TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

Guía Práctica para Estudiantes: Diseño Web con CSS

Objetivo

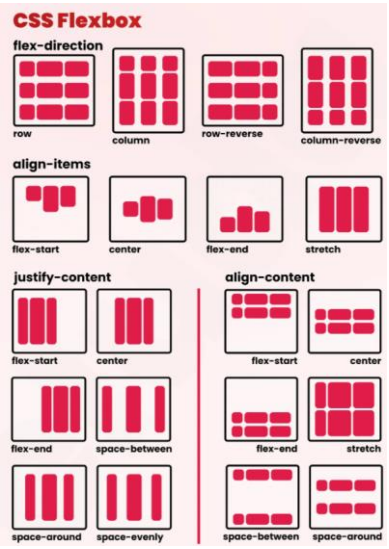
Aprender los conceptos fundamentales para diseñar y construir páginas web modernas y responsivas.

Introducción

A continuación, te explicaré cómo usar **Flexbox**, **CSS Grid**, **Media Queries**, **Transiciones** y **Animaciones** para hacer que tus sitios web sean más funcionales y atractivos. Cada sección incluirá una explicación detallada con ejemplos prácticos para que puedas experimentar y aplicar los conceptos.

1. Layout con Flexbox

¿Qué es Flexbox?



Es un sistema de diseño en CSS que facilita organizar y alinear elementos dentro de un contenedor la cual no puedes trabajar con filas y columnas al mismo tiempo. Funciona principalmente con **dos partes**:

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

El contenedor principal

- Es el elemento que contiene otros elementos dentro de él.
- Para que se active Flexbox, se le aplica la propiedad **display: flex;** en CSS.

Los elementos hijos

- Son los elementos dentro del contenedor.
- Flexbox permite **alinear**, **ordenar** y **distribuir** estos elementos de manera eficiente, tanto horizontal como verticalmente.

Estructura básica de Flexbox

El sistema Flexbox trabaja con un contenedor y sus elementos hijos. El contenedor se define con **display: flex;,** y los elementos hijos se alinean dentro de él.

Propiedades de contenedor Flexbox:

Descripción	Definición
display: flex;	Define el contenedor como un contenedor flexible
flex-direction	Define la dirección de los elementos dentro del contenedor. Puede ser `row` (por defecto, los elementos se alinean horizontalmente) o `column` (los elementos se alinean verticalmente).
justify-content	Alinea los elementos a lo largo del eje principal (horizontal si `flex-direction` es `row`).
align-items	Alinea los elementos a lo largo del eje transversal (vertical si `flex-direction` es `row`).
align-self	Permite que un elemento dentro del contenedor flex se alinee de forma diferente a los demás elementos.

Ejemplo de Flexbox:

