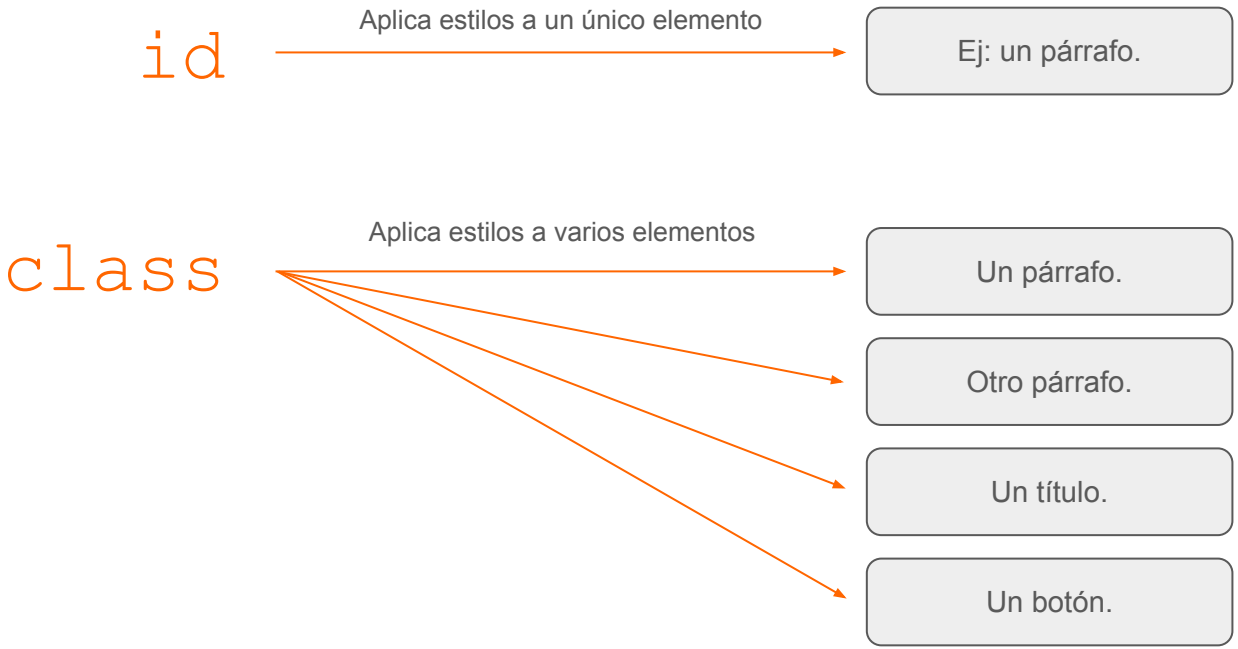
```
p#pepito {
    color: red;
}
```

Esto suele ser poco usado.

Esto selecciona el elemento `p` con el `id=pepito`.

```
<p id="pepito">Lorem ipsum</p>
```

Tipos de selectores: `id` vs `class`



Tipos de selectores: pseudo classes

```
a:link {...}

a:visited {...}

a:hover {...}

a:active {...}

a:focus {...}
```

Esto le da estilos al elemento `a` según su estado.

Jerarquía de selectores

(primera aproximación)

Jerarquía de selectores CSS (primera aproximación)

Reglas:

- Los `id` son más específicos que los `class` y
- `class` es más específico que `element`.
- Si dos reglas son igual de específicas, la última gana.

```css
h2 {
    color: blue;
}
h2 {
    color: red;
}
```

El elemento `<h2>` queda de color **rojo**.

```css
#unId {
    color: blue;
}
.unaClase {
    color: red;
}
```

El elemento `<p id="unId" class="unaClase">` queda de color **azul**.

Tips

Algunos tips

- Convención de nombres para `id` y `class`:
  - Utilizar nombres que describan el contenido, no la presentación.
    Ej: Es preferible el nombre `texto-importante` que `texto-rojo`.
  - Utilizar minúsculas y guiones (`-`) cuando sea necesario. Ayudará la legibilidad.
  - Utilizar guiones para mostrar que una clase o un ID es parte de otra cosa (ej. `cabezal`, `cabezal-logo`, `cabezal-menu`).
- Es posible hacer comentarios en CSS y es recomendable hacerlo.

```
/* ............ */
```

CSS – Varios valores en la misma propiedad

A veces, se le pueden asignar varios valores a la misma propiedad, separados por un espacio. Por ejemplo, esto sucede con la propiedad `font` (ver docs).

```css
body {
    color: #4286f4; /* Azul */
    background-color: grey;
    font: italic 1.4rem sans-serif;
}
```

```css
body {
    color: #4286f4; /* Azul */
    background-color: grey;
    font-style: italic;
    font-size: 1.4rem;
    font-family: sans-serif;
}
```