



# Unidades de medidas no CSS

Você deve ter reparado que usamos o **px** para algumas dimensões de altura e largura de elementos HTML, fazemos isso usando o CSS, por exemplo uma determinada imagem da qual queremos que tenha 500px de largura fazemos o seguinte:

```
img {  
  width: 500px;  
}
```

Mas o que exatamente são esses pixels? Existem outras medidas que podemos utilizar para dimensionar elementos do HTML utilizando o CSS? A resposta é sim, e veremos quais maneiras existem, porém com o foco nas que mais são utilizadas e mais fazem sentido para o dia a dia de quem usa o CSS.

## Medidas absolutas x Medidas relativas

Temos dois tipos de medidas quando falamos de CSS, as absolutas são referências mais físicas, como pixels, centímetros, milímetros, medidas que não dependem de alguma referência "pai" nem nenhuma outra, são apenas medidas absolutas, fixas, e por outro lado temos as medidas relativas que como o próprio nome diz elas possuem uma relatividade de acordo com alguma referência, porém fique tranquilo, nessa aula iremos entender essas duas possibilidades de medidas no CSS.

## Medidas absolutas

### Pixels (px)

Pixels são os pontinhos luminosos da tela do seu monitor, portanto, o pixel é o menor elemento na tela de um dispositivo, porém isso não quer dizer que essa medida no CSS equivale a um pixel da tela do seu dispositivo, mas sim o que chamamos de pixel de referência que na verdade é um pouco maior do que o pixel real da sua tela.

O benefício dessa medida é que ela leva a proximidade da tela em consideração, ao usarmos um smartphone que seguramos mais próximo, o pixel terá o tamanho parecido com o de um monitor mais afastado de nós.

O pixel só é a melhor opção quando tratamos de uma aplicação que será feita para apenas um tipo de dispositivo, por exemplo apenas para notebooks ou desktops, porém a realidade é que precisamos sempre pensar que nosso usuário pode acessar a aplicação tanto do desktop/ notebook quanto de um smartphone, por isso existem medidas melhores do que o pixel.

Quanto a sintaxe não há segredo, basta informar quantos pixels quer que um elemento tenha:

```
div {  
  width: 300px  
}
```

E é claro que podemos usar os pixels para definir tamanhos de fontes, saiba que por padrão o navegador emprega no elemento `<p>` o tamanho de 16 px.

```
p {  
  font-size: 24px;  
}
```

Podemos também definir margin, padding e tudo que envolve medida usando os pixels:

```
div {  
  padding: 32px;  
  margin: 8px 16px;  
  width: 300px;  
  height: 150px;  
}
```

Tudo que envolve medida dentro do CSS podemos usar os pixels e as outras unidades de medidas que veremos nessa aula, o que manda é do que você precisa e o que acha mais fácil para a sua aplicação em questão.

## Points (pt)

Essa é uma unidade mais conhecida pelos designers, mais voltado para tipografia, é uma medida que geralmente é usada para tamanhos de fontes, raramente você verá projetos que utilizam pt para definir tamanhos de fontes, pois ele é usado mais para quando uma fonte deve ter especificamente determinado tamanho, no caso para materiais que serão impressos por exemplo, sua sintaxe é bastante simples, assim como quase tudo no desenvolvimento web:

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

12pt é praticamente o mesmo que 16px, só por curiosidade.

## in (inches/polegadas)

Essa é ainda mais rara de se ver por não ter muito uma necessidade de usar, mas saiba que existe essa forma:

```
label {  
  font-size: 0.17in;  
}
```

0.17in é praticamente igual aos 16px.

## Centímetro e Milímetro (cm / mm)

Existe também a possibilidade de usarmos centímetros e milímetros para determinar dimensões dentro do CSS, assim como algumas unidades vistas aqui, essa não é muito utilizada:

```
h1 {  
  font-size: 1cm;  
}  
  
h2 {  
  font-size: 12mm;  
}
```

## Paica (pc)

Essa também é uma unidade mais voltada para tipografia, mas raramente será usada em seus projetos, é extremamente difícil ver alguém que usa, mas é bom saber que existe.

1pc = 16px

```
p {  
  font-size: 1pc;  
}
```

Veja uma relação entre as unidades absolutas → 1 in = 2,54cm = 25,4mm = 72pt = 6pc

## Medidas relativas

Agora veremos unidades que são mais utilizadas, você verá mais desses tipos e também os pixels, saiba que é mais interessante utilizar medidas relativas do que as absolutas em seus projetos.

