



GERMINARE
Escola de Negócios

Uma década germinando líderes.

 PicPay

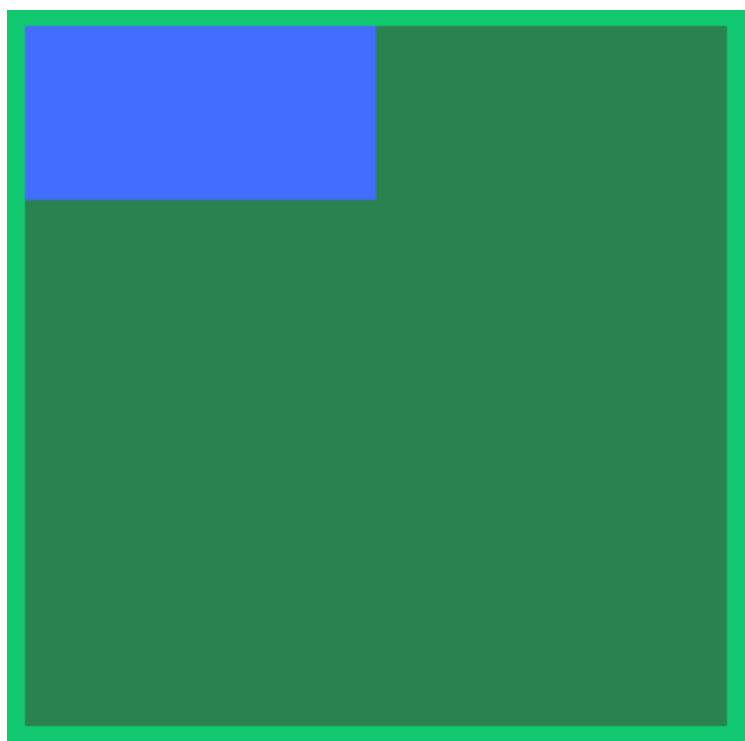


<HTML>CSS: Unidades</HTML>

EQUIPE DE TI – 2022

Porcentagem (%): width

De acordo com a documentação (<https://www.w3.org/TR/css-values-4/#percentages>) indica um valor que se refere a uma fração de um outro valor de referência. Normalmente utilizada para se definir um tamanho relativo a seu pai.



```
#externo {  
  height: 200px;  
  width: 200px;  
  background-color: #2a8250;  
}  
  
#interno {  
  height: 50px;  
  width: 50%;  
  background-color: #426dff;  
}
```

Porcentagem (%): width

Quando o elemento não é envolvido por nenhum container ou algo do tipo, a referência para o seu comprimento em porcentagem será o <body> do documento:

Já no caso abaixo, define-se apenas um quadrado, que terá sua largura de 70%. Nesse caso a sua referência é o <body>, portanto, ocupará 70% de seu width:



```
#externo2 {  
  height: 200px;  
  width: 70%;  
  background-color: #2a8250;  
}
```



GERMINARE

Escola de Negócios

Uma década germinando líderes.

PicPay

Porcentagem (%): font-size

Apesar de não ser o recomendado, é possível usar porcentagem para delimitar o font-size dos elementos:

Para delimitar o tamanho da fonte (font-size), o mesmo acontece. Considera-se o tamanho da fonte de seu pai ao definir a fonte de seu filho com porcentagem (%). A fonte padrão é de 16px, definindo parágrafos < p > com determinada porcentagem, deve-se obter um tamanho de fonte proporcional a ele:

100%

80%

60%

50%

40%

30%

A referência pode mudar caso seja inserida, por exemplo, uma div envolvendo esses elementos textuais que tenha um determinado *font-size* de 30px:

100%

80%

60%

50%

40%

30%

```
<div
  style="
    font-size: 30px;
    background-color: ■ white;
    width: min-content;
    margin: 0 auto;
    border-radius: 10px;
    padding: 2px;
  "
>
```



GERMINARE

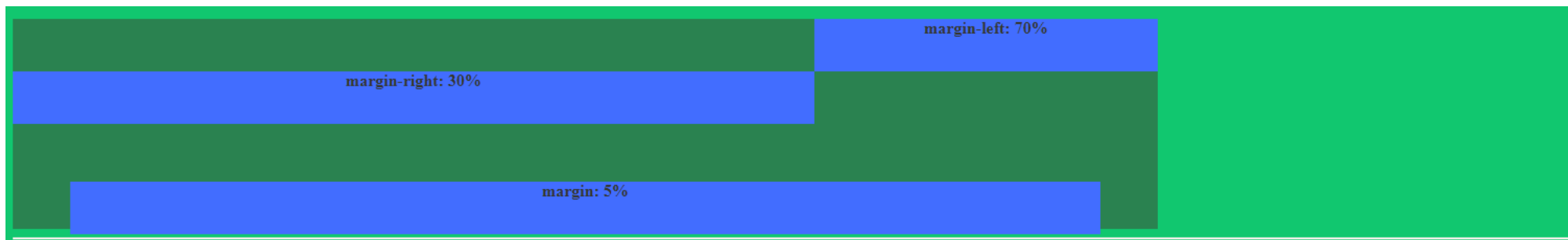
Escola de Negócios

Uma década germinando líderes.

PicPay

