

Programación Web  
Frontend Developer

POTRERO DIGITAL

# Clase 6

Luis Amarilla

[luis.amarilla@compromiso.org](mailto:luis.amarilla@compromiso.org)

## Pseudo-clases

La pseudo-clase **:link** (o **a:link**) permite especificar el aspecto de los enlaces que todavía no han sido visitados.

La pseudo-clase **:visited** (o **a:visited**) permite especificar el aspecto de los enlaces que sí han sido visitados.

La pseudo-clase **:active** permite especificar el aspecto de un elemento cuando se hace clic sobre él (y mientras se mantiene el botón del mouse apretado).

La pseudo-clase **:hover** permite especificar el aspecto del elemento sobre el que se encuentra el mouse.

# Dinamismo e interactividad

La pseudo-clase **:hover** permite especificar el aspecto del elemento cuando el mouse se encuentre sobre él.

```
nav {  
    background-color: red;  
}  
nav:hover {  
    background-color: blue;  
}
```

```
a {  
    color: #000000;  
    text-decoration: none;  
}  
a:hover {  
    color: red;  
}
```

## Formularios

```
<form>
  <label for="nombre">Nombre y Apellido</label>
  <input type="text" name="nombre" value="escribir Nombre">

  <label for="email">Correo electrónico</label>
  <input type="text" name="email" value="escribir Correo">

  <label for="mensaje">Mensaje</label>
  <textarea name="name" rows="10" cols="80">Mensaje</textarea>

  <input class="xx" type="submit" value="enviar">
</form>
```



## Responsive Design

**Responsive Web Design** o también conocido como **Diseño Web Adaptable**, es una técnica de diseño y desarrollo web que mediante el uso de estructuras flexibles (contenedores flexibles, imágenes y video flexibles) especificadas en CSS, logran adaptar un sitio web al entorno del dispositivo en el que se encuentre. Con ésta técnica se consigue que el contenido del diseño de tu sitio se vea bien y pueda ser legible para los usuarios en cualquier dispositivo con el que se interactúe.

# Unidades de medida CSS

## Unidades fijas:

**px:** Píxeles

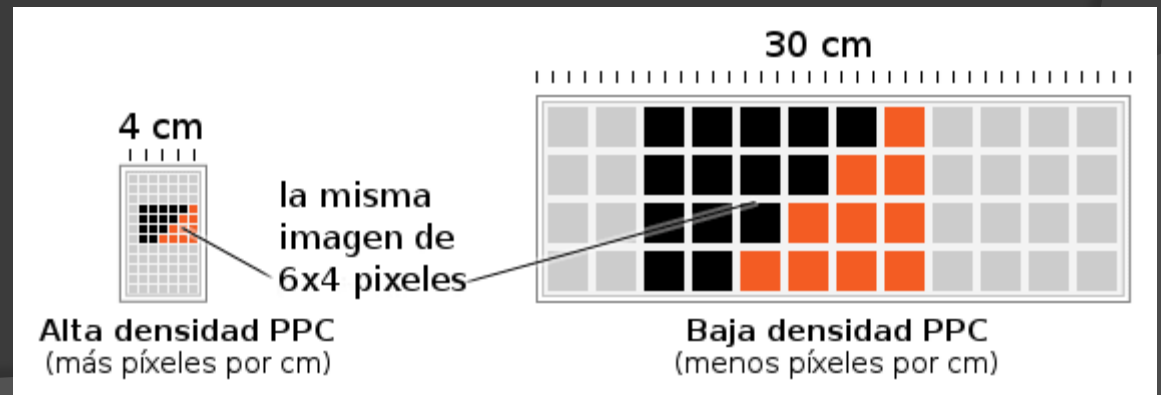
**in:** Pulgadas (1 in es igual a 96px)

**pt:** Puntos (1 pt es igual a 1/72 in)

**cm:** Centímetros

**mm:** Milímetros

**pc:** Picas



## Unidades Relativas:

Las unidades relativas de CSS son aquellas que tienen en cuenta el contexto donde se encuentran. Son relativas a las dimensiones del contenedor donde se han definido

**%:** Porcentaje

**em:** Tamaño heredado

**rem:** Tamaño de raíz

El tamaño del texto predeterminado en un navegador es de 16 píxeles

**VW:** Viewport width

**vh:** Viewport Height

1vw = 1% de la anchura del dispositivo.

1vh = 1% de la altura del dispositivo.