## EM

Essa medida você provavelmente verá e usará bastante no CSS para o tamanho de fontes, ela é uma medida relativa pois tem ligação direta com o elemento "pai" que a envolve, vamos entender direito como isso funciona:

```
<div id="div_pai">
  div pai
  <div id="div_filho">
    div filho
  </div>
</div>
```

Dado o HTML acima, observe o CSS abaixo

```
#div_pai {
  font-size: 16px;
}

#div_filho {
  font-size: 3em;
  /* Nesse caso 3em seria como 16px x 3 -> 48px */
}
```

Veja que o elemento pai possui 16px como tamanho de fonte, no elemento filho usamos a fonte como 3em, isso quer dizer que essa fonte terá 3 vezes o tamanho da fonte do elemento pai, assim seria também se usássemos 0.5em, essa fonte teria metade da fonte do elemento pai, nesse caso, seria de 8px.

## REM

Essa unidade assim como EM é bastante interessante e bastante utilizada, ela pode facilitar muito a sua vida no dia a dia com o CSS, essa medida tem relação com o elemento raiz do HTML, ou seja, o próprio elemento <html>, e sempre que você usar outro elemento com a unidade REM esse elemento terá como referência o elemento <html>, por exemplo se você definir o elemento <html> com 10px e em seguida usar 2rem em um elemento, esse elemento terá 2x o tamanho do elemento raiz, ou seja, nesse caso 20px.

```
<div>
  <p>Esse texto não terá o tamanho da fonte alterada</p>

  <p id="txt2x">Esse texto terá o tamanho de 2 vezes o tamanho do elemento raiz</p>

  <p id="txt3x">Esse texto terá o tamanho de 3 vezes o tamanho do elemento raiz</p>

  <p id="txt05x">Esse texto terá metade do tamanho do elemento raiz</p>
</div>
```

Agora no CSS observe o exemplo que faz o uso de tamanhos de fontes com o REM

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

#txt2x {
  font-size: 2rem;
}

#txt3x {
  font-size: 3rem;
}

#txt05x {
  font-size: 0.5rem;
}
```

Nesse exemplo acima o que irá acontecer é que o primeiro elemento <p> terá um tamanho de fonte igual a 10px pois não mexemos nele, e o elemento raiz possui 10px como fizemos, por padrão seria 16px caso não determinássemos tamanho algum, mas como alteramos ficou em 10px. O segundo parágrafo com o id #txt2x terá o equivalente a 20px, pois 2rem é 2x o tamanho do elementos raiz que no nosso caso é 10px, assim como o parágrafo de id #txt3x terá o equivalente a 30px pois está definido como 3rem, e o último parágrafo com o id #txt05x terá 5px de tamanho, pois usamos 0.5rem, isso seria como dividir 10px por 2 (ou multiplicar por 0.5).

Podemos usar do REM para fazer um pequeno truque do qual pode facilitar muito nossa vida ao trabalhar com tamanhos de fontes do CSS, observe o seguinte, como foi dito, os navegadores empregam por padrão o tamanho de 16 pixels, com exceção

dos elementos que possuem tamanhos maiores como o <h1> por exemplo, perfeito, então podemos alterar o tamanho do elemento raiz <html> de forma com que possamos ter um padrão onde 1rem = 10px, 1.2rem = 12px, assim fica mais fácil de trabalharmos, veja como é fácil:

Se **10px equivale a 0.625rem**, podemos fazer o seguinte

```
html {
  font-size: 62.5%;
  /* ou: */
  font-size: 10px;
  /* ou: */
  font-size: 0.625rem; /* 10px => Equivale a 10/16 = 0.625 */
}
```

E a partir disso podemos trabalhar da seguinte forma