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

En este ejemplo:

```
<!DOCTYPE html>
<html lang="es"> <!-- Declaración del tipo de documento y especificación del lenguaje español -->
<head>
  <meta charset="UTF-8"> <!-- Codificación de caracteres para admitir símbolos especiales y acentos -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!-- Hace que la página sea responsiva en dispositivos móviles -->
  <title>Ejemplo de Flexbox</title> <!-- Título de la página que se muestra en la pestaña del navegador -->

  <!-- Estilos internos de CSS -->

     <style>
    /* Contenedor principal que utiliza Flexbox */
    .contenedor {
      display: flex; /* Convierte el contenedor en un contenedor flexbox */
      flex-direction: column; /* Alinea los elementos hijos en una columna vertical */
      justify-content: space-between; /* Distribuye el espacio entre los elementos */
      align-items: center; /* Centra los elementos horizontalmente en el contenedor */
      height: 200px; /* Altura del contenedor */
      background-color: #f0f0f0; /* Color de fondo gris claro */
      padding: 100px; /* Espaciado interno alrededor del contenedor */
      gap: 20px; /* Espacio entre los elementos hijos */
    }

    /* Estilo de cada caja dentro del contenedor */
    .caja {
      background-color: #3498db; /* Color de fondo azul */
      color: white; /* Color del texto blanco */
      padding: 20px; /* Espaciado interno dentro de la caja */
      width: 100px; /* Ancho fijo de la caja */
      height: 100px; /* Altura fija de la caja */
      text-align: center; /* Centra el texto horizontalmente */
    }
  </style>
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

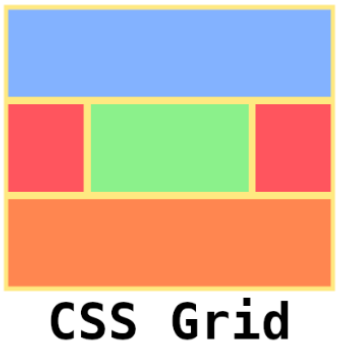
```
</head>
<body>
  <!-- Contenedor principal con las cajas -->
  <div class="contenedor">
    <div class="caja">Caja1</div> <!-- Primera caja -->
    <div class="caja">Caja2</div>
    <div class="caja">Caja3</div>
  </div>
</body>
</html>
```

- Las cajas dentro del contenedor `.contenedor` están alineadas horizontalmente con `justify-content: space-between`, lo que les da espacio entre ellas.
- Se alinean verticalmente en el centro con `align-items: center`.

---

2. Layout con CSS Grid

¿Qué es CSS Grid?



Es otro modelo de diseño que permite crear layouts bidimensionales (tanto filas como columnas al mismo tiempo). A diferencia de Flexbox, que es más adecuado para layouts unidimensionales, Grid permite tener un control total sobre el espacio en filas y columnas.

Estructura básica de CSS Grid

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

El sistema Grid se basa en un contenedor definido con `display: grid;` y luego se utilizan propiedades para definir el número de filas, columnas y el espacio entre los elementos.

Propiedades principales de CSS Grid:

Descripción	Definición
display: grid;	Define el contenedor como una cuadrícula.
grid-template-columns	Define el número y el tamaño de las columnas.
grid-template-rows	Define el número y el tamaño de las filas.
gap	Establece el espacio entre las filas y columnas.
grid-column y grid-row	Permiten que un elemento ocupe varias columnas o filas.

Ejemplo de CSS Grid:

```
<!DOCTYPE html>
<html lang="es"> <!-- Declaración del tipo de documento HTML5 y el idioma español -->

<head>
  <meta charset="UTF-8"> <!-- Configuración de la codificación de caracteres para admitir acentos y caracteres especiales -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!- - Hace que el sitio sea responsivo para dispositivos móviles -->
  <title>Ejemplo de CSS Grid</title> <!-- Título de la página visible en la pestaña del navegador -->

  <style>
    /* Contenedor principal con CSS Grid */
    .contenedor-grid {
      background-color: #d6d7d1;
      display: grid; /* Convierte el contenedor en una cuadrícula (grid) */
      grid-template-columns: repeat(3, 1fr); /* Define 3 columnas de igual tamaño usando fracciones */
      gap: 20px; /* Establece un espacio de 10px entre filas y columnas */
      padding: 10px; /* Espaciado interno entre el borde del contenedor y las celdas */
    }

    /* Estilo general para cada caja dentro del contenedor */
```

```
    .caja {
      background-color: #2ecc71; /* Color de fondo verde */
      color: white; /* Color del texto blanco */
      padding: 20px; /* Espaciado interno para aumentar el tamaño de las
cajas */
      text-align: center; /* Centra el texto horizontalmente */
    }
  </style>
</head>

<body>
  <!-- Contenedor principal que utiliza CSS Grid -->
  <div class="contenedor-grid">
    <div class="caja">Caja 1</div> <!-- Primera caja -->
    <div class="caja">Caja 2</div> <!-- Segunda caja -->
    <div class="caja">Caja 3</div> <!-- Tercera caja -->
    <div class="caja">Caja 4</div> <!-- Cuarta caja -->
    <div class="caja">Caja 5</div> <!-- Quinta caja -->
    <div class="caja">Caja 6</div> <!-- Sexta caja -->
  </div>
</body>
</html>
```

En este ejemplo:

- Usamos `grid-template-columns: repeat(3, 1fr);` para crear una cuadrícula con 3 columnas de igual tamaño.
- Los elementos `.caja` se distribuyen automáticamente en las filas y columnas de la cuadrícula.

---

3. Diseño Responsivo y Media Queries

¿Qué es el Diseño Responsivo?

El diseño responsivo se refiere a crear páginas web que se adaptan a diferentes tamaños de pantalla, como móviles, tabletas y escritorios.

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

Media Queries

Las Media Queries permiten que el diseño se ajuste según las características del dispositivo. Se utilizan para aplicar estilos específicos en función de la **anchura** (o altura) de la ventana del navegador.

Sintaxis básica:

```
@media tipo-dispositivo and (condición) {
  /* Estilos CSS aplicados cuando se cumple la condición */
}
@media screen and (max-width: 768px) {
  /* Estilos para pantallas con un ancho máximo de 768px */
}
```

Ejemplo de Media Queries:

```html
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Diseño Responsivo</title>

  <style>
    /* Estilo general del cuerpo de la página */
    body {
      font-family: Arial, sans-serif; /* Define la fuente principal del
texto */
    }

    /* Contenedor principal que utiliza Flexbox */
    .contenedor {
      display: flex; /* Convierte el contenedor en un contenedor Flexbox */
      flex-wrap: wrap; /* Permite que los elementos hijos salten a una nueva
línea si no caben */
      gap: 10px; /* Define un espacio de 10px entre los elementos hijos */
    }

    /* Estilos básicos para las cajas */
    .caja {
      background-color: #e74c3c; /* Color de fondo rojo */
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

```
      color: white; /* Color del texto blanco */
      padding: 20px; /* Espaciado interno dentro de cada caja */
      width: 100%; /* Por defecto, las cajas ocupan el 100% del ancho del
contenedor */
      text-align: center; /* Centra el texto horizontalmente dentro de la
caja */
    }

    /* Media query para pantallas con un ancho mínimo de 768px (pantallas
más grandes) */
    @media screen and (min-width: 768px) {
      .caja {
        width: 30%; /* Las cajas ocupan el 30% del ancho del contenedor en
pantallas grandes */
      }
    }

    /* Media query para pantallas con un ancho máximo de 768px (pantallas
pequeñas) */
    @media screen and (max-width: 768px) {
      .caja {
        width: 100%; /* Las cajas ocupan el 100% del ancho del contenedor en
pantallas pequeñas */
      }
    }
  </style>
</head>

<body>
  <!-- Contenedor principal que incluye las cajas -->
  <div class="contenedor">
    <div class="caja">Caja 1</div> <!-- Primera caja -->
    <div class="caja">Caja 2</div> <!-- Segunda caja -->
    <div class="caja">Caja 3</div> <!-- Tercera caja -->
  </div>
</body>

</html>
```

En este ejemplo:

- Cuando la pantalla es más ancha que 768px, las cajas ocupan el 30% del ancho de la pantalla.
- Cuando la pantalla tiene un ancho menor a 768px, las cajas ocupan el 100% del ancho disponible.

---

4. Transiciones y Animaciones

¿Qué son las Transiciones y Animaciones en CSS?

Las **transiciones** permiten que los cambios en los estilos se realicen suavemente a lo largo del tiempo, mientras que las **animaciones** permiten crear movimientos o efectos complejos y repetitivos.

Transiciones

Se activan cuando hay un cambio en una propiedad CSS, como cuando se pasa el ratón sobre un elemento. La transición especifica el tiempo que debe tomar ese cambio.

Sintaxis básica:

```
.elemento {
  transition: propiedad tiempo tipo-de-velocidad;
}
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

Ejemplo de Transición:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Transición CSS</title>
  <style>
    /* Estilos para la caja */
    .caja {
      width: 200px;
      height: 200px;
      background-color: #3498db;
      transition: all 0.5s ease; /* Transición de todos los estilos en 0.5
segundos */
      margin: 50px auto; /* Centrado de la caja */
    }

    /* Estilo al pasar el ratón */
    .caja:hover {
      background-color: #2ecc71; /* Cambio de color al pasar el ratón */
      transform: scale(1.2); /* Aumento de tamaño al pasar el ratón */
    }
  </style>
</head>
<body>

  <!-- Caja interactiva -->
  <div class="caja"></div>

</body>
</html>
```

En este ejemplo:

- Cuando se pasa el ratón sobre la caja, el color de fondo cambia de azul a verde y se aumenta su tamaño gradualmente.

