

HTML + CSS



6

Episodio VI

El retorno de las unidades de medida



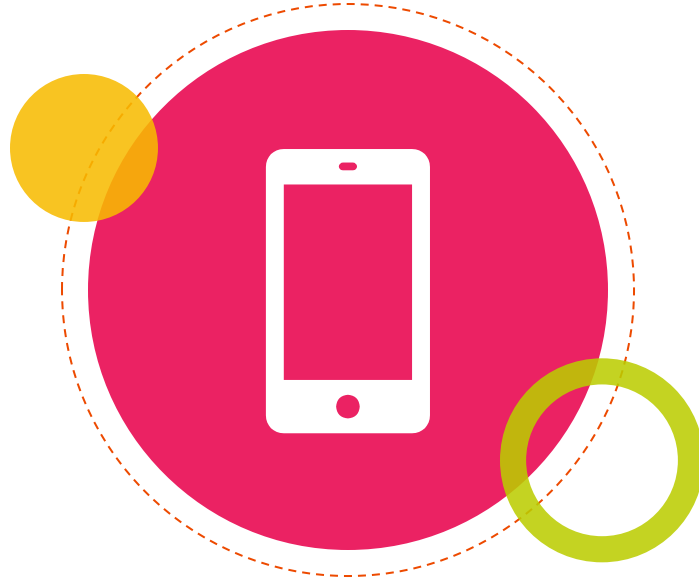


Contenido

🎯 Responsive web design

- Mobile First
- Meta / Viewport
- Medidas relativas
 - 📏 %
 - 📏 vw
 - 📏 vh
 - 📏 em
- Media Queries

Mobile First

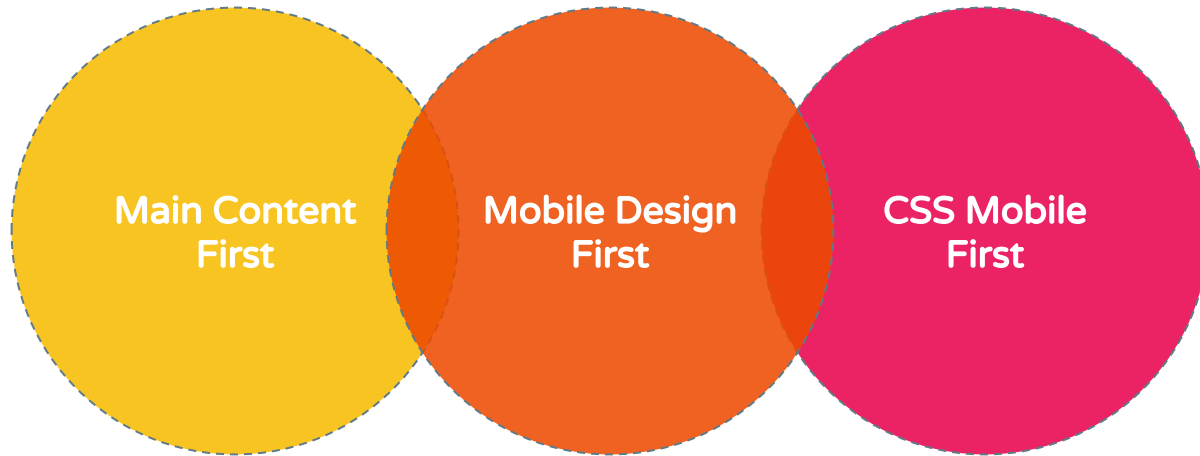


Estructurar la web **pensando primero**
en un **dispositivo móvil.**

Mobile First



Mobile First





Tag para RWD

El tag **<meta> viewport** da al browser instrucciones de cómo se debe dimensionar y escalar la página web al cargarse.

```
<meta  
  name="viewport"  
  content="width=device-width, initial-scale=1"  
>
```

Esta es la estructura básica de este tag, en ocasiones puede que lo veamos con más información.



Practiquemos

Realizar el punto **#1** de la guía de ejercicios.

Unidades de medida relativas



Medidas que están relacionadas con su padre contenedor directo.