```
p {
  font-size: 1.4rem; /* que equivale a 14px */
}

h2 {
  font-size: 3.6rem; /* que equivale a 36px */
}
```

Bem fácil né? Esse é um dos truques que você irá aprender por aqui, mas mais importante que esse truque é você entender a diferença entre o EM e o REM e sempre que possível utilizá-los em seus projetos.

## Porcentagem (%)

A porcentagem também é largamente utilizada em aplicações para a WEB, a partir do momento que você entende como ela funciona numa página WEB muita coisa começa a fazer mais sentido, e principalmente saber diferenciar porcentagem de vw e vh como veremos a frente, em resumo, um elemento ocupará determinada porcentagem de onde ele se encaixa, veremos exemplos que irão clarear as ideias sobre essa forma de definir larguras e alturas.

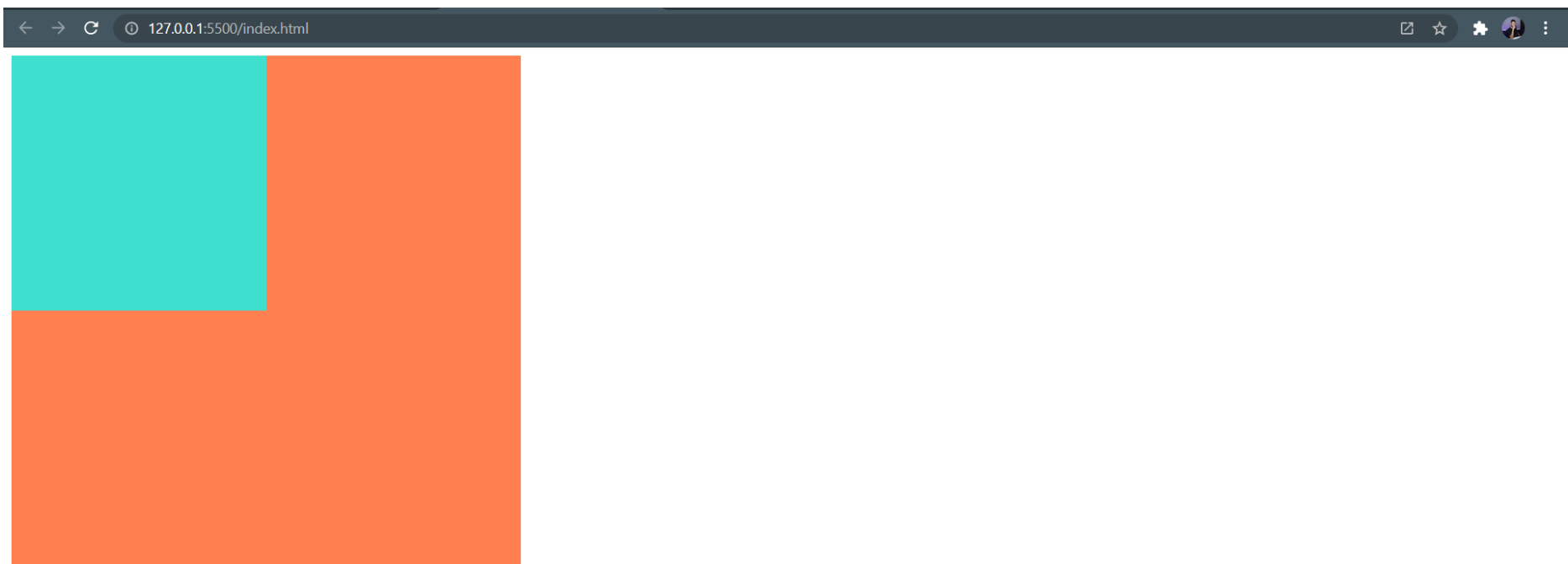
Imagine que temos uma div definida com 500px de largura e 500px de altura, e dentro dessa div temos outra div, essa agora com largura e altura definidas com o valor 50%, o que você acha que irá acontecer? Nesse caso, a div "filha" ocupará 50% do largura e altura da div "pai", ou seja, 250px:

```
<style>
  #div1 {
    width: 500px;
    height: 500px;
    background-color: coral;
  }

  #div2 {
    width: 50%;
    height: 50%;
    background-color: turquoise;
  }
</style>

<div id="div1">
  <div id="div2"></div>
</div>
```

O resultado que temos é o seguinte



Então lembre-se sempre que a porcentagem aplicada em um elemento tem relação com onde ela está inserida, ou seja, sempre será relativo ao pai do elemento.

## viewport width (VW) e viewport height (VH)

Essas unidades tem relação com a largura do dispositivo do qual a aplicação está sendo visualizada, por exemplo se for em uma tela de 1000px, e usarmos um elemento com 100vw esse elemento terá uma largura que ocupa a tela toda do dispositivo, ou seja, o elemento terá 1000px, se for em uma tela de 500px, o elemento de 100vw terá 500px, porém se tiver 50vw o elemento terá 250px, você pode pensar que é bem parecido com a porcentagem que vimos anteriormente, e realmente é, mas a diferença é que a porcentagem tem relação com o elemento pai e o vw e o vh são relativos à tela do dispositivo.

Veremos um exemplo da diferença entre usar % e vw ou vh em uma aplicação:

