

# Interfaz de usuario

There is a design principle called separation of concerns that aims to create scalable, more easily maintainable coding architectures. This principle guides us to keep our structure (HTML) separate from our presentation (CSS). As layout is presentational, we will generally want to use CSS to position our elements.

# HTML - Hyper Text Markup Language

Hyperlink:  
texto que provee acceso  
a otro texto con un link

Se encarga de la  
presentación  
de la información

- Describe la *estructura del contenido* de un sitio web utilizando un *lenguaje de marcado*
- Los elementos HTML, representados por etiquetas, son los bloques constructores
- Los navegadores interpretan el contenido dentro de los elementos HTML

# ¿Por qué el contenido necesita estructura?

El niño cuenta ovejas en la noche para poder dormir. **Estructura gramática**

Sujeto Verbo Predicado

```

<!DOCTYPE html>
<html>
<head>
  <title>Blog para padres</title>
  <meta name="keywords" content="Blog opinión padres hijos tecnicas consejos ayuda paternidad">
  <meta name="author" content="blogparapadres.com">
  <meta name="description" content="Blog para padres es un compilado de consejos prácticos para ayudar en cualquier momento de la paternidad. Actualizado continuamente con opiniones de expertos.">
</head>
<body>
  <div class="ArticleContent">
    <div class="tweet-selection">
      
      <p>"El niño cuenta ovejas en la noche para poder dormir."</p>
    </div>
    <div class="copy-container">
      <h1>Técnicas para dormir</h1>
      <h3>Cuentos de cuna</h3>
      <p>...</p>
      <p>...</p>
      <h3>Cansancio físico</h3>
      <p>...</p>
      <p>...</p>
      <p>...</p>
      <h3>Alimentación balanceada</h3>
      <p>...</p>
    </div>
  </div>
</body>
</html>

```

Jerarquía de información:

- Fácil lectura, resalta lo más importante
- Accesibilidad, para navegar correctamente por el contenido. No solo por un navegador sino por otros programas cómo un lector de pantalla (screen reader). <https://webaim.org/techniques/alttext/> y <https://www.codecademy.com/articles/how-to-setup-screen-reader>
- Ser encontrado por sistemas de búsqueda

# Elementos HTML

## etiqueta HTML + contenido + etiqueta HTML

<code>&lt;!DOCTYPE html&gt;</code>	- La declaración <code>&lt;!DOCTYPE html&gt;</code> define un documento HTML5. Aparece una vez al principio del documento y representa el tipo de documento. Ayuda al explorador a mostrar el contenido correctamente.
<code>&lt;html&gt;&lt;/html&gt;</code>	- El elemento <code>&lt;html&gt;&lt;/html&gt;</code> es el elemento raíz de una página HTML, un documento HTML debe empezar y terminar con esta etiqueta.
<code>&lt;head&gt;&lt;/head&gt;</code>	- El elemento <code>&lt;head&gt;&lt;/head&gt;</code> contiene información metadata del documento y no se muestra en el navegador. Puede contener: scripts, estilos CSS, metadata, etc.
<code>&lt;title&gt;&lt;/title&gt;</code>	- El elemento <code>&lt;title&gt;&lt;/title&gt;</code> especifica un título
<code>&lt;body&gt;&lt;/body&gt;</code>	- El elemento <code>&lt;body&gt;&lt;/body&gt;</code> contiene el contenido visible de la página
<code>&lt;h1&gt;&lt;/h1&gt;</code> al <code>&lt;h6&gt;&lt;/h6&gt;</code>	- Los elementos del <code>&lt;h1&gt;&lt;/h1&gt;</code> al <code>&lt;h6&gt;&lt;/h6&gt;</code> definen encabezados. Van del más importante (1) al menos importante (6): <a href="https://www.w3.org/WAI/tutorials/page-structure/headings/">https://www.w3.org/WAI/tutorials/page-structure/headings/</a>
<code>&lt;p&gt;&lt;/p&gt;</code>	- El elemento <code>&lt;p&gt;&lt;/p&gt;</code> define un párrafo
<code>&lt;a href="url"&gt;&lt;/a&gt;</code>	- El elemento <code>&lt;a href="url"&gt;&lt;/a&gt;</code> define un link
<code>&lt;img src="file.jpg"&gt;</code>	- El elemento <code>&lt;img src="file.jpg"&gt;</code> define una imagen. <a href="https://internetingishard.com/html-and-css/links-and-images/#absolute-relative-and-root-relative-links">https://internetingishard.com/html-and-css/links-and-images/#absolute-relative-and-root-relative-links</a>
<code>&lt;br&gt;</code>	- <code>&lt;br&gt;</code> es un elemento vacío que define un salto de línea
<code>&lt;div&gt;</code>	- El elemento <code>&lt;div&gt;</code> es utilizado como un contenedor para otros elementos HTML

“There is often a trade-off between **complexity** and either **readability** or **targetability**. As developers, we want to find the right balance between creating enough coherent structure to enhance readability and targetability without venturing into *divitus* territory”

– Codeacademy y Chris Coyier

<https://css-tricks.com/css-beginner-mistakes-1/>



# IDENTACIÓN

mantenibilidad, trabajo colaborativo y legibilidad

<https://validator.w3.org/>

# Atributos HTML

- Proveen información adicional a los elementos HTML
- Se identifican por pares de nombre/valor



