/\* RESET \*/

\*{

    width: 100%;

}

html,body,div,span,applet,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,a,abbr,acronym,address,big,cite,code,del,dfn,em,img,ins,kbd,q,s,samp,small,strike,strong,sub,sup,tt,var,b,u,i,center,dl,dt,dd,ol,ul,li,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,article,aside,canvas,details,embed,figure,figcaption,footer,header,hgroup,menu,nav,output,ruby,section,summary,time,mark,audio,video {

    margin: 0;

    padding: 0;

    border: 0;

    font-size: 100%;

    font: inherit;

    vertical-align: baseline;

}

article,aside,details,figcaption,figure,footer,header,hgroup,menu,nav,section {

    display: block;

}

body {

    line-height: 1;

}

ol,ul {

    list-style: none;

}

blockquote,q {

    quotes: none;

}

blockquote:before,blockquote:after,q:before,q:after {

    content: '';

    content: none;

}

table {

    border-collapse: collapse;

    border-spacing: 0;

}

/\* FUENTES \*/

@font-face {

    font-family: 'Open Sans';

    src: url('fonts/OpenSans-Regular.ttf') format('truetype');

    font-style: normal;

    font-weight: normal;

  }

/\* BOTON \*/

.boton {

    color: rgb(255, 255, 255);

    font-size: 16px;

    line-height: 16px;

    padding: 6px;

    border-radius: 10px;

    font-family: Georgia, serif;

    font-weight: normal;

    text-decoration: none;

    font-style: normal;

    font-variant: normal;

    text-transform: none;

    background-image: linear-gradient(to right, rgb(28, 110, 164) 0%, rgb(35, 136, 203) 50%, rgb(20, 78, 117) 100%);

    box-shadow: rgb(0, 0, 0) 5px 5px 15px 5px;

    border: 2px solid rgb(28, 110, 164);

    display: inline-block;

}

    .boton:hover {

    background: #1e6696;

}

    .boton:active {

    background: #0d3d5e;

}

/\* BACKGROUND \*/

.fondo{

background: #D0E4F5 url("logo.jpg") no-repeat scroll 0px 0px;

}

/\* BOX-SHADOW \*/

.sombra{

-webkit-box-shadow: 8px 5px 12px 2px rgba(0,0,0,0.56);

box-shadow: 8px 5px 12px 2px rgba(0,0,0,0.56);

}

/\* TEXT \*/

.texto{

font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;

font-size: 16px;

letter-spacing: 2px;

word-spacing: 2px;

color: #000000;

font-weight: 400;

text-decoration: none solid rgb(68, 68, 68);

font-style: normal;

font-variant: normal;

text-transform: none;

}

/\* BORDE \*/

.borde{

outline: 3px ridge rgba(28,110,164,0.89);

outline-offset: 4px;

border-radius: 3%;

}

/\* TRANSFORM \*/

.transformado{

-moz-transform: scale(1.1) translate(7px, -3px) skew(10deg, 6deg);

-webkit-transform: scale(1.1) translate(7px, -3px) skew(10deg, 6deg);

-o-transform: scale(1.1) translate(7px, -3px) skew(10deg, 6deg);

-ms-transform: scale(1.1) translate(7px, -3px) skew(10deg, 6deg);

transform: scale(1.1) translate(7px, -3px) skew(10deg, 6deg);

}

/\* MEDIA QUERYS \*/

@media screen and (max-width: 600px), (max-height: 500px) {

}

/\* SELECTORES Y PSEUDOCLASES

\* all elements

div all div tags

div,p all divs and paragraphs

div p paragraphs inside divs

div > p all p tags, one level deep in div

div + p p tags immediately after div

div ~ p p tags preceded by div

.classname all elements with class

#idname element with ID

div.clas snamedivs with certain classname

div#idname div with certain ID

#idname \* all elements inside #idname

Pseudo classes

a:link link in normal state

a:active link in clicked state

a:hover link with mouse over it

a:visited visited link

p::after{content:"yo";}add content after p

p::before add content before p

input:checked checked inputs

input:disabled disabled inputs

input:enabled enabled inputs

input:focus input has focus

input:in-range value in range

input:out-of-range input value out of range

input:valid input with valid value

input:invalid input with invalid value

input:optional no required attribute

input:required input with requred attribute

input:read-only with readonly attribute

input:read-write no readonly attrib.

