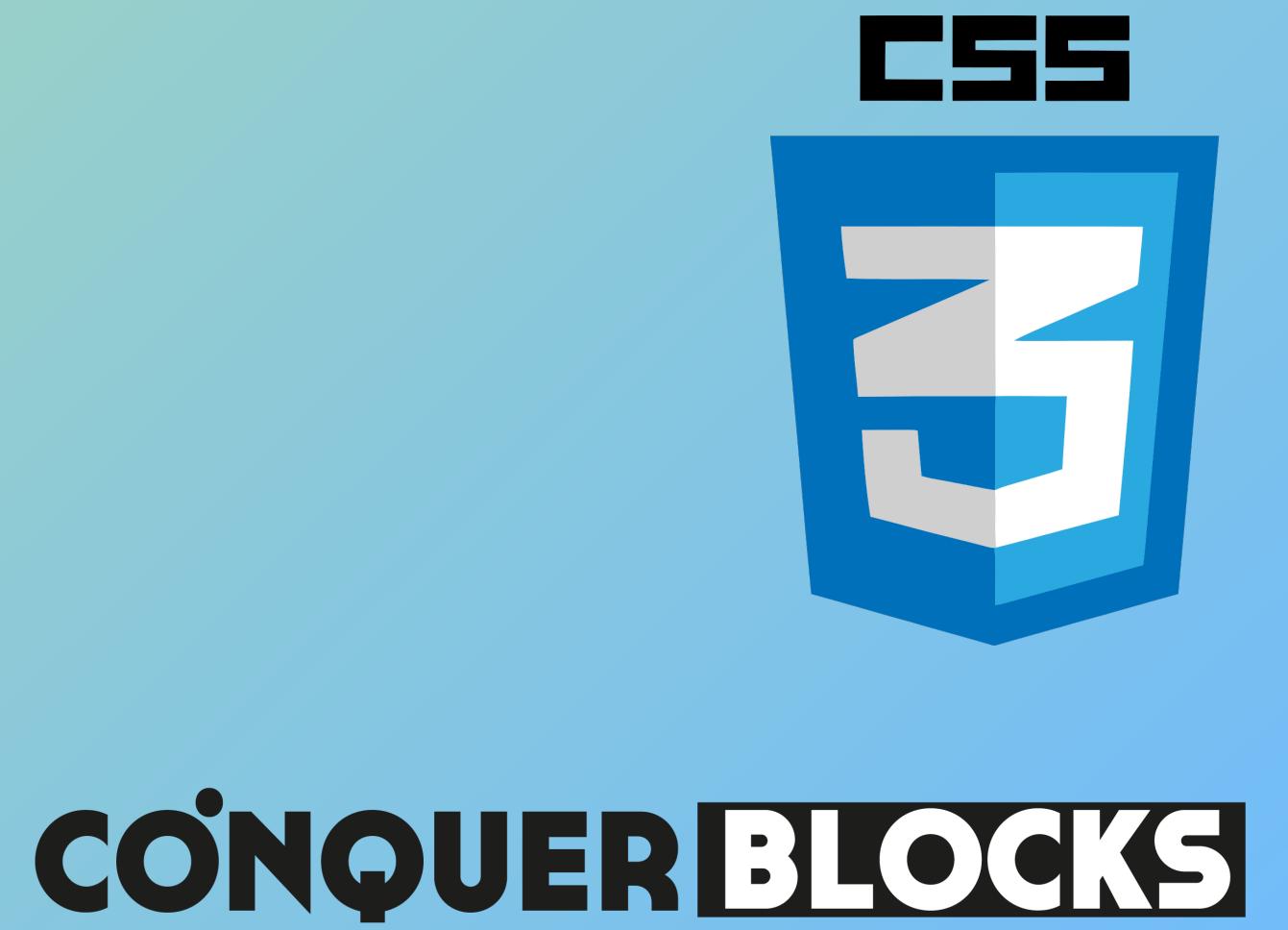


# {css}

Clase 02



# <índice>

## CSS Clase 02: Unidades de medida en CSS

**¿Qué son las unidades y cuando se utilizan?**

---

**Unidades absolutas**

---

**Unidades relativas: %, em y rem**

---

**Unidades del viewport**

---

**Funciones de cálculo**

---

**Colores en CSS**

---

# Unidades de medida

# Unidades de medida

- En CSS necesitaremos utilizar tamaños
- Altos, anchos, tamaño de fuente, interlineados etc...
- Antes de empezar a ver propiedades, veremos las unidades
- Ejemplo de propiedad con unidad

# Unidades de medida

```
selector {  
    propiedad : valor ;  
    propiedad : valor ;  
    propiedad : valor ;  
}
```

— **UNA REGLA POR LÍNEA**

terminada con punto y coma.

— **TERMINACIÓN DE REGLA**

opcional (al ser la última)

# Unidades de medida

```
p{  
    width: 700px;  
}
```

# Unidades de medida

**2 Tipos de unidades**

Unidades relativas

Unidades absolutas

# Unidades de medida

## Consejo

Cuando estemos comenzando, empezaremos a usar unidades absolutas.

Conforme nuestra habilidad sea mayor, le cogeremos el gusto a las unidades relativas

# **Unidades absolutas**

# Unidades absolutas

- Las unidades absolutas son las más sencillas. Ya que no cambian
- Se trata de unidades que son siempre iguales a una unidad de medida física.
- La más conocida con diferencia es el píxel, una medida fácil para comenzar a trabajar en CSS.

# Unidades absolutas

- Son ideales en contextos donde las medidas no varían como pueden ser en medios impresos (documentos, impresiones, etc...)
- Poco adecuadas para la web, ya que no tienen la capacidad de adaptarse a diferentes resoluciones o pantallas

# Unidades absolutas

## Tipos de unidades absolutas

| Unidad | Significado  | Medida aproximada |
|--------|--------------|-------------------|
| in     | Pulgadas     | 1in = 25.4mm      |
| cm     | Centímetros  | 1cm = 10mm        |
| pc     | Picas        | 1pc = 4.23mm      |
| mm     | Milímetros   | 1mm = 1mm         |
| pt     | Puntos       | 1pt = 0.35mm      |
| px     | Píxels       | 1px = 0.26mm      |
| Q      | Cuarto de mm | 1Q = 0.248mm      |

# Unidades absolutas

## Ejemplos con unidades absolutas

```
/* El cuerpo de la página debe mostrar un margen de media pulgada */  
body { margin: 0.5in; }
```

```
/* Los elementos <h1> deben mostrar un interlineado de 2 centímetros */  
h1 { line-height: 2cm; }
```

```
/* Las palabras de todos los párrafos deben estar separadas 4 milímetros entre si */  
p { word-spacing: 4mm; }
```

```
/* Los enlaces se deben mostrar con un tamaño de letra de 12 puntos */  
a { font-size: 12pt; }
```

```
/* Los elementos <span> deben tener un tamaño de letra de 1 pica */  
span { font-size: 1pc; }
```

# Unidades absolutas

La principal ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar, sin necesidad de realizar cálculos intermedios. Su principal desventaja es que son muy poco flexibles y no se adaptan fácilmente a los diferentes medios

# Unidades absolutas

De todas las unidades absolutas, las más usadas son **px** para pantalla y **pt** para impresión.

Si, se pueden establecer hojas de estilo para imprimir, para eso está el atributo media.

# Unidades absolutas

## CSS3 Media Types

| Value  | Description   |
|--------|---|
| all    | Used for all media type devices                       |
| print  | Used for printers                                     |
| screen | Used for computer screens, tablets, smartphones etc.  |
| speech | Used for screenreaders that "reads" the page out loud |

# Unidades absolutas

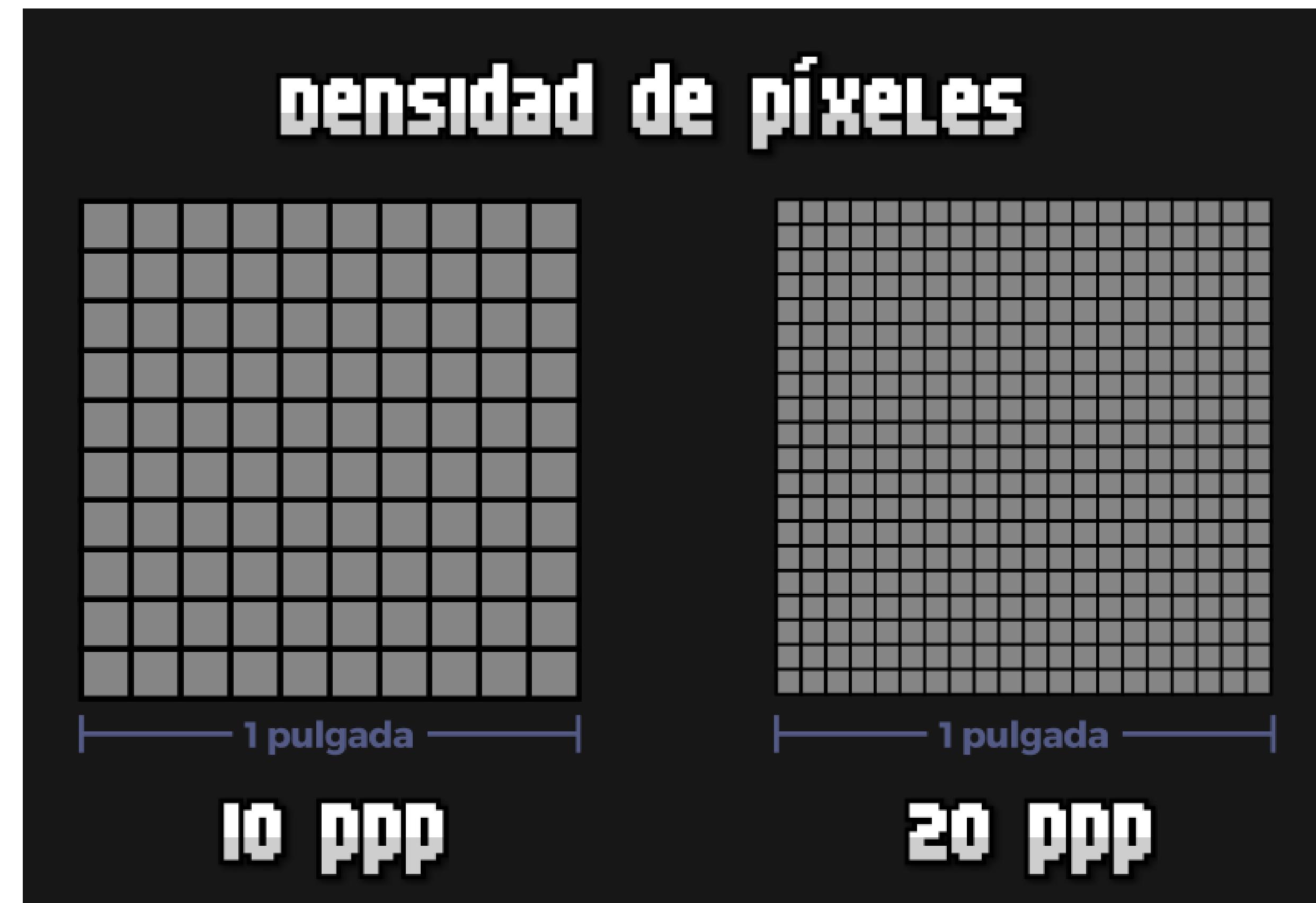
```
<link rel="stylesheet" href="css/estilos.css" media="screen">
```

# Unidades absolutas

**Demo con width, height y  
font-size  
article, div y p**

# Unidades absolutas

## Ojo con densidad de píxeles



# Unidades relativas

# Unidades relativas

- Las unidades relativas son un tipo de medida mucho más potente y habitual en el CSS que creamos generalmente.
- Al contrario que las unidades absolutas, las unidades relativas dependen de algún otro factor

# Unidades relativas

## Tipos de unidades relativas

| Unidad     | Medida aproximada   | Ejemplo  |
|------------|---|--|
| %          | Relativa a herencia (concretamente, al elemento padre)    | 50% = mitad del parent                           |
| em / rem   | Relativo al <code>font-size</code> en ese elemento        | $1.5\text{em} = 1.5 * 16\text{px} = 24\text{px}$ |
| ex / rex   | Relativo a la altura del carácter x minúscula             | $1\text{ex} \sim 0.5\text{em}$                   |
| ch / rch   | Relativo al ancho del carácter 0                          | $1\text{ch} \sim 1 \text{ carácter}$             |
| cap / rcap | Relativo a la altura del primer carácter en mayúscula     |  |
| ic / ric   | Relativo al ancho del glifo 水 CJK (Chino-Japones-Koreano) |  |
| lh / rlh   | Relativo al <code>line-height</code> en ese elemento      |  |

# Unidades relativas

Probablemente, las más conocidas son %, em y rem, por lo que vamos a empezar con ellas.

# Unidades relativas

## La unidad de porcentaje %

Los porcentajes llevan mucho tiempo en el universo de CSS, sin embargo requieren cierta destreza y deben usarse con cuidado en nuestras primeras fases de aprendizaje, ya que el valor de porcentaje establecido depende del tamaño del elemento **padre inmediato**.

# Unidades relativas

Demo con width, height y  
font-size  
article, div y p

# Unidades relativas

## La unidad em

La unidad em se utiliza para definir un tamaño en base al tamaño que tiene establecido el navegador en la tipografía (tamaño de la fuente) en el elemento HTML actual donde se aplica. Por defecto, y para simplificar, se puede asumir que 1em es un valor aproximado a **16px**.

# Unidades relativas

## La unidad em



# Unidades relativas

## La unidad em

Imaginemos que el tamaño de la fuente establecida en el navegador del usuario es exactamente 16px. Una cantidad 1em equivaldría a 16px, mientras que una cantidad de 2em sería justo el doble: 32px. Por otro lado, una cantidad de 0.5em sería justo la mitad: 8px.

# Unidades relativas

Demo con em

# Unidades relativas

## La unidad rem (root element)

Muchas de las unidades que vamos a ver en esta sección, están prefijadas por la letra r. Esta letra simboliza la palabra root (raíz), por lo que en este caso estamos hablando de unidades em en el elemento raíz.

# Unidades relativas

## La unidad rem (root element)

Esta unidad toma la idea de la unidad em, pero en lugar de tomar el tamaño font-size del elemento actual, toma el tamaño del elemento raíz, es decir, el elemento <html> o la pseudoclase :root. Esta forma nos permite trabajar con múltiplos del tamaño base.

# Unidades relativas

## La unidad rem (root element)

```
:root {  
    font-size: 22px;      /* Tamaño base */  
}  
  
h1 {  
    font-size: 2rem;     /* El doble: 2 x 22px = 44px */  
}  
  
h2 {  
    font-size: 1rem;     /* El mismo: 1 x 22px = 22px */  
}
```

# Unidades relativas

**Demo con rem**

# Unidades relativas

## La unidad ex

La unidad ex es una unidad menos conocida, que hace referencia al tamaño de alto de la primera letra minúscula de la tipografía. Por simplificar, podemos asumir que 1ex es aproximadamente 0.5em, es decir, la mitad del tamaño del font-size del elemento actual.

# Unidades relativas

## La unidad ex

La unidad ex es una unidad menos conocida, que hace referencia al tamaño de alto de la primera letra minúscula de la tipografía. Por simplificar, podemos asumir que 1ex es aproximadamente 0.5em, es decir, la mitad del tamaño del font-size del elemento actual.

# Unidades del viewport

# Unidades del viewport

- Existen unas unidades de nueva generación que resultan muy útiles, porque dependen del **viewport** (región visible de la página web en el navegador). Con estas unidades podemos hacer referencia a un porcentaje concreto del tamaño específico que tengamos en la ventana del navegador.

# Unidades del viewport

- Es decir, cada vez que hagamos referencia a una unidad precedida por v estamos haciendo referencia a un % del tamaño total de la ventana. De esta forma, podemos hacer que elementos concretos tengan valores proporcionales al tamaño de la ventana del navegador.

# Unidades del viewport

| Unidad | Significado      | Medida aproximada                       | Más info                                |
|--------|------------------|---|---|
| vw     | viewport width   | 1vw = 1% del ancho del navegador        |   |
| vh     | viewport height  | 1vh = 1% del alto del navegador         |   |
| vmin   | viewport minimum | 1vmin = 1% del alto o ancho (el mínimo) |   |
| vmax   | viewport maximum | 1vmax = 1% del alto o ancho (el máximo) |   |
| vi     | viewport inline  | Versión lógica inline de vw y/o vh      | <a href="#">Ver propiedades lógicas</a> |
| vb     | viewport block   | Versión lógica en bloque de vw y/o vh   | <a href="#">Ver propiedades lógicas</a> |

# Unidades del viewport

## La unidad vw (viewport width)

La unidad vw es un tipo de unidad que hace referencia al ancho del viewport (región visible de la página en el navegador). Por ejemplo, si definimos 50vw, estamos indicando un 50% del ancho actual del navegador.

# Unidades del viewport

Demo

# Unidades del viewport

## La unidad vh (viewport height)

Por su parte, la unidad vh hace referencia de la misma forma al alto del viewport. Si con la unidad vw podemos hacer referencia al ancho, con vh hacemos referencia al alto.

# Unidades del viewport

En este ejemplo, mientras 50vw referencia al 50% del ancho del navegador, con la propiedad height a 75vh estaremos haciendo referencia al 75% del alto del navegador. Si nuestro navegador tiene un tamaño de 1080px de alto, con 75vh estaríamos indicando unos 810px.

```
.container {  
    width: 50vw;  
    height: 75vh;  
}
```

# Unidades del viewport

**La unidad vmin (viewport min)  
y vmax (viewport max)**

# Unidades del viewport

**Demo**

# Unidades del viewport

Consejo

# Unidades del viewport

La elección de unidades es algo muy personal, pero en estos apuntes se intentará utilizar siempre las siguientes unidades:

- **px** para distancias pequeñas (anchos de bordes, márgenes, etc.).
- **rem** para tamaños de fuente, salvo que se quiera aprovechar en el diseño la composición de tamaños, en cuyo caso se usarán porcentajes %.
- **vw, vh y porcentajes %** para tamaños de elementos grandes y distribución de elementos por la pantalla.

# Funciones de cálculo

# Funciones de cálculo

## FUNCIONES CSS MATEMÁTICAS

En este artículo vamos a ver ciertas funciones CSS matemáticas, que no son más que funciones de apoyo que podemos utilizar en CSS para realizar cálculos u operaciones sencillas de una forma fácil y sencilla, sin tener que abandonar CSS y/o recurrir a Javascript.

# Funciones de cálculo

| Función CSS   | Descripción   |
|---|---|
| <strong>Funciones de cálculos</strong>                                      |   |
| <code>calc()</code>   | Permite <b>calcular operaciones</b> con unidades CSS como <code>px</code> , <code>%</code> , <code>vw</code> , <code>vh</code> u otras (incluso combinadas) |
| <strong>Funciones de comparación</strong>                                   |   |
| <code>min()</code> / <code>max()</code>                                     | Permite calcular el valor <b>mínimo o máximo</b> de las unidades indicadas.   |
| <code>clamp()</code>  | Permite calcular valores «ajustados». Equivalente a <code>max(MIN, min(VAL, MAX))</code> .  |
| <strong>Funciones escalonadas</strong>                                      |   |
| <code>round()</code> / <code>mod()</code> / <code>rem()</code>              | Funciones que permiten redondear, obtener el módulo o el resto.   |
| <strong>Funciones trigonométricas</strong>                                  |   |
| <code>sin()</code> / <code>cos()</code> / <code>tan()</code>                | Permite obtener el <b>seno, coseno o tangente</b> de un valor.  |
| <code>asin()</code> / <code>acos()</code> / <code>atan()</code>             | Permite obtener el <b>arcoseno, arcocoseno o arcotangente</b> de un valor.  |
| <code>atan2()</code>  | Aplica la función <b>arcotangente de dos parámetros</b> .   |
| <strong>Funciones exponenciales</strong>                                    |   |
| <code>pow()</code> / <code>sqrt()</code> / <code>hypot()</code>             | Permite realizar potenciación, raíces cuadradas o hipotenusa.   |
| <code>log()</code> / <code>exp()</code>                                     | Permite realizar logaritmos o potencias de <code>e</code> .   |
| <strong>Otras</strong>  |   |
| <code>e</code> / <code>pi</code> / <code>Infinity</code> / <code>NaN</code> | Palabras clave que representan valores matemáticos. <a href="#">Ver constantes o NaN</a>  |
| <code>attr()</code>   | Permite obtener el <b>valor de un atributo</b> desde CSS.   |

# Funciones de cálculo

## Función calc()

Es posible que en algunas ocasiones necesitemos indicar valores precalculados por el navegador, ya porque sea más cómodo o porque simplemente queremos hacer el código más organizado o fácil de mantener. Por ejemplo, la suma de dos valores que a priori desconocemos o no sabemos exactamente cuánto suman, pero que el navegador si puede calcularlo fácilmente.

# Funciones de cálculo

```
.elemento {  
    width: calc(200px + 1em);  
    height: 300px;  
    background: blue;  
}
```

# Funciones de cálculo

Como se puede ver, se pueden usar operaciones como: sumas ( + ), restas ( - ), multiplicaciones ( \* ) o divisiones ( / ) que utilicen alguna de las unidades soportadas por CSS, como por ejemplo, números, dimensiones, porcentajes, tiempos, ángulos, etc...

# Funciones de cálculo

## Función min() y max()

En algunas ocasiones, necesitaremos que en una propiedad se aplique un valor entre varias posibilidades posibles, y lo que te interesa es el valor más pequeño.

CSS permite realizar esto utilizando la función min() que nos permite elegir el valor más pequeño de 2 o más posibilidades aplicadas por parámetro:

```
.elemento {  
    width: min(200px, 25%);  
    height: 200px;  
    background: red;  
}
```

# Funciones de cálculo

Esto aplicaría un width de 200px, siempre y cuando el 25% del elemento padre sea más grande de 200px. Todo este cálculo, generalmente realizado en Javascript debido a su cualidad de lógica de cálculo, puede realizarse desde CSS si sólo se utiliza exclusivamente para esta finalidad.

Recuerda que tanto la función min() como la función max() permite tener más de 2 parámetros, no tiene que tener obligatoriamente 2 parámetros.

# Colores en CSS

# Colores en CSS

Hay múltiples formas de definir un color en CSS, algunas más sencillas, otras más complejas.

# Colores en CSS

Antes de nada vamos a ver 2 propiedades que os va a servir para poder hacer pruebas.

color

background-color

# Colores en CSS

| Propiedad        | Valor | Significado  |
|------------------|-------|--|
| color            | COLOR | Cambia el <b>color del texto</b> que está en el interior de un elemento. |
| background-color | COLOR | Cambia el <b>color de fondo</b> de un elemento.                          |

```
.element {  
    background-color: black; /* Color de fondo */  
    color: white;           /* Color de texto */  
}
```

# Colores en CSS

## Formas de establecer colores

| Esquema   | Descripción   | Más info  |
|---|---|---|
| <b>Espacio de color RGB</b>                           |   |   |
| red   | Establece un color mediante palabras clave.   | <a href="#">Ver palabras clave predefinidas</a> |
| rgb()   | Usa una función <code>rgb()</code> (rojo, verde y azul).  | <a href="#">Ver función RGB</a>                 |
| #rrggbb   | Notación RGB abreviada en hexadecimal. <b>Notación recomendada</b>                              | <a href="#">Ver formato Hexadecimal</a>         |
| hsl()   | Usa una función <code>hsl()</code> (color, saturación y brillo).                                | <a href="#">Ver función HSL</a>                 |
| hwb()   | Usa una función <code>hwb()</code> (color, claridad y oscuridad).                               | <a href="#">Ver función HWB</a>                 |
| <b>Espacio de color independiente del dispositivo</b> |   |   |
| lab()   | Usa una función <code>lab()</code> y <code>oklab()</code> (luminosidad CIE, eje A y eje B).     | <a href="#">Ver función LAB y OKLAB</a>         |
| lch()   | Usa una función <code>lch()</code> y <code>oklch()</code> (luminosidad CIE, saturación, color). | <a href="#">Ver función LCH y OKLCH</a>         |

# Colores en CSS

## Palabras clave de color

# Colores en CSS

## Función RGB

# Colores en CSS

## Formato Hexadecimal

# Colores en CSS

Demo

# <Despedida>

Email

**bienvenidosaez@gmail.com**

Instagram

**@bienvenidosaez**

Youtube

**youtube.com/bienvenidosaez**

**CONQUERBLOCKS**