

Neurri unitateak

CSSetan bi luzera erabiltzen dira: relatiboa eta absolutua.

Luzera absolutuak			Luzera relatiboak	
Unitatea	Izena	Baliokidetasuna	Unitatea	Baliokidetasuna
cm	Zentimetroak	1cm = 96px/2,54	em	Edukiontziazen letra-tamaina.
mm	Milimetroak	1mm = 1cm-ko 1/10	ex	Uneko letra-tipoaren x altuerari dagokio (gutxi erabiltzen da).
Q	Milimetroaren laurdena	1Q = 1cm-ko 1/40	ch	"0" (zero) zabalerari dagokio.
in	Hazbetekoa	1in = 2,54cm = 96px	rem	root elementuaren letra-tamainari dagokio.
pc	Pikak	1pc = lin-ko 1/6	vw	Leihoaren zabaleraren %1.
pt	Puntuak	1pt = lin-ko 1/72	vh	Leihoaren altueraren %1.
px	Pixelak	1px = lin-ko 1/96	vmin	Leihoaren dimentsio txikiaren % 1.
			vmax	Leihoaren dimentsio handienaren % 1.

Garrantzitsua da desberdintasuna ezagutzea, gauzek zer dimentsio izango dituzten ulertzeko.

Neurri unitateak

Luzera absolutuko unitateak

Oro har, beti tamaina berdina dutela jotzen da.

Hau da modurik errazena, elementu baten tamaina zehatza adierazten da pantaila edozein dela ere: resoluzio handiko pantaila duen ordenagailu batean, tableta edo mugikorra.

Balio hauetako gehienak erabilgarriagoak izaten dira diseinua inprimatutako formatuan ateratzen direnean pantailaren irteeran erabiltzen direnean baino.

Milimetroak, zentimetroak, ... erabil daitezke, baina maiz erabiltzen den balioa *px* (pixelak) da.

Luzera erlatiboko unitateak

Luzera erlatiboko unitateak beste zerbaiti buruzkoak dira: Edukiontzi elementuaren tamainara, elementu nagusiaren letra tamainara, leiho grafikoaren tamainara, etab.

Unitate erlatiboek elementuei balio zehatza ez ematea ahalbidetuko dute.

Unitate erlatiboak erabiltzearen abantaila da, arretaz planifikatuta, lor daitekeela testua edo beste elementu batzuk orriko gainerako guztiarekin eskalatzea, egungo dispoitiboetara egokituz.

Ehunekoak

CSS propietate askok ehuneko-balioak har ditzakete, betiere luzerak horien elementu edukiontziaren tamainari dagokionez zehazteko.

Hainbat luzera propietate ehunekoak erabili ahal dituzte, hala nola *width*, *margin* eta *padding*.

Hurrengo adibidean ehunekoak erabiltzen dira *width* balio baten, nabaritu nola balioa kalkulatu den elementu nagusiaren *width* atributuaren ehunekoaren arabera.

Lehenengo kutxaren tamaina pixel-unitatetan zehaztuta dago, eta bigarrena, ehuneko-unitateetan.

Nabigatzailearen leihoaren zabalera minimizatu funtzionamendua ikusteko. Lehenengo kutxak bere tamaina mantentzen duen bitartean, bigarrena edukiontzi nagusiaren tamainara egokitzen da, kasu honetan elementu raizea.

```
<div class="box px">450px wide box</div>  
<div class="box percent">50% wide box</div>
```

```
.box {  
  background: lightgray;  
  margin: 4px 0;  
}  
.px {  
  width: 450px;  
}  
.percent {  
  width: 50%;  
}
```

450px wide box

50% wide box

Bigarren adibide honen desberdintasuna, bi kutxak 500 pixeleko zabalera duen edukiontzi baten barruan daudela da.

Zabalera pixel-unitatetan zehaztuta duen kutxak, aurreko adibidearen zabalera bera du; bigarren kutxa, aldiz, 500 pixelen %50koa da, edukiontziaren tamainara egokitzen.

```
<div class="wrapper">
  <div class="box px">450px wide box</div>
  <div class="box percent">50% wide box</div>
</div>
```

```
.box {
  background: lightgray;
  margin: 4px 0;
}
.px {
  width: 450px;
}
.percent {
  width: 50%;
}
.wrapper {
  width: 500px;
}
```

450px wide box

50% wide box

Hurrengo adibidearen eta aurrekoaren arteko ezberdintasuna da kutxen edukiontziaren tamaina ehuneko balore baten bidez zehaztuta dagoela.