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

Animaciones

Las animaciones permiten crear movimientos complejos. Usamos las reglas `@keyframes` para definir los pasos de la animación.

Sintaxis básica:

```
@keyframes nombre-de-la-animación {
  0% {
    /* Estado inicial */
  }
  100% {
    /* Estado final */
  }
}

.elemento {
  animation: nombre-de-la-animación tiempo tipo-de-velocidad;
}
```

Ejemplo de Animación:

En este ejemplo:

- La caja cambia de color y rota 360 grados de manera infinita, con una duración de 2 segundos por ciclo.

Ejercicio

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>

    /* Definición de la animación con @keyframes */
    @keyframes animarCaja {
  /* Estado inicial de la animación (0%) */
    0% {
        background-color: #3498db; /* Color de fondo azul */
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

```
        transform: rotate(0deg); /* La caja no tiene rotación */
    }

  /* Estado final de la animación (100%) */
    100% {
        background-color: #e74c3c; /* Color de fondo rojo */
        transform: rotate(360deg); /* La caja gira 360 grados (vuelta
completa) */
    }
}

/* Estilo de la caja animada */
    .caja {
        width: 200px; /* Ancho de la caja de 200 píxeles */
        height: 200px; /* Altura de la caja de 200 píxeles */
        animation: animarCaja 2s infinite; /* Aplica la animación
"animarCaja" durante 2 segundos y se repite indefinidamente */
    }

    </style>
</head>
<body>
    <div class="caja"></div>
</body>
</html>
```

Ejercicio práctico 1: creación de un página que tenga los conceptos de flexbox, grid, Media query, animaciones y transiciones.

Html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo Completo con Flexbox, Grid y Animaciones</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <!-- Encabezado -->
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

```
  <header class="header"> <!-- Contenedor del encabezado de la página -->
    <h1>Ejemplo Completo</h1> <!-- Título principal de la página -->
    <nav class="nav"> <!-- Sección de navegación (barra de menú) -->
      <ul> <!-- Lista de elementos de navegación -->
        <li><a href="#">Inicio</a></li> <!-- Enlace a la sección de inicio
(aquí solo es un marcador de posición) -->
        <li><a href="#">Servicios</a></li> <!-- Enlace a la sección de
servicios -->
        <li><a href="#">Contacto</a></li> <!-- Enlace a la sección de
contacto -->
      </ul>
    </nav>
  </header>

  <!-- Sección Principal -->
  <main> <!-- Contenedor principal para todo el contenido de la página -->

    <!-- Galería de imágenes -->
    <section class="gallery"> <!-- Sección que contiene la galería de
imágenes -->
      <h2>Galería de Imágenes</h2> <!-- Título de la galería -->
      <div class="grid-container"> <!-- Contenedor que usará CSS Grid para
organizar las imágenes -->
        <!-- Elementos de la galería -->
        <div class="grid-item"><img src="rastreogps1.png" alt="Imagen
1"></div> <!-- Imagen 1 de la galería -->
        <div class="grid-item"></div> <!-- Espacio vacío en el grid -->
        <div class="grid-item">3</div> <!-- Un elemento con texto (para
ocupar espacio) -->
        <div class="grid-item"><img src="disponibi.jpg" alt="Imagen
1"></div> <!-- Imagen 2 de la galería -->
        <div class="grid-item">5</div> <!-- Otro elemento con texto -->
        <div class="grid-item">6</div> <!-- Otro elemento con texto -->
      </div>
    </section>

    <!-- Tarjetas usando Flexbox -->
    <section class="cards-section"> <!-- Sección para mostrar tarjetas -->
      <h2>Nuestras Tarjetas</h2> <!-- Título de la sección de tarjetas -->
      <div class="cards"> <!-- Contenedor que usará Flexbox para organizar
las tarjetas -->
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

