**CSS**

**Estructura:**

Selector {

Propiedad: valor;

Ejemplo;

p{

color: red;

**Estilos en línea:** asignamos atributos a cada una de las etiquetas:

<h1 style = “color:blue”> <h1>

Estilo en archive externo CSS

El archivo CSS lo incorporamos en el head :

<Link rel “styleshee” href= .// la ruta del archivo >

Selectores CSS:

**Mediante etiqueta:**

Ejemplo:

H1{Color:blue

**Mediante clases:**

Asignamos una clase a las etiquetas : “class =”titulo rojo”

Identificamos la clase en el archivo CSS con .”nombre de la clase”

**Mediante ID:**

Asignamos un id en la etiqueta: <p id=” párrafo principal”>

En CSS lo referenciamos con #

#parrafo-principal {color:green

**Selectores múltiples:**

H1,h2 {color:verde

H1, .titulo-verde{

**Selectores por relación:**

Darle un estilo a todas las etiquetas de una sección, por ejemplo:

Nav a{

El peso de cada asignación:

De mayor a menor relevancia:

1. ID
2. CLASE
3. ETIQUETA

Frente a distintas valoraciones de estilos de una misma etiqueta, tiene mas peso la ultima.

**Tipos de letra:**

**Css-tricks.com**

* **Color htmlcolorcodes.com**
* **Font-family cambiar** la fuente de nuestra página : cssfontstack.com
* **Font-size –** tamaño del texto (px pixeles)
* **Font-weight** tamaño de la fuente
* **Text-decoration css-tricks text decoration:**
  + none: no line is drawn, and any existing decoration is removed.
  + underline: draws a 1px line across the text at its baseline.
  + line-through: draws a 1px line across the text at its “middle” point.
  + overline: draws a 1px line across the text, directly above its “top” point.
  + inherit: inherits the decoration of the parent.
* **Text-transform:**
  + **Uppercase 🡪 todo en mayuscula**
  + **Lowercase 🡪 todo en minuscula**
  + **Capitalize 🡪 la primer letra en mayúscula**

**Diseño del texto:**

Pongo un contenedor y dentro muevo el texto.

<div class=”contenedor”>

<h2 class=”titulo2”>

Text-align: center // rigth // left el texto al centro

Letter-spacing : espaciado entre cada letra

Font-face:

Google fonts 🡪 me descargo la fuente

@font-face {

Font-family: “the valleyFont”;

Src: url (….. dirección del archivo)

**Modelo de caja**

Height: cambio al altura de la caja

Width: cambio el tamaño general de la caja

Max-width: se acomoda el tamaño de la pagina

Border: el tamaño (px) solid // dotted // color

Margin: espacio entre cada elemento

Padding: el espacio dentro del elemento

Sombras : cssmatic.com/box-shadow

**Display y position:**

Creo cajas

<div>

Display:block 🡪 por defecto del elemento DIV – **los elementos que ocupen todo el ancho de la pagina**

Display:inline 🡪 la caja ocupa el tamaño d**el contenido**

Display:inline-block : ancho y alto definido

Display: flex 🡪 elementos flexibles que ayuda a posicionar un elemento al lado del otro. Es un elemento que **se debe poner al contenedor no al contenido.**

Display: none 🡪 oculta un elemento

**Position**

relative: top 20 px y left 20px;

Static: flujo normal de web

Fixed: elijo la posicion: bottom :// top // right// left

**Imagenes y texto superpuesto.**

Usamos una estructura de :

* Un div = caja contenedora de la imagen y del texto.
* La imagen por un lado
* Un nuevo div con el texto que quiero centrar
* Y un nuevo div para centrar el texto

Enlace de referencia:

https://www.freecodecamp.org/espanol/news/como-usar-la-propiedad-posicion-en-css-para-alinear-elementos/#:~:text=En%20position%3A%20relative%20%2C%20el%20elemento,esquina)%20de%20su%20elemento%20padre.

**Ejercicio Jimmy Lion**

**Header con un logo y menú de navegación**

**Sección con un div 🡪 imagen de movimiento**

**Div con un logo para compaginar**