

1. UNIDADES RELATIVAS

Estas unidades dependen del entorno o de otros elementos y permiten diseños más flexibles y adaptativos.

1.1. EM (RELATIVO AL ELEMENTO PADRE)

- **Descripción:** Se basa en el tamaño de fuente del elemento padre.
- **Ejemplo:**

```
p {  
  font-size: 1.5em; /* 1.5 veces el tamaño de la fuente del padre */  
}
```

- **Diferencia clave:** Se acumula si se usa en elementos anidados.

```
div {  
  font-size: 2em; /* 2 veces el tamaño base */  
}  
  
p {  
  font-size: 1.5em; /* 1.5 veces el tamaño del div (2em) => 3em total */  
}
```

1.2. REM (RELATIVO AL HTML)

- **Descripción:** Se basa en el tamaño de fuente del elemento raíz (html).
- **Ejemplo:**

```
html {  
  font-size: 16px;  
}  
  
p {  
  font-size: 1.5rem; /* 1.5 veces el tamaño base (16px) => 24px */  
}
```

- **Ventaja:** No se acumula con elementos anidados, lo que evita efectos no deseados.

1.3. % (RELATIVO AL ELEMENTO PADRE)

- **Descripción:** Define un tamaño basado en un porcentaje del tamaño del elemento contenedor.
- **Ejemplo:**

```
div {  
  
  width: 50%; /* 50% del ancho del contenedor padre */  
  
}
```

- **Uso común:** Dimensiones de cajas (width, height).

1.4. VW Y VH (VIEWPORT WIDTH/HEIGHT)

- **Descripción:** Se basan en el tamaño de la ventana gráfica (viewport).
- 1vw = 1% del ancho de la ventana.
- 1vh = 1% de la altura de la ventana.
- **Ejemplo:**

```
div {  
  
  width: 50vw; /* 50% del ancho de la ventana */  
  height: 100vh; /* 100% de la altura de la ventana */  
  
}
```

1.5. VMIN Y VMAX

- **Descripción:**
 - vmin: Basado en el menor valor entre vw y vh.
 - vmax: Basado en el mayor valor entre vw y vh.
- **Ejemplo:**

```
div {  
  
  width: 50vmin; /* 50% del menor entre el ancho o alto de la ventana */  
  
}
```

2. UNIDADES ABSOLUTAS

Son fijas y no dependen de otros elementos, lo que las hace menos flexibles.

2.1. PX (PÍXELES)

- **Descripción:** Define un tamaño fijo en píxeles.
- **Ejemplo:**

```
p {  
  font-size: 16px;  
}
```

- **Ventaja:** Predecible y consistente.
 - **Desventaja:** No es adaptable a diferentes pantallas de forma automática.
-

2.2. PT (PUNTOS)

- **Descripción:** Usado principalmente en impresión. 1pt = 1/72 de pulgada.
- **Ejemplo:**

```
p {  
  font-size: 12pt;  
}
```

2.3. CM, MM, IN (CENTÍMETROS, MILÍMETROS, PULGADAS)

- **Descripción:** Se basan en medidas físicas reales.
- **Ejemplo:**

```
div {  
  width: 10cm;  
}
```

- **Uso limitado:** Principalmente para medios impresos.
-

3. UNIDADES FLEXIBLES (GRID Y FLEXBOX)

3.1. FR (FRACCIÓN)

- **Descripción:** Representa una fracción del espacio disponible en un contenedor de cuadrícula (grid).
- **Ejemplo:**

```
.grid {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
}
```

- 1fr = 1 parte, 2fr = 2 partes.

3.2. AUTO

- **Descripción:** El tamaño se ajusta automáticamente al contenido.
- **Ejemplo:**

```
div {  
  width: auto;  
}
```

COMPARACIÓN Y USOS RECOMENDADOS

Unidad	Usos Comunes	Ventajas	Desventajas
px	Bordes, padding, márgenes	Consistencia	No es flexible
em	Tipografía, padding, márgenes	Escalable	Puede acumularse
rem	Tipografía, diseño general	Predecible, evita acumulación	Requiere definir tamaño raíz
%	Ancho y alto de contenedores	Flexible	Depende del padre
vw, vh	Elementos a pantalla completa	Adaptable al viewport	Puede ser inconsistente en móviles
fr	Grid layouts	Distribución precisa del espacio	Solo funciona en grid
auto	Dimensiones dinámicas	Simple y adaptable	Menos control

¿Cuál usar y cuándo?

- **Diseños fluidos y responsivos:**
 - Usa rem, %, vw, vh y fr.
- **Tamaños fijos o precisos:**
 - Usa px, cm o mm para impresión o pixel-perfect designs.
- **Tipografía:**
 - Usa rem para mantener consistencia y em si quieres escalar basándote en el contenedor.