```
        <!-- Tarjeta 1 -->
        <article class="card">
          <h3>Tarjeta 1</h3> <!-- Título de la tarjeta 1 -->
          <p>Contenido de la tarjeta 1.</p> <!-- Descripción de la tarjeta 1
-->
        </article>
        <!-- Tarjeta 2 -->
        <article class="card">
          <h3>Tarjeta 2</h3> <!-- Título de la tarjeta 2 -->
          <p>Contenido de la tarjeta 2.</p> <!-- Descripción de la tarjeta 2
-->
        </article>
        <!-- Tarjeta 3 -->
        <article class="card">
          <h3>Tarjeta 3</h3> <!-- Título de la tarjeta 3 -->
          <p>Contenido de la tarjeta 3.</p> <!-- Descripción de la tarjeta 3
-->
        </article>
      </div>
    </section>
  </main>

  <!-- Pie de página -->
  <footer class="footer"> <!-- Contenedor del pie de página -->
    <p>&copy; 2024 Ejemplo Completo. Todos los derechos reservados.</p> <!--
Mensaje de derechos de autor -->
  </footer>
</body>
</html>
```

Css

```
/* ===== Reset básico ===== */
* {
  margin: 0; /* Elimina el margen por defecto de todos los elementos */
  padding: 0; /* Elimina el relleno por defecto de todos los elementos */
  box-sizing: border-box; /* Asegura que el padding y el borde se incluyan
en el tamaño total del elemento */
}
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

```css
body {
  font-family: Arial, sans-serif; /* Define la fuente utilizada en el cuerpo de la página */
  line-height: 1.6; /* Aumenta la altura de línea para mejorar la legibilidad */
  background-color: #f9f9f9; /* Establece un color de fondo claro */
  color: #333; /* Define un color de texto oscuro */
}

/* ===== Encabezado ===== */
.header {
  background-color: #2c3e50; /* Color de fondo oscuro para el encabezado */
  color: #fff; /* Color de texto blanco */
  padding: 20px; /* Espaciado interno alrededor del encabezado */
  text-align: center; /* Centra el contenido dentro del encabezado */
  transition: background-color 0.3s ease; /* Define una transición suave para el cambio de color de fondo */
}

.header:hover {
  background-color: #34495e; /* Cambia el color de fondo cuando el ratón pasa sobre el encabezado */
}

.nav ul {
  list-style: none; /* Elimina los puntos de la lista */
  display: flex; /* Utiliza Flexbox para organizar los elementos en línea */
  justify-content: center; /* Centra los elementos de la lista */
  margin-top: 10px; /* Espacio superior entre el encabezado y la barra de navegación */
}

.nav li {
  margin: 0 15px; /* Espacio horizontal entre los elementos de la lista */
}

.nav a {
  color: #fff; /* Establece el color de los enlaces como blanco */
  text-decoration: none; /* Elimina el subrayado por defecto de los enlaces */
  font-weight: bold; /* Hace el texto de los enlaces en negrita */
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

```
  transition: color 0.3s ease; /* Transición suave para el cambio de color
de los enlaces */
}

.nav a:hover {
  color: #1abc9c; /* Cambia el color del enlace cuando el ratón pasa sobre
él */
}

/* ===== Galería usando CSS Grid ===== */
.gallery {
  padding: 20px; /* Espaciado interno de la galería */
  text-align: center; /* Centra el texto dentro de la galería */
}

.grid-container {
  display: grid; /* Utiliza CSS Grid para organizar los elementos */
  grid-template-columns: repeat(3, 1fr); /* Crea 3 columnas de igual tamaño
*/
  gap: 10px; /* Espacio entre las celdas del grid */
  margin-top: 20px; /* Espacio superior entre la galería y otros elementos
*/
}

.grid-item {
  background-color: #3498db; /* Color de fondo azul para las celdas */
  color: #fff; /* Color de texto blanco */
  text-align: center; /* Centra el contenido dentro de cada celda */
  padding: 40px; /* Espaciado interno de cada celda */
  border-radius: 8px; /* Bordes redondeados */
  font-size: 1.5rem; /* Tamaño de fuente más grande */
  transition: transform 0.3s ease, background-color 0.3s ease; /* Transición
para el cambio de color y la transformación */
}