```
<div id="div1">
  <div id="div2"></div>
  <div id="div3"></div>
</div>
```

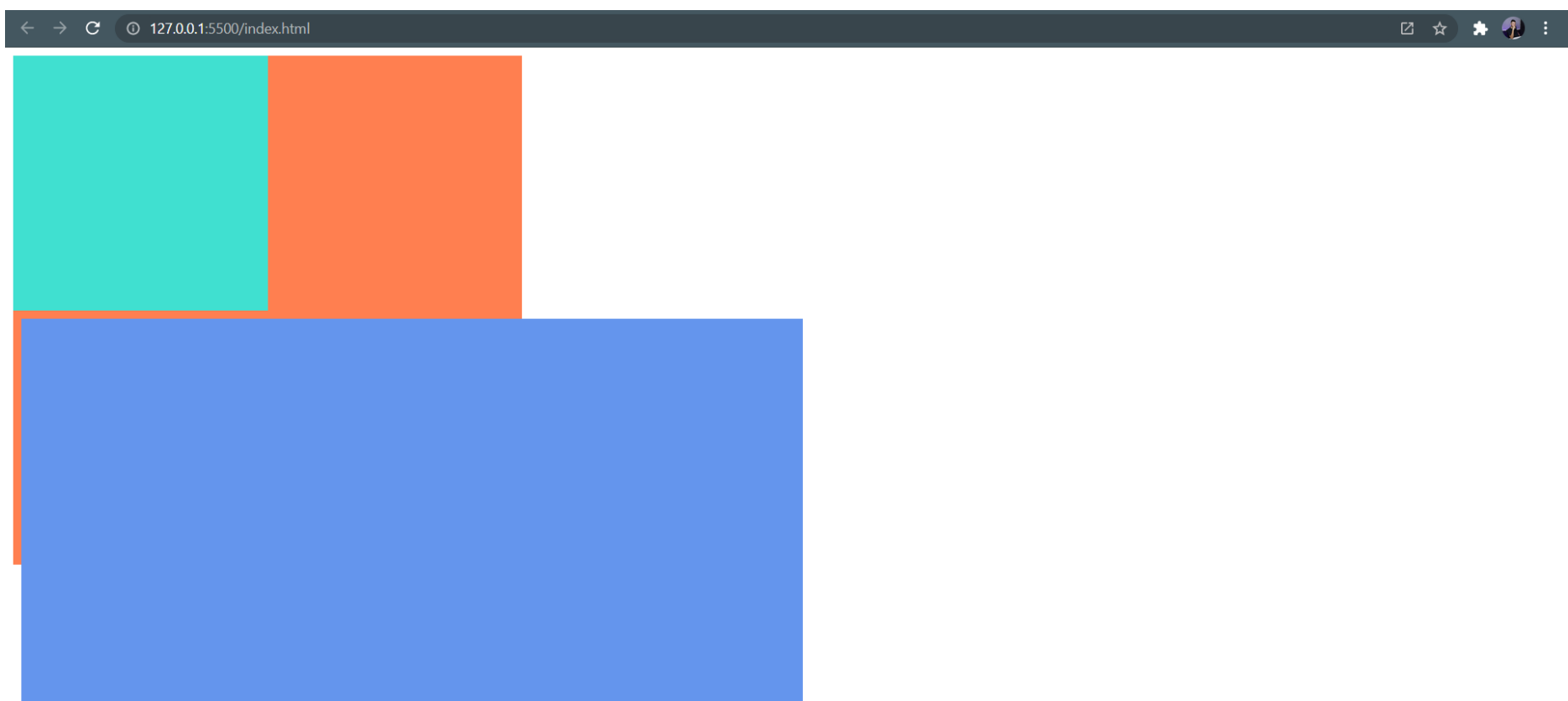
Acima temos um HTML com uma div "pai" e duas "filhas", agora no CSS observe o que vamos fazer e em seguida o resultado final:

```
#div1 {
  width: 500px;
  height: 500px;
  background-color: coral;
}

#div2 {
  width: 50%;
  height: 50%;
  background-color: turquoise;
}

#div3 {
  width: 50vw;
  height: 50vh;
  margin: 8px;
  background-color: cornflowerblue;
}
```

A margin da div 3 é apenas para visualizarmos melhor o que aconteceu com essa div 3 em questão:



Observe bem que é exatamente como foi dito, a div que usa porcentagem se referencia pela div pai, e a div que usa vw e vh se referencia pelo tamanho da tela, **quando usar uma ou outra?** Isso depende 100% do que você pretende fazer na sua aplicação.

Muito bacana, vimos aqui várias formas de dimensionarmos elementos e fontes usando CSS, e quando usar um ou outro sempre depende do que você quer fazer, mas em resumo, prefira usar as medidas relativas no lugar das fixas.