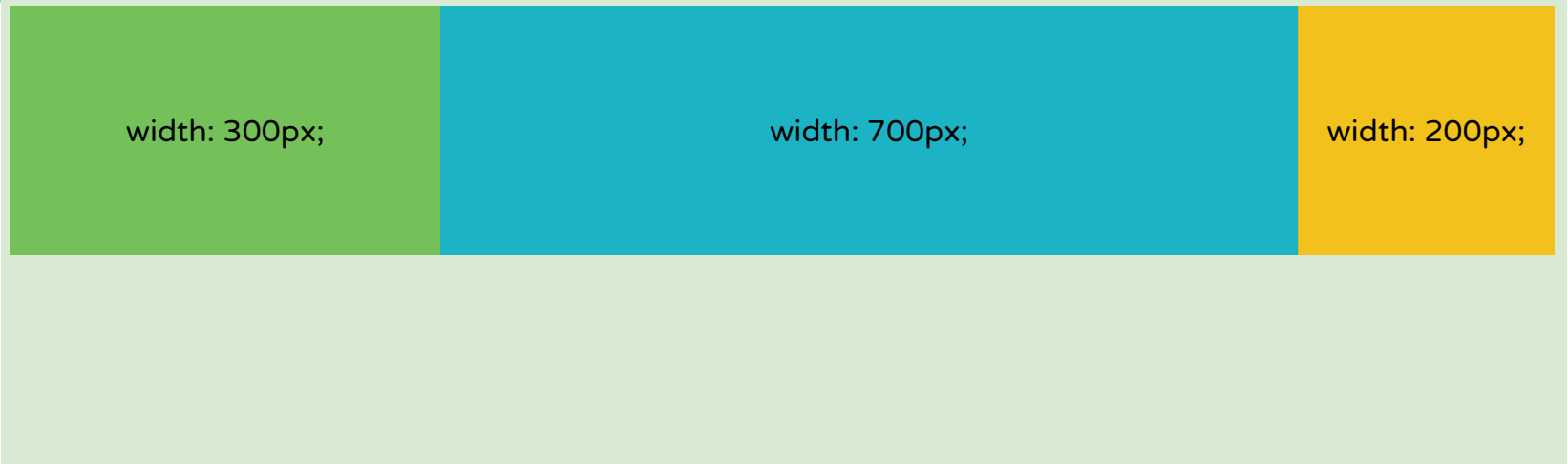


Los Porcentajes – %

Cualquier medida expresada en porcentaje, **SIEMPRE** estará relacionada con la medida (en ese mismo eje) del **elemento padre** que la contiene.

```
.box{  
  width: 25%;  
}
```

- Generalmente los altos no se trabajan en medidas porcentuales.

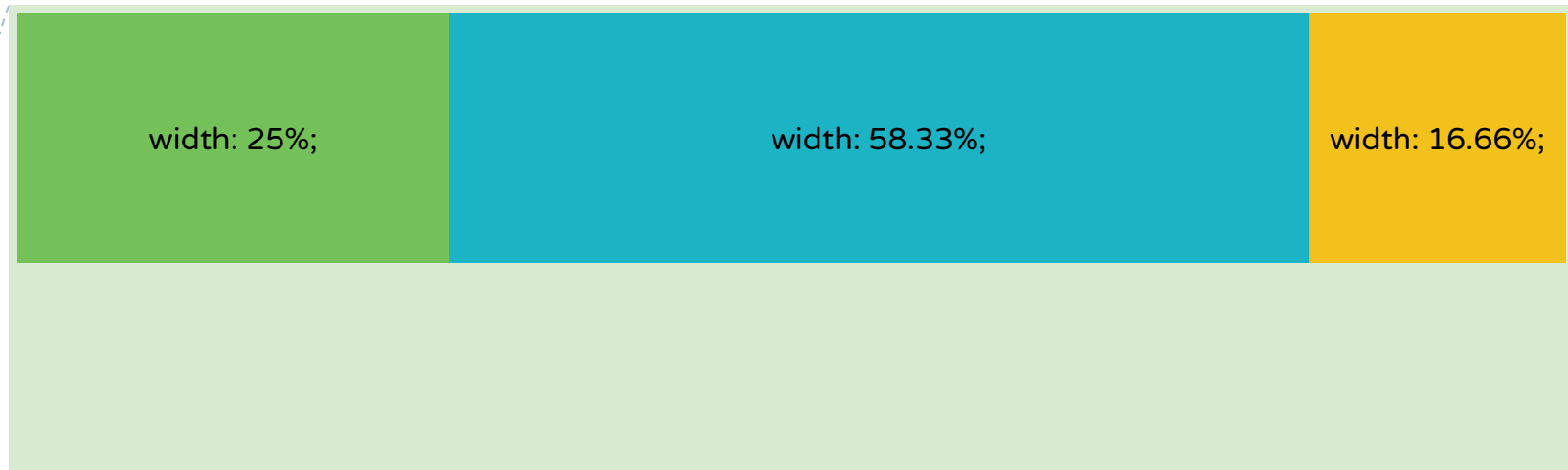


width: 300px;