**vmin:** Viewport menor, entre altura o anchura

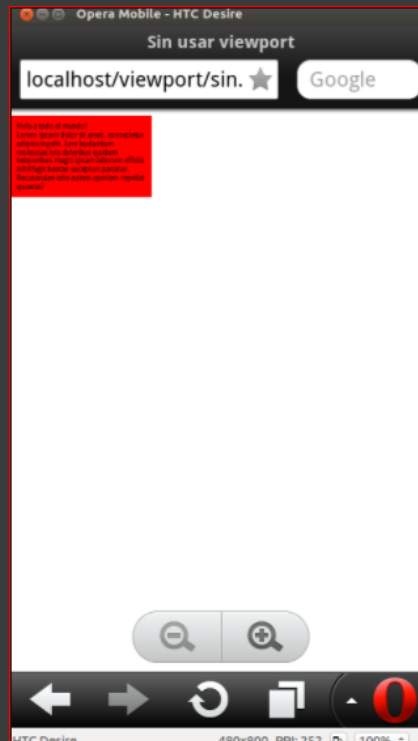
**vmax:** Viewport mayor, entre altura o anchura

**ex:** anchura de la fuente para la letra "x"

**ch:** la anchura del carácter "0" (cero)

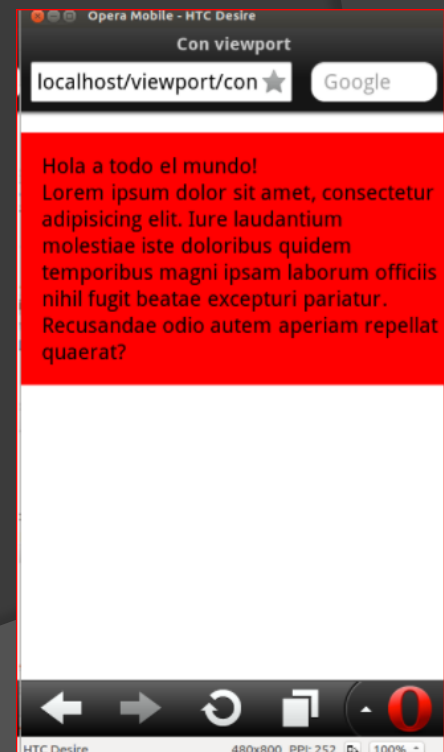
## Etiqueta meta viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```



La etiqueta meta viewport define qué área de pantalla estará disponible, cuál es el nivel de escalado que mostrará el navegador en su estado inicial y hace que las paginas web optimicen su formato para adaptarse a los distintos dispositivos y el tamaño de sus pantallas.

La etiqueta meta viewport debe declararse dentro de **<head>** en HTML



Sin meta viewport

Con meta viewport



# Box-sizing

La propiedad **box-sizing** define el comportamiento del tamaño de una caja:

**content-box** es el comportamiento CSS por defecto para el tamaño de la caja (box-sizing). Si se define el ancho de un elemento en 100 píxeles, la caja del contenido del elemento tendrá 100 píxeles de ancho, y el ancho de cualquier borde o relleno se añadirá al ancho final.

**border-box** le dice al navegador que debe tener en cuenta para cualquier valor que se especifique de borde o de relleno para el ancho o alto de un elemento. Es decir, si se define un elemento con un ancho de 100 píxeles. Esos 100 píxeles incluirán cualquier borde o relleno que se añadan, y la caja de contenido se encogerá para absorber ese ancho extra. Esto típicamente hace mucho más fácil dimensionar elementos.

## Fuentes de tipo ttf

```
@font-face {  
  font-family: "arabe";  
  src: url("../fuentes/MATURASC.ttf")  
  format('truetype');  
}  
  
p {  
  font-family: "arabe";  
}
```

Declaración de  
fuente

Archivo .ttf

Atención: no se cierra con  
punto y coma

Tipo de archivo

Aplicación de  
fuente

## Fuentes WOFF (Web Open Font Format)

### Como utilizar la librería de fuentes desde CSS

```
@import url("https://fonts.googleapis.com/css?family=Flamenco");  
  
p {  
  font-family: "Flamenco";  
  font-size: 5em;  
}
```



Declaración de la ruta  
donde se encuentra  
fuente WOFF

Aplicación de fuente

<https://fonts.google.com>

## Fuentes WOFF (Web Open Font Format)

### Declarar la librería de fuentes desde HTML y utilizarla desde CSS

```
<head>  
  <link rel="preconnect" href="https://fonts.googleapis.com">  
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
  <link href="https://fonts.googleapis.com/css2?family=Italianno&display=swap" rel="stylesheet">  
</head>
```

Declaración de la Librería  
de fuentes en el archivo  
HTML (dentro de head)

```
p {  
  font-family: 'Italianno', cursive;  
}
```

Aplicación de fuente  
en el archivo CSS