.grid-item:hover {
  background-color: #2980b9; /* Cambia el color de fondo cuando el ratón
pasa sobre la celda */
  transform: scale(1.05); /* Aumenta ligeramente el tamaño de la celda */
}
```

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

```css
.grid-item img {
  width: 100%; /* La imagen ocupa todo el ancho de la celda */
  height: 200px; /* Establece una altura fija para la imagen */
  object-fit: cover; /* Ajusta la imagen recortando si es necesario para mantener las proporciones */
  display: block; /* Elimina el espacio blanco debajo de la imagen */
  transition: transform 0.3s ease; /* Aplica una transición para la transformación de la imagen */
}

/* ===== Sección de Tarjetas usando Flexbox ===== */
.cards-section {
  padding: 20px; /* Espaciado interno de la sección de tarjetas */
  text-align: center; /* Centra el texto de las tarjetas */
}

.cards {
  display: flex; /* Utiliza Flexbox para organizar las tarjetas en línea */
  justify-content: space-around; /* Distribuye las tarjetas de manera uniforme */
  flex-wrap: wrap; /* Permite que las tarjetas se ajusten a la siguiente línea si no caben */
}

.card {
  background-color: #fff; /* Color de fondo blanco para las tarjetas */
  border: 1px solid #ddd; /* Bordes finos grises para las tarjetas */
  border-radius: 8px; /* Bordes redondeados */
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Sombra suave para darle profundidad a la tarjeta */
  margin: 10px; /* Espacio entre las tarjetas */
  padding: 20px; /* Espaciado interno de cada tarjeta */
  width: 30%; /* Establece el ancho de las tarjetas al 30% */
  transition: transform 0.3s ease, box-shadow 0.3s ease; /* Transición para los efectos de transformación y sombra */
}

.card:hover {
  transform: translateY(-10px); /* Eleva la tarjeta cuando el ratón pasa sobre ella */
```

TALENTO TECH VALLE ACTIVIDAD: CSS 103 – 104 CLASE 6	

```
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2); /* Aumenta la sombra para dar
más énfasis al efecto de elevación */
}

/* ===== Animación al cargar las tarjetas ===== */
@keyframes fade-in {
  from {
    opacity: 0; /* Inicia con la tarjeta completamente transparente */
    transform: translateY(20px); /* Inicia con la tarjeta desplazada hacia
abajo */
  }
  to {
    opacity: 1; /* Termina con la tarjeta completamente visible */
    transform: translateY(0); /* Termina con la tarjeta en su posición
original */
  }
}

.card {
  animation: fade-in 0.8s ease-in-out; /* Aplica la animación 'fade-in' a
las tarjetas con duración de 0.8s */
}

/* ===== Pie de página ===== */
.footer {
  background-color: #2c3e50; /* Color de fondo oscuro para el pie de página
*/
  color: #fff; /* Color de texto blanco */
  text-align: center; /* Centra el texto del pie de página */
  padding: 10px 0; /* Espaciado interno en la parte superior e inferior del
pie de página */
  margin-top: 20px; /* Espacio superior entre el pie de página y otros
elementos */
}

/* ===== Media Queries ===== */
/* Dispositivos medianos (pantallas con un ancho máximo de 768px) */
@media (max-width: 768px) {
  .grid-container {
    grid-template-columns: repeat(2, 1fr); /* Reduce las columnas del grid a
2 */
```

TALENTO TECH VALLE ACTIVIDAD: CSS 103 – 104 CLASE 6	

```
  }

  .card {
    width: 45%; /* Ajusta el tamaño de las tarjetas al 45% */
  }
}

/* Dispositivos pequeños (pantallas con un ancho máximo de 480px) */
@media (max-width: 480px) {
  .grid-container {
    grid-template-columns: 1fr; /* Reduzca el grid a una sola columna */
  }

  .cards {
    flex-direction: column; /* Organiza las tarjetas en una columna en lugar
de una fila */
    align-items: center; /* Centra las tarjetas en el contenedor */
  }

  .card {
    width: 90%; /* Aumenta el tamaño de las tarjetas al 90% del ancho
disponible */
  }

  .nav ul {
    flex-direction: column; /* Organiza la barra de navegación verticalmente
*/
  }

  .nav li {
    margin: 5px 0; /* Reduce el espacio entre los elementos de la lista */
  }
}
```

Material complementario

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

https://css-tricks.com/snippets/css/complete-guide-grid/

https://www.w3schools.com/css/css_rwd_mediaqueries.asp

TALENTO TECH VALLE **ACTIVIDAD: CSS** **103 – 104** **CLASE 6**	

Actividad: creación de un restaurante

Objetivo del Proyecto:

Crear una página web para un restaurante que sea **responsiva**, moderna y con interacciones visuales atractivas. Esta página debe incluir una **galería de platos**, una **sección de menús**, una **barra de navegación interactiva** y un **pie de página** con la información de contacto.

Los siguientes **conceptos clave** deben estar implementados y evidenciados en el diseño:

- **Flexbox**: Para organizar y distribuir elementos en la página (por ejemplo, en la barra de navegación, en las tarjetas de menús, etc.).
- **CSS Grid**: Para la galería de imágenes (muestra los platos del restaurante en un diseño de cuadrícula que se ajuste según el tamaño de la pantalla).
- **Media Queries**: Para que la página sea **responsiva**, adaptándose a diferentes tamaños de pantalla (escritorio, tabletas, móviles).
- **Animaciones y Transiciones**: Para agregar dinamismo y mejorar la experiencia del usuario, como la transición suave de colores en la barra de navegación, efectos al pasar el ratón sobre las tarjetas de los menús, y animaciones al cargar la página.

Requerimientos del Proyecto:

1. **Encabezado (header)**:
    - **Nombre del restaurante** en el encabezado.
    - Una **barra de navegación** con al menos 4 enlaces: **Inicio**, **Menú**, **Galería**, **Contacto**.
    - La barra de navegación debe estar distribuida con **Flexbox** y ser **responsiva**. En pantallas pequeñas, la navegación debe cambiar de horizontal a vertical.

- **Sección Principal (main)**:
    - **Galería de imágenes** (usando **CSS Grid**):
        - Muestra 6 platos del restaurante.
    - Las imágenes deben ocupar el **100% del ancho** de su contenedor, y deben tener un **efecto hover** (al pasar el ratón) donde se ampliarán un poco, dando un efecto de **zoom**.
2. **Sección de Menú (section.menu)**:

- Crea una **tarjeta para cada plato** del menú (por ejemplo, **Entrantes**, **Plato Principal**, **Postres**, **Bebidas**).
- Cada tarjeta debe tener:
  - Un **título** (nombre del plato).
  - Una **breve descripción** del plato.
  - Un **precio**.
  - Una **imagen pequeña** del plato.
- Utiliza **Flexbox** para alinear las tarjetas del menú.
- Al pasar el ratón sobre una tarjeta, debe **expandirse ligeramente** y cambiar el color de fondo usando **transiciones**.

3. **Animaciones**:
   - **Efectos hover**:
     - Las tarjetas de los platos deben tener una animación al pasar el ratón que las haga **crecer** y mostrar una ligera **sombra**.
   - **Efectos hover en la barra de navegación**:
     - Al pasar el ratón sobre los enlaces de la barra de navegación, el **color de los enlaces debe cambiar** suavemente.
4. **Pie de Página (footer)**:
   - Incluir la **información de contacto** del restaurante (dirección, teléfono, correo electrónico).
   - Usar un **diseño responsivo** con **Flexbox** para organizar la información en el pie de página.

Conclusión

Con los conceptos de **Flexbox**, **CSS Grid**, **Diseño Responsivo**, **Transiciones** y **Animaciones**, tienes una poderosa caja de herramientas para diseñar páginas web modernas, atractivas y funcionales. Experimenta con estos conceptos para crear sitios web que se adapten a cualquier dispositivo y ofrezcan una excelente experiencia de usuario.