Beraz, leihoaren tamaina aldatzen denean, edukiontziak bere tamaina leihoaren tamaina berriarekiko ere egokitzen du, eta, aldi berean, ehunekotan adierazitako kutxak bere tamaina edukiontziaren tamaina berriarekiko ere egokitzen du.

Bitartean, pixeletan adierazitako kutxak tamaina finkoari eusten dio beti, baita edukiontzia baino handiagoa denean ere.

Nabigatzailearen leihoaren zabalera minimizatu eta azken bi adibideen arteko ezberdintasunak konparatu.

```
<div class="wrapper">
  <div class="box px">450px wide box</div>
  <div class="box percent">50% wide box</div>
</div>
```

```
.box {
  background: lightgray;
  margin: 4px 0;
}
.px {
  width: 450px;
}
.percent {
  width: 50%;
}
.wrapper {
  width: 60%;
}
```

450px wide box

50% wide box

Ehunekoak *font-size* propietatean ere ikus daitezke. Bertan, testua elementu nagusiaren tamainarekin lotuta dago zuzenean.

```
<p>Full size text: 18px</p>
<div class="fs-18-px">
  <p class="fs-50-percent">50% (New size text: 9px)</p>
  <p class="fs-200-percent">200% (New size text: 36px)</p>
</div>
```

```
.fs-18-px {
  font-size: 18px;
}
.fs-50-percent {
  font-size: 50%;
}
.fs-200-percent {
  font-size: 200%;
}
```

Full size text: 18px

50% (New size text: 9px)

200% (New size text: 36px)

REM eta EM

Unitate relatiborik ohikoenetakoak dira, eta testua eta horiekin lotutako proportzioak dimentsionatzeko erabiltzen ohi dira.

Unitate hauek birkalkulatzeko dira definitu diren beste balio batzuekin alderatuta, eta oso garrantzitsuak dira web orrien irisgarritasuna diseinatzeko garaian.

Unitate erlatiboen eta absolutuen arteko korrespondentzia kalkulatzeko hiruko erregela egiten da. Zehaztutako zenbakia aurrez zehaztutako tamaina lehenetsiarekin biderkatuko da, erudian 16px (1em = 16px):

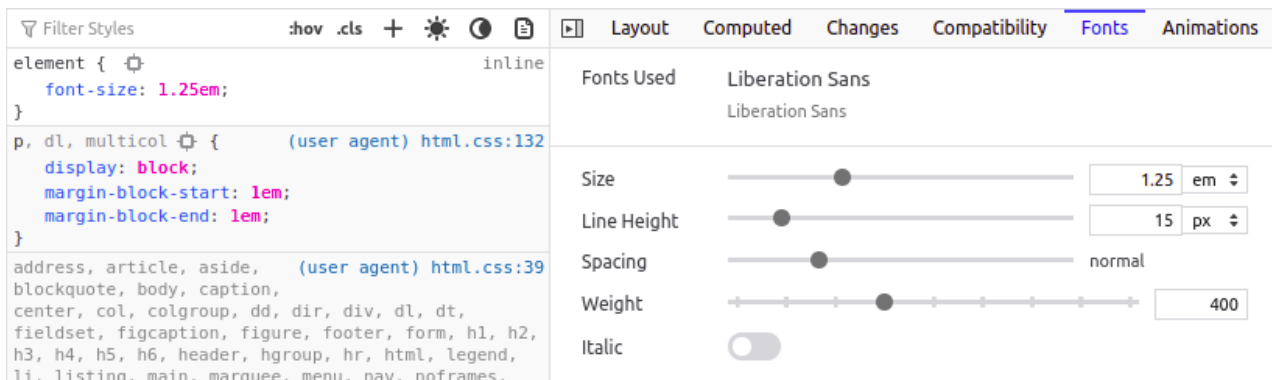
rem/em	pixel	emaitza
1	→ 16	X → $(0,5 * 16) / 1 = 8\text{px}$
0,5	→ X	

Beste balore batzuen eredu:

rem/em **pixel**
 1.25rem/em → $1.25 * 16 = 20\text{px}$
 1.5rem/em → $1.5 * 16 = 24\text{px}$
 2rem/em → $2 * 16 = 32\text{px}$
 2.25rem/em → $2.25 * 16 = 36\text{px}$
 2.5rem/em → $2.5 * 16 = 40\text{px}$
 etabar.