width: 700px;

width: 200px;

contenedor padre con **1200px** en width



contenedor padre con **1200px** en **max-width**

REGLA DE 3 SIMPLE

width: 25%;
 $(300/1200)*100$

width: 58.33%;
 $(700/1200)*100$

width: 16.66%;
 $(200/1200)*100$

En este caso nuestra “base” es **1200px**

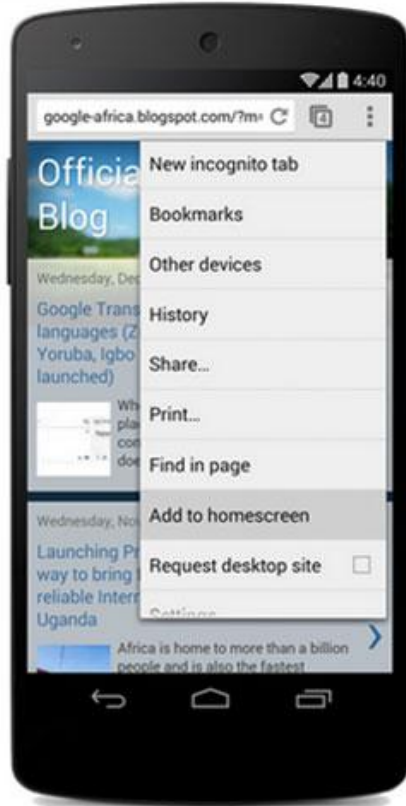
Viewport Measures – vw / vh

Cualquier medida expresada en **vw/vh** tomará **SIEMPRE** como eje referencial **al viewport** del documento.

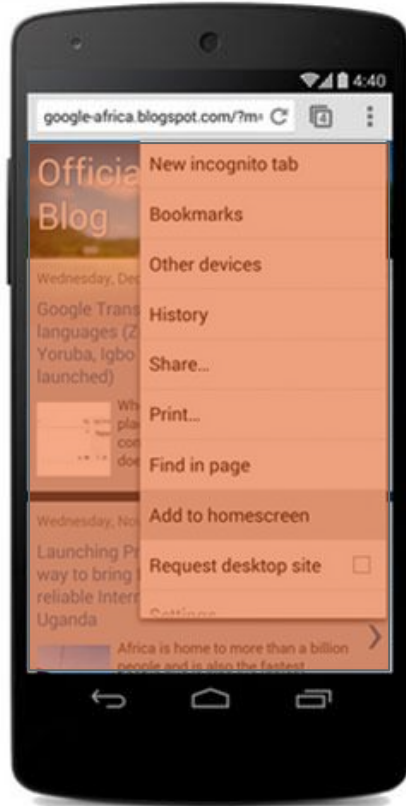
```
.box{  
  width: 25vw;  
  height: 50vh;  
}
```

- **Viewport** hace referencia a la “caja visible” de contenido dentro de un navegador.

El Viewport



El Viewport






A diagram showing a large gray rectangle representing a viewport. Inside the top-left corner of the gray rectangle is a smaller blue rectangle. The blue rectangle contains the text 'width: 30vw; height: 50vh;'. To the right of the gray rectangle, the text 'viewport height de 450px' is written vertically. Below the gray rectangle, the text 'viewport width de 320px' is written horizontally. The entire diagram is framed by a dashed blue circle with various colored circles (blue, green, yellow, orange, pink) scattered around it.

width: 30vw;
height: 50vh;

viewport height de 450px

viewport width de 320px

En este caso, la caja tendrá 0.3 veces el tamaño de ancho del viewport (**96px**) y 0.5 veces el alto del viewport (**225px**)



A diagram illustrating the use of viewport units in CSS. It features a large gray rectangle representing the viewport. Inside the top-left corner of this rectangle is a smaller teal rectangle. The teal rectangle contains the text 'width: 30vw; height: 50vh;'. To the right of the gray rectangle, the text 'viewport height de 250px' is written vertically in red. Below the gray rectangle, the text 'viewport width de 660px' is written horizontally in red. At the bottom of the diagram, a paragraph of text explains the relationship between the box size and the viewport dimensions.

width: 30vw;
height: 50vh;

viewport height de 250px

viewport width de 660px

En este caso, la caja tendrá 0.3 veces el tamaño de ancho del viewport (**198px**) y 0.5 veces el alto del viewport (**125px**)



Los em's – em

Los em's son **unidades** de medida que se **recomiendan usar** en todo lo relacionado con **tipografías**.