div:empty element with no children

p::first-letter first letter in p

p::first-line first line in p

p:first-of-type first of some type

p:last-of-type last of some type

p:lang(en) p with en language attribute

:not(span) element that's not a span

p:first-child first child of its parent

p:last-child last child of its parent

p:nth-child(2) second child of its parent

p:nth-child(3n+1) nth-child (an + b) formula

p:nth-last-child(2) second child from behind

p:nth-of-type(2) second p of its parent

p:nth-last-of-type(2) ...from behind

p:only-of-type unique of its parent

p:only-child only child of its parent

:root documents root element

::selection portion selected by user

:target highlight active anchor

\*/

/\* ABSOLUTE, RELATIVE, STICKY, FIXED \*/

.padrerelativo{

    position: relative;

}

.hijosticky{

    position:sticky;

    top:0;

}

/\* FLOAT, CLEAR \*/

.divflotante{

    float: left;

}

.siguientediv{

    clear: both;

}

/\* FLEX \*/

.cajaflexpadre{

    display: flex;

    align-content: space-around;/\*alineacion vertical\*/

    flex-direction: column-reverse;/\*row, column..\*/

    flex-wrap: nowrap; /\*pasa o no a la siguiente linea\*/

    justify-content: space-between;/\*alineacion horizontal\*/

    align-items: center;/\*alineacion vertical\*/

}

.cajaflex{

    order: 0;

    flex-grow: 0;

    flex-shrink: 0;

    flex-basis: 3px auto;

    align-self: flex-end;

}

/\* ---- GRID ----- \*/

.padreGRID1 {

    /\*

    Para centrar al padre, hay que poner

    al abuelo (body en este caso) :

    display: flex;

    justify-content: center;

    Después sí funciona esto aquí en el centro,

    si no, lo deja por defecto a la izquierda (start):

    width: 90%;

    \*/

    display: grid;

    /\*

    display: -ms-inline-grid;

    display: -moz-inline-grid;

    \*/

    grid-template-columns: [\_1of4\_column] 100px

                            [\_2of4\_column] 50px

                            [\_3of4\_column] 150px

                            [\_4of4\_column] 30px;

    grid-template-rows: [\_1of4\_line] 100px

                        [\_2of4\_line] 30px

                        [\_3of4\_line] 100px

                        [\_4of4\_line] 60px;

    /\*Alineacion Horizonal\*/

    justify-items: stretch;

    /\*Alineacion Vertical\*/

    align-items: stretch;

    /\*Distribución Horizontal\*/

    justify-content: stretch;

    /\*Distribución Vertical\*/

    align-content: space-between;

    /\*Distribución Horizonal y Vertical,

      las dos a la vez: \*/

    place-content: stretch center;

    /\*da tamaño auto y además se asegura que

    se dibujen celdas si se establecen fuera del grid\*/

    /\*

    rellena primero los huecos, hay que llevar cuidado

    de usarlo, varía el orden de colocación

    grid-auto-flow:dense;

    \*/

    grid-auto-rows: 50px;

    /\*gap es el espacio entre celdas \*/

    gap: 2em;

}

.hijo{

    border: 1em solid black;

    text-align: center;

    vertical-align: middle;

    height: 100%;

}

#amarillo {

    background-color: yellow;

    grid-column: 2/4;

    grid-row: 3/5;

    justify-self: center;

    align-self: center;

    padding: 10px;

}

/\*areas\*/

#verde {

    background-color: greenyellow;

    grid-area: 1/1/2/5;

}

#naranja {

    background-color: orange;

    grid-area: 2/1/3/5;

}

#rojo{

    background-color: red;

    grid-area:6/4/7/5;

}

#gris{

    background-color: grey;

    /\*se sale del grid la fila hasta 9

    pero como tiene el grid-auto-row definido

    se coloca igualmente saliéndose del grid\*/

    grid-area:2/2/9/3;