Korrespondentziak kalkulatzeko beste orri asko daude, adibidez: [w3school bihurgailua](#)

Halaber, dimentsioak ikus eta manipula daitezke *firefox* garapenerako tresnaren ikuskatzailetik.



font atalean aukera ematen du korrespondentziak ikusteko zein tamainak manipulatzeko

Nabigatzaile gehienetan, aurrez zehaztutako iturriaren tamaina 16 pixelekoa izaten da, eta unitate erlatiboek oinarri horretatik abiatuta kalkulatu dituzte tamainak.

- HTML etiketarako oinarri hori CSSren bidez aldatzen bada oinarritzko tamaina berri bat ezartzen da, eta orriaren gainerako unitate erlatiboak kalkulatzeko oinarri berria bihurtuko da.
- Era berean, erabiltzaile batek bere nabigatzaileko iturriaren tamaina egokitzen badu, unitate erlatiboak kalkulatzeko oinarri berria erabiltzaileak aukeratutakoa izango da.

Adibidez, ikusmen arazoak dituen batek nabigatzailea konfigura dezake, eta orrialdeak birdimentsionatuko dira, konfigurazioan sartutako balioei buruzko kalkulu guztiak eginez.

REM vs EM

Hurrengo adibidean, bi unitateak (*rem* eta *em*) erabiltzen dira orriaren izenburuen eta parrafoen tamaina ezartzeko. Balore berdina bi edukiontzietan; beraz, hauek dituzten testu guztiak tamaina bera izango dituzte.

Eredurako 16 pixeleko oinarritzko tamaina hartzekotan: **1.25rem/em** → $1.25 * 16 = 20\text{px}$

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="box">
      <h1 class="rem-title">REM based title</h1>
      <p class="rem-paragraph">Sed consectetur congue ipsum ut auctor. In viverra volutpat arcu, in porta sem condimentum eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.</p>
    </div>
    <div class="box">
      <h1 class="em-title">EM based title</h1>
      <p class="em-paragraph">Sed consectetur congue ipsum ut auctor. In viverra volutpat arcu, in porta sem condimentum eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.</p>
    </div>
  </body>
</html>
```

```
style.css
body {
  display: flex;
}
.box {
  flex: 1;
  text-align: center;
  border: 1px solid;
}
.rem-title {
  font-size: 1.25rem;
}
.rem-paragraph {
  font-size: 1.25rem;
}
.em-title {
  font-size: 1.25em;
}
.em-paragraph {
  font-size: 1.25em;
}
```

REM based title

Sed consectetur congue ipsum ut auctor. In viverra volutpat arcu, in porta sem condimentum eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

EM based title

Sed consectetur congue ipsum ut auctor. In viverra volutpat arcu, in porta sem condimentum eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

Ikus dezagun nola *html* elementuaren tamaina aldatzen (CSS bidez edo nabigatzailea konfiguratutzen), eduki guztia oinarri berrira moldatuko den.

```
html { font-size: 10px }
```

10 pixeleko oinarritzat tamaina berria hartzerakoan: **1.25rem/em** → 1 * 16 = **12.5px**

REM based title

Sed consectetur congue ipsum ut auctor. In viverra volutpat arcu, in porta sem condimentum eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

EM based title

Sed consectetur congue ipsum ut auctor. In viverra volutpat arcu, in porta sem condimentum eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

Zein ezberdintasun dago bien artean? Unitateak nahiko antzekoak dira, baina badute funtsezko desberdintasun bat:

- **em** unitateak «nire elementu nagusiaren letra tamaina» esan nahi du.
Hau da, oraingo elementuaren letra tamaina hartzen du erreferentziazat.
- **rem** unitateak «sustrai elementuaren letra tamaina» esan nahi du (*rem* «root em»-etik dator).
Hau da, root elementuaren letra tamaina du erreferentziazat, *<html>* etiketa.

Oraingoan, lehenengo adibidetik abiatuta, soilik testuetan dauden edukiontzien iturriaren tamaina aldatuko da.

```
.box {  
  flex: 1;  
  text-align: center;  
  border: 1px solid;  
  font-size: 10px;  
}
```

REM based title

Sed consectetur congue ipsum ut auctor. In viverra volutpat arcu, in porta sem condimentum eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

EM based title

Sed consectetur congue ipsum ut auctor. In viverra volutpat arcu, in porta sem condimentum eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae.

Bi edukiontzien jabetza aldatu arren, honela ikus daiteke nola:

- *rem* unitatearekin definitutako elementuak bere horretan mantentzen dira, oinarria «root» elementuan definitutako tamaina izaten jarraitzen du.
- *em* unitatearen bidez definitutako testuak, ordea, birdimentsionatu egin dira; izan ere, oinarriak (haiek dituen elementu nagusia) beste dimentsio bat du, eta azken horri egokitzen zaie.

Ikus daiteke azken honetan ere nola testuarekin lotutako proportzioak (line-height, margin, etab.) oinarriaren tamaina berrira egokitzen direla, berez nabigatzaileak unitate relatibo hauen bidez ezarrita dituelako.

```
/* default CSS browser values for all h1 elements */  
h1 {  
  display: block;  
  font-size: 2em;  
  margin-top: 0.67em;  
  margin-bottom: 0.67em;  
  margin-left: 0;  
  margin-right: 0;  
  font-weight: bold;  
}  
try it!
```

```
/* default CSS browser values for all p elements */  
p {  
  display: block;  
  margin-top: 1em;  
  margin-bottom: 1em;  
  margin-left: 0;  
  margin-right: 0;  
}  
try it!
```

Hurrengo ereduak habiatutako zerrenda multzo bat: Hiru zerrenda daude guztira, eta bi adibideek HTML bera dute.

Desberdintasun bakarra da lehenengoak *em* unitatean ezarrita dagoela eta bigarrenak *rem* unitatean.

index1.html

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <ul class="rem-list">
      <li>Item 1</li>
      <li>Item 2</li>
      <li>
        <ul>
          <li>Item 3.1</li>
          <li>
            <ul>
              <li>Item 3.1.1</li>
              <li>Item 3.1.2</li>
            </ul>
          </li>
          <li>Item 3.2</li>
        </ul>
      </li>
    </ul>
  </body>
</html>
```

index2.html

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <ul class="em-list">
      <li>Item 1</li>
      <li>Item 2</li>
      <li>
        <ul>
          <li>Item 3.1</li>
          <li>
            <ul>
              <li>Item 3.1.1</li>
              <li>Item 3.1.2</li>
            </ul>
          </li>
          <li>Item 3.2</li>
        </ul>
      </li>
    </ul>
  </body>
</html>
```

style.css

```
.rem-list li {
  font-size: 1.5rem;
}
.em-list li {
  font-size: 1.5em;
}
```

- Item 1
- Item 2
- - Item 3.1
 - Item 3.1.1
 - Item 3.1.2
 - Item 3.2

- Item 1
- Item 2
- - Item 3.1
 - Item 3.1.1
 - Item 3.1.2
 - Item 3.2

- `` elementu baten barruan *rem* unitate mota duten `` elementuek, sustrai-elementuaren tamaina hartzen dute (`<html>`).

Los elementos `` dentro de un elemento `` con una clase de rems toman su tamaño del elemento raíz (`<html>`).

Honek esan nahi du letraren tamaina ez dela handitzen habia-maila berri bakoitzean

- `` elementu baten barruan *em* unitate mota duten `` elementuek, edukiontzi elementuaren tamaina hartzen dute.

Beraz, elkarren segidako habiatze maila bakoitzean, letra tamaina pixkanaka handitzen da, letra tamaina *1.5em* balorearekin adierazten baita.

Bere elementu edukiontziaren letra tamaina baino 1,5 aldiz handiagoa hartzen duenez maila bakoitzean progresiboki haunditzen joango da.