1em comenzará siendo igual a **16px** (a menos que seteemos lo contrario)

```
p{  
  font-size: 1em;  
  line-height: 1.5em;  
}
```

- Toda medida (que no sea el font-size) expresada en em's tomará **SIEMPRE** como base referencial el **font-size** que tenga el mismo elemento que estemos modificando.

```
<body>  
  <div>  
    <p>  
      <strong><strong>  
    </p>  
  </div>  
</body>
```

font-size: 16px;
font-size: 1em;
font-size: 1em;
font-size: 1em;

`<body>` tiene 16px en font-size, los demás elementos por default vienen con 1em en font-size.

Por lo tanto, todos los demás elementos tienen 16px de font-size.

```
<body>  
  <div>  
    <p>  
      <strong><strong>  
    </p>  
  </div>  
</body>
```

font-size: 16px;
font-size: 2em;
font-size: 1em;
font-size: 1em;

Si cambiamos el font-size de `<div>` a 2em, éste tendrá 2 veces el tamaño del font-size de su padre.

`<div>` ahora tiene 32px de font-size.

Por lo tanto los hijos de `<div>` ahora cambiaron con solo haber tocado a su padre.

Los hijos de `<div>` quedaron con 32px en font-size.

```
<body>  
  <div>  
    <p>  
      <strong><strong>  
    </p>  
  </div>  
</body>
```

font-size: 16px;

font-size: 2em;

font-size: 1.5em;

font-size: 1em;

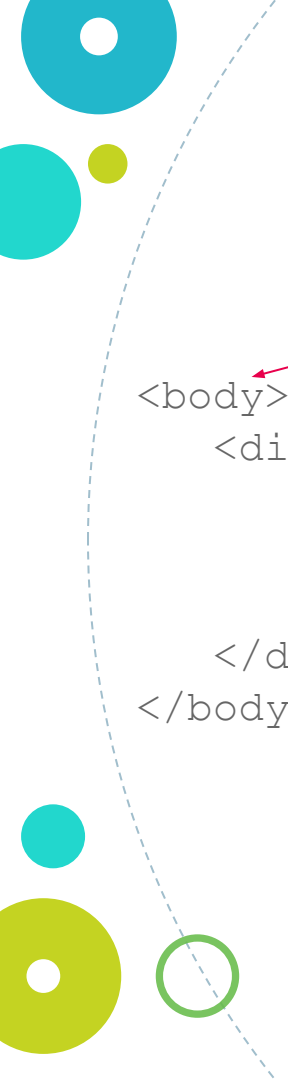
<div> tiene 32px de font-size.

Si a <p> le decimos font-size: 1.5em, <p> tendrá 1.5 veces el tamaño que tenga el font-size de su padre.

<p> ahora tiene 48px en font-size.

 tiene 1 vez lo que tiene su padre en font-size.

 tiene 48px en font-size.



```
<body>
  <div>
    <p>
      <strong><strong>
    </p>
  </div>
</body>
```

font-size: 10px;

font-size: 2em;

font-size: 1.5em;

font-size: 1em;

Si ahora cambiamos la base del documento (<body>) y la llevamos a 10px en font-size. Todos los elementos cambian de tamaño sin haber "tocado" su font-size.

<div> = 20px de font-size

<p> = 30px de font-size

 = 30px de font-size





Practiquemos

Realizar el punto #2 de la guía de ejercicios.



Los Media Queries

Conjunto de reglas CSS que permiten re-organizar el contenido dependiendo de las condiciones de visualización del documento.

Siempre se deben escribir al final de nuestra hoja de CSS.



Media-querie para Mobile **First**

```
@media (min-width: 460px) {  
    /* reglas de CSS */  
}
```

Al especificar **min-width**, estamos diciendo “si como mínimo hay Npx de ancho, apliquemos esto” algo como “desde este punto hacia arriba”.



Media-querie para Mobile Last

```
@media (max-width: 960px) {  
    /* reglas de CSS */  
}
```

Al especificar **max-width**, estamos diciendo “si como máximo hay Npx de ancho, apliquemos esto” algo como “desde este punto hacia abajo”.



Media-querie definiendo la **orientación**

```
@media (max-width: 768px) and (orientation: portrait){  
    /* reglas de CSS */  
}
```

Al especificar **la orientación (portrait o landscape)**, estamos diciendo “si como máximo hay Npx de ancho y además el dispositivo está en posición vertical/horizontal, apliquemos esto”.



Practiquemos

Realizar el punto #3 de la guía de ejercicios.

Gracias!



¿Consultas?