}

/\*---- GRID2 -----\*/

.padreGRID2 {

    min-width: 100%;

    min-height: 100vh;

    display: grid;

    grid-template-columns: repeat(25,1fr);

    grid-template-rows: repeat(25,1fr);

    grid-auto-rows: 1fr;

    grid-auto-columns:1fr;

    gap: 2em;

}

#cabecera {

    /\*grid-area: startcolumn/startline/endcolumn/endline\*/

    grid-area: 1/1/6/26;

    background-color: yellow;

}

#logo {

    grid-area:2/3/4/7;

    background-color: white;

}

#textarea {

    grid-area:2/19/3/23;

    background-color: white;

}

#lupa {

    grid-area:2/24/3/25;

    background-color: white;

}

#bandera1{

    grid-area:4/22/5/23;

    background-color: white;

}

#bandera2{

    grid-area:4/24/5/25;

    background-color: white;

}

#zona\_menu{

    grid-area: 6/2/8/20;

    background-color: blueviolet;

}

#zona\_aside{

    grid-area: 8/20/23/26;

    background-color: green;

}

#zona\_carrousel{

    grid-area: 10/5/16/18;

    background-color: greenyellow;

}

#zona\_tarjetas{

    grid-area: 18/2/22/16;

    background-color: pink;

}

#zona\_pie{

    grid-area: 23/1/26/26;

    background-color: grey;

}

/\* -- GRID 3 --- \*/

.padreGRID3 {

    min-width: 100%;

    min-height: 100vh;

    display: grid;

    grid-template-columns: repeat(7,auto);

    grid-template-rows: repeat(5,auto);

    grid-template-areas:

    "cab cab cab cab cab cab cab"

    "men men men men men men men"

    ". car car car . asi ."

    ". tar tar tar . asi ."

    "pie pie pie pie pie pie pie";

    gap: 2em;

}

#cab{

    background-color: beige;

    grid-area: cab;

}

#men{

    background-color: blue;

    grid-area: men;

}

#car{

    background-color: blanchedalmond;

    grid-area: car;

}

#tar{

    background-color: blueviolet;

    grid-area: tar;

}

#asi{

    background-color: bisque;

    grid-area: asi;

}

#pie{

    background-color: burlywood;

    grid-area: pie;

}

/\*-- abreviado --\*/

#ejemplogridabreviado{

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

    grid-gap: 20px 5px;

}

**CSS Grid**

Grid Template Columns

grid-template-columns: 40px 50px auto 50px 40px

Grid Template Rows

grid-template-rows: 25% 100px auto

Grid Template Areas

grid-template-areas: "a b c" | none

Grid Template

grid-template: "a a a" 20% "b b b" auto | 100px 1fr / 50px 1fr

Grid Column Gap

grid-column-gap: 10px

Grid Row Gap

grid-row-gap: 10px

Justify Items

justify-items: start | end | center | stretch

Align Items

align-items: start | end | center | stretch

Justify Content

justify-content: flex-start | flex-end | center | space-between | space-around | space-evenly

Align Content

align-content: flex-start | flex-end | center | space-between | space-around | stretch

Grid Auto Columns

grid-auto-columns: 100px | max-content | min-content

Grid Auto Rows

grid-auto-rows: 100px | max-content | min-content

Grid Auto Flow

grid-auto-flow: row | column | row dense | column dense

Grid Column Start

grid-column-start: 2 | areaname | span 2 | span areaname | auto

Grid Column End

grid-column-end: 2 | areaname | span 2 | span areaname | auto

Grid Row Start

grid-row-start: 2 | areaname | span 2 | span areaname | auto

Grid Row End

grid-row-end: 2 | areaname | span 2 | span areaname | auto

Grid Column

grid-column: 3 / span 2

Grid Row

grid-row: 3 / span 2

Justify Self

justify-self: start | end | center | stretch

Align Self

align-self: start | end | center | stretch

**Dynamic Content**

CSS Variable

--variable-name: value

Variable Usage

var(--variable-name)

Counter Reset

counter-reset: name-of-counter

Counter Increment

counter-increment: name-of-counter

Counter Dynamic Value

content: counter(name-of-counter)

Attribute Dynamic Value

content: attr(name-of-attribute)