<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>This is a heading</h1>

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

<p>This is another paragraph.</p>

</body>

</html>

# Hojas de estilo

# CSS - Cascading Style Sheets



**Cascade:**  
Las reglas de estilo  
se propagan hacia abajo

- Lenguaje para *estilar* el contenido HTML de un sitio web
- Las reglas de estilo de mayor precedencia sobre escriben las reglas de menor precedencia

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

# Archivo .css

- CSS puede seleccionar elementos HTML usando la etiqueta del elemento
- Los ids son el selector más específico en CSS, seguidos por clases y etiquetas
- Se pueden asignar estilos encadenando selectores de elementos que cumplan condiciones específicas. Ej: solo elementos *h1* que además cumplan con tener una clase *blog-title*
- Se pueden asignar estilos a elementos anidados y serán aplicados solo a ellos. Ej: elementos (*li*) de una lista ordenada que a su vez tenga asignada la clase *hobbies*

```
h1 {
  color: maroon;
}

table, th {
  border: 5px solid red;
}

.title {
  color: teal;
}

.uppercase {
  text-transform: uppercase;
}

#article-title {
  font-family: cursive;
  text-transform: capitalize;
}

.publish-time {
  color: gray;
}
```

```
h2.destination {
  font-family: cursive;
}

.description h5 {
  color: teal;
}

h5 {
  color: rebeccapurple!important;
}

h5, p {
  font-family: Georgia;
}
```

# Flexbox layout

- *Flex container* dicta una estructura (*layout*) para el contenido parecido a bloques
- Forma un bloque contenedor para el contenido
  - *display: flex*; establece el contexto para el formato de flex
  - *flex-direction* Determina la dirección (principal) en la que se muestran los elementos: row, column
  - *flex-wrap* determina si el contenedor es para una o múltiples líneas: wrap, nowrap
  - *justify-content* alinea los elementos en el eje principal: flex-start, flex-end, center, space-between, space-around, space-evenly
  - *align-items* alinea los elementos en el eje secundario (perpendicular a la dirección del contenedor): flex-start, flex-end, center, baseline, stretch

<https://yoksel.github.io/flex-cheatsheet/>

**“UI designers have the opportunity and resources to educate their users in a way that can be enjoyable and memorable, immersing them in information instead of simply letting them skim through the site looking for what they need. This gives the user a reason—and desire—to learn more, allowing them to continue exploring and playing.”**

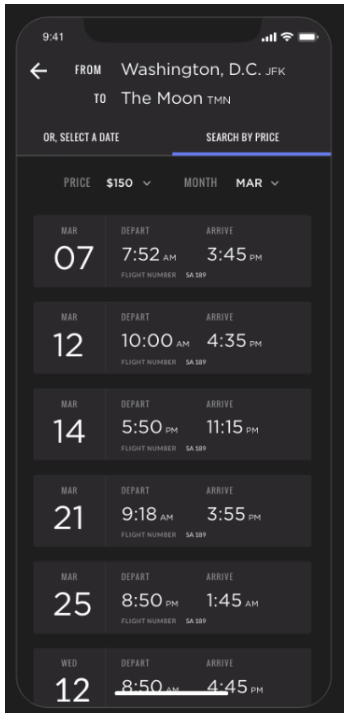
*–Corinne Nicewick • Jan 9, 2019*

**“It’s important to remember that beautiful websites and apps aren’t always engaging. The user may come in, absorb the beauty, but find nothing else worth sticking around for. Or, worse, leave more confused than they were when they started.”**

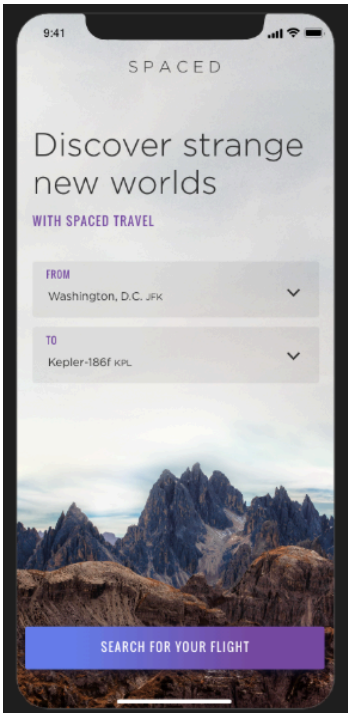
*–Corinne Nicewick • Jan 9, 2019*



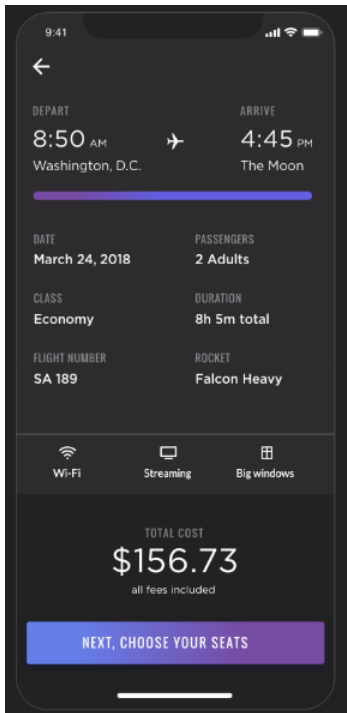
# Laboratorio 1



Opción 1



Opción 2



Opción 3