**Navbar (desktop and mobile)**

    <nav class="main-navbar">

      <ul>

        <li>

          <a href="hero.html">Hero</a>

        </li>

        <li>

          <a href="cards.html">Cards</a>

        </li>

        <li>

          <a href="tiles.html">Tiles</a>

        </li>

        <li>

          <a href="layout.html">Layout</a>

        </li>

      </ul>

    </nav>

* Comenzamos estilizando la Navbar para Mobile:

.main-navbar {background: var(--darkGrey);

padding: 0 2rem}

.main-navbar li {list-style: none}

.main-navbar a {

  text-transform: capitalize;

  display: block;

  padding: 1rem 0;

  color: var(--primaryColor);

  letter-spacing: var(--mainSpacing);

  transition: var(--mainTransition);

}

.main-navbar a:hover {color: var(--mainBlack)}

Aquí queremos **Colores** y que **los Links ocupen todo el LARGO** de la NavBar Mobile (**Display: Block**). Cuando **Hover** el texto cambia a Negro.

* Query para Screens mas Grandes:

@media screen and (min-width: 992px) {

  .main-navbar ul {max-width: 1170px;

/\*No superara dicho width por mas que la pantalla se agrande\*/

    margin: 0 auto;

/\*Centramos la lista al medio Horizontal\*/

    background: var(--mainBlack);

    display: grid;

    grid-template-columns: repeat(4, 1fr);

/\*Separamos la main-navbar en 4fr,pero

el contenido de cada FR no esta centrado\*/

  }

  .main-navbar a {text-align: center;

/\*Centramos el contenido (text) en el element\*/

  }

}

**Hero Section**

Si tenemos small screen size el bg será solid color, si se agranda será un Bg IMG.

    <header class="hero-header">

      <div class="hero-banner">

        <h1>Welcome to Grid</h1>

        <p>

          Lorem ipsum dolor sit amet, consectetur adipisicing elit. In repudiandae, aut qui

autem explicabo impedit!

        </p>

      </div>

    </header>

* Comenzamos con Mobile Style (sin BGimg, para mayor fluidez y carga rápida):

.hero-header {

  min-height: 100vh; /\*ocupe toda la screen\*/

  background-color: var(--primaryColor);

  color: var(--mainWhite);

  display: grid;

  justify-content: center; /\*centrado horizontal a la CHILD\*/

  align-items: center; /\* centrado vertical Child - Falta centrar content\*/

}

.hero-banner {

  /\* background: blue; para verlo al editarlo, luego lo removemos \*/

  /\* text-align: center; Centramos el Content, si necesario. Aqui queremos todo al left side. \*/

}

.hero-banner p {

  width: 80%;

  max-width: 560px; /\*estiramiento maximo\*/

  font-size: 1rem;

}

* Query para Bigger Size Screens:

@media screen and (max-width: 600px) {

  .hero-banner {

    padding: 0 1rem;

    /\*Caudno supere 600px, se aleja el text del borde\*/

  }

}

@media screen and (min-width: 768px) {

  .hero-header {

    background: linear-gradient(rgba(241, 80, 37, 075), rgba(0, 0, 0, 0.75)),

      url("./images/image-2.jpeg") center/cover no-repeat fixed;

    /\*imagen de fondo y ung radient\*/

  }

}

**Cards Section**

Queremos un Titulo, Mobile: 1 card layout, Tablet: 2x2 y Desktop: 3 por fila. Haremos Query y sin Query.

    <section class="cards">

      <div class="title">

        <h1>Three Column Layout</h1>

      </div>

      <div class="cards-center">

        <!-- Single Card -->

        <article class="card">

          <img src="./images/size-image-1.jpeg" alt="" />

          <!-- cardinfo -->

          <div class="card-info">

            <h3>Card Title</h3>

            <p>

              Lorem ipsum dolor sit amet consectetur adipisicing elit. Hic, nemo?

            </p>

          </div>

          <!-- cardfooter -->

          <div class="card-footer">

            <span><i class="fab fa-twitter"></i></span>

            <span><i class="fab fa-facebook"></i></span>

            <span><i class="fab fa-squarespace"></i></span>

            <span><i class="fab fa-linkedin"></i></span>

          </div>

        </article>

        <!-- End Single Card -->

🡪 Repetimos la cantidad de veces necesaria la card

* Style for Mobile First (1 column layout)

.title {text-align: center}

.cards {padding: 4rem 0}

.cards-center {

  width: 80vw;

  max-width: 1170px;

  margin: 2rem auto;

  /\*Hasta aqui dimos tamaño a la page y centramos las cards\*/

}

.card {border: 1px solid var(--darkGrey);

  margin-bottom: 3rem}

.card-info {padding: 1rem}

.card-footer {background: var(--darkGrey);

  padding: 0.5rem 1rem;

  display: grid;

  grid-template-columns: repeat(4, 1fr);

  justify-items: center;

  /\*Separamos en 4 cols y centramos contenido\*/

}

.card-footer span {

  color: var(--primaryColor);

/\*Cambiamos el color al icono\*/

}

* Query para Bigger Screens:

@media screen and (min-width: 768px) {

  .cards-center {

    display: grid;

    grid-template-columns: repeat(2, 1fr);

    column-gap: 2rem;

    /\*Column Layout y espacio\*/

  }

  @media screen and (min-width: 1170px) {

    .cards-center {

      grid-template-columns: repeat(3, 1fr);

    }

  }

}

* Without Query:

Cosas a saber: Max Width (1170px), Cuantos espacios y dichas medidas (Gaps 2rem). Crearemos x3 Layout y x4 Layout.

Gaps 🡪 1rem = 16px // Gaps de 2rem = 32px… 4 gaps = 96px.

Columna 🡪 1170px **–** Gap ( x3 = 64px) = **1106px**

**1106px** **/** Cantidad Cols (3) = **368.666px**

* For 3 Columns with size of: 368px

.cards-center {

  display: grid;

  column-gap: 2rem;

  grid-template-columns: repeat(auto-fill, minmax(368px, 1fr));

}

¿Qué pasa? We start with 1 Column, the moment Grid has **room to place** another **Columns** of **368px** it will place them one next to each other. Even though there is **NO CONTENT** (having just one card) Grid **Auto-Fill will still create the columns. (Fil lis good to work with APIs, creating ítems before knowing the amount of content / cards)**

* For 4 Columns with size of 268.5px

.cards-center {

  display: grid;

  column-gap: 2rem;

  grid-template-columns: repeat(auto-fill, minmax(268.5px, 1fr));

}

**Row Height:**

What happens if we have different content within one card? We are using data where we don’t have the control over it.

Ejemplo: La descripción de una card es muy larga y nos estira la card en comparación al resto. Atacamos directo a Cada Card

.card {

  display: grid;

  grid-template-columns: 1fr; /\*Aclaramos que cada tarjeta tenga 1 sola columna\*/

  grid-template-rows: auto 1fr auto; /\* img, info, footer\*/

}

Decimos que la CARD ocupe 1 Columna (1fr) y que sus Filas (image, description, footer) sean: **Auto 1fr Auto** 🡪 **Esto hará que el P o description ocupe el espacio necesario y que la IMG / Footer sea automatico dicho P.**

**Tiles Section**

Una linda section donde dispondremos las imágenes de distintos widths y heights.

    <section class="tiles">

      <div class="title">

        <h1>Tiles Layout</h1>

      </div>

      <div class="tiles-center">

        <!-- Single Tile -->

        <article class="tile-1">

          <img src="./images/size-image-1.jpeg" alt="" />

        </article>

        <!-- End Single Tile -->

* Repetimos as Necessary

.tiles {

  padding: 4rem 0;

}

.tiles-center {

  width: 90vw;

  max-width: 1170px;

  margin: 2rem auto; /\*hasta aqui acomodamos las imagenes en la page\*/

  display: grid;

  row-gap: 2rem; /\*Separacion entre imgs\*/

}

* Comenzamos con las Tiles de 2 Col and 3 Col:

@media screen and (min-width: 768px) {

  .tiles-center {

    grid-template-columns: 1fr 1fr;

    column-gap: 2rem;

  }

}

@media screen and (min-width: 992px) {

  .tiles-center {

    grid-template-columns: 1fr 1fr 1fr;

    column-gap: 2rem;

  }

}

* Ahora para pantallas mas Grandes haremos un Tile Layout especifico:
* Cosas a **saber de antemano Rows and Columns:**

**4 Col // 5 Row**

@media screen and (min-width: 1170px) {

  .tiles-center {

    grid-template-columns: repeat(4, 1fr);

    grid-template-rows: repeat(5, 200px);

    grid-template-areas:

      "a a a b"

      "a a a c"

      "d d d d"

      "e f f f"

      "e g h i";

  }

  .tile-1 {grid-area: a}

  .tile-2 {grid-area: b}

  .tile-3 {grid-area: c}

  .tile-4 {grid-area: d}

  .tile-5 {grid-area: e}

  .tile-6 {grid-area: f }

  .tile-7 {grid-area: g}

  .tile-8 {grid-area: h}

  .tile-9 {grid-area: i}

}

* En este punto esta armada la Layout, Pasa que las imágenes se superponen, debemos aclarar el valor para la IMG:

  .tiles img {

    height: 100%;

    object-fit: cover;

  }

**Layout Section**

This section will change its size depending on screen. Tendremos la Navbar ajustable a las screens (sidebar, mobile, header) y el footer igual. Asu vez el contenido se adaptará para dar lugar.

    <main class="main">

      <!-- Header -->

      <header class="header">

        <h2>Grid Makes layout Easy</h2>

      </header>

      <!-- End Header -->

      <!-- Navbar -->

      <nav class="layout-navbar">

        <ul>

          <li>

            <a href="./hero.html">Hero</a>

          </li>

          <li>

            <a href="./cards.html">Cards</a>

          </li>

          <li>

            <a href="./tiles.html">Tiles</a>

          </li>

          <li>

            <a href="./layout.html">Layout</a>

          </li>

        </ul>

      </nav>

      <!-- End NavBar -->

      <!-- Content -->

      <section class="section">

        <p>Lorem75</p>

      </section>

      <!-- End Content -->

      <!-- Sidebar -->

      <section class="seidebar">

        <span><i class="fab fa-twitter"></i></span>

        <span><i class="fab fa-facebook"></i></span>

        <span><i class="fab fa-squarespace"></i></span>

        <span><i class="fab fa-linkedin"></i></span>

      </section>

      <!-- End Sidebar -->

      <!-- Footer -->

      <div class="footer">

        <p>Copyrith &copy; all rights reserved.</p>

      </div>

      <!-- End Footer -->

    </main>

    <!-- End Layout Section -->

Comenzamos con lo base:

.header {

  background: var(--primaryColor);

  color: var(--mainWhite);

  text-align: center;

  padding: 0.5rem 0;

}

.layout-navbar {background: var(--darkGrey)}

.layout-navbar li {list-style-type: none}

.layout-navbar a {

  padding: 1rem;

  display: inline-block;

  text-transform: capitalize;

  letter-spacing: var(--mainSpacing);

  font-size: 1.2rem;

  color: var(--mainBlack);

  transition: var(--mainTransition);

}

.layout-navbar a:hover {color: var(--offWhite)}

.section {padding: 2rem}

.section p {font-size: 1rem}

.sidebar {

  background: var(--primaryColor);

  padding: 1rem 0;

  display: grid;

  grid-template-columns: repeat(4, 1fr);

  justify-items: center;

  color: var(--mainWhite);

  font-size: 1.2rem;

}

.footer {

  background: var(--mainBlack);

  color: var(--mainWhite);

  text-align: center;

  text-transform: capitalize;

  padding: 0.5rem 0;

}

.footer p {font-size: 1.2rem}

* ¿Que pasaría si borramos algo del contenido? Se achicará todo! Entonces hay que dar una Min Height y que la page no se modifique según el contenido, que todo el resto sea AUTO.

.main {

  min-height: 100vh;

  display: grid;

  grid-template-columns: 1fr;

  grid-template-rows: auto auto 1fr auto auto;

}

Pensamos en la cantidad de filas por elemento:

Header, Navbar, CONTENIDO, Sidebar, Footer

* Creamos un Query para poner la NavBar horizontal

@media screen and (min-width: 768px) {

  .layout-navbar ul {

    display: grid;

    grid-template-columns: repeat(4, 1fr);

    justify-items: center;

  }

}

* Creamos un Query para la Navbar Vertical enfrentada a la Sidebar, con contenido en el medio.
* A saber:

Screen size (when), Columns (3), Rows (3).

* Se puede hacer con Naming Girds y Grid Area.
* Comenzamos armando las Rows y Cols:

@media screen and (min-width: 992px) {

  .main {

    grid-template-columns: [start] 200px [col-1-end] 1fr [col-2-end] 50px [end];

    grid-template-rows: [start] auto [row-1-end] 1fr [row-2-end] auto [end];

  }

}

En este caso queremos **nombrar** asu vez las **LINEAS.**

* ¿Dónde queremos los Elementos? Dentro del Media Query:

  .header {

    grid-column: start/end;

    grid-row: start/row-1-end;

  }

  .layout-navbar {

    grid-column: start/col-1-end;

    grid-row: row-1-end/row-2-end;

  }

  .section {

    grid-column: col-1-end/col-2-end;

    grid-row: row-1-end/row-2-end;

  }

  .sidebar {

    grid-column: col-2-end/end;

    grid-row: row-1-end/row-2-end;

  }

  .footer {

    grid-column: start/end;

    grid-row: row-2-end/end;

  }

}

* Queda retocar el contenido para cada elemento:

  .layout-navbar ul {

    grid-template-columns: 1fr;

    align-items: center;

    min-height: 100%;

  }

Esto hará que los LI ocupen el 100% de la altura del Parent y

Align-ítems los centra.

  .sidebar {

    grid-template-columns: 1fr;

    align-items: center;

    min-height: 100%;

  }

Lo mismo para la SIdebar, Acomodar los Iconos al medio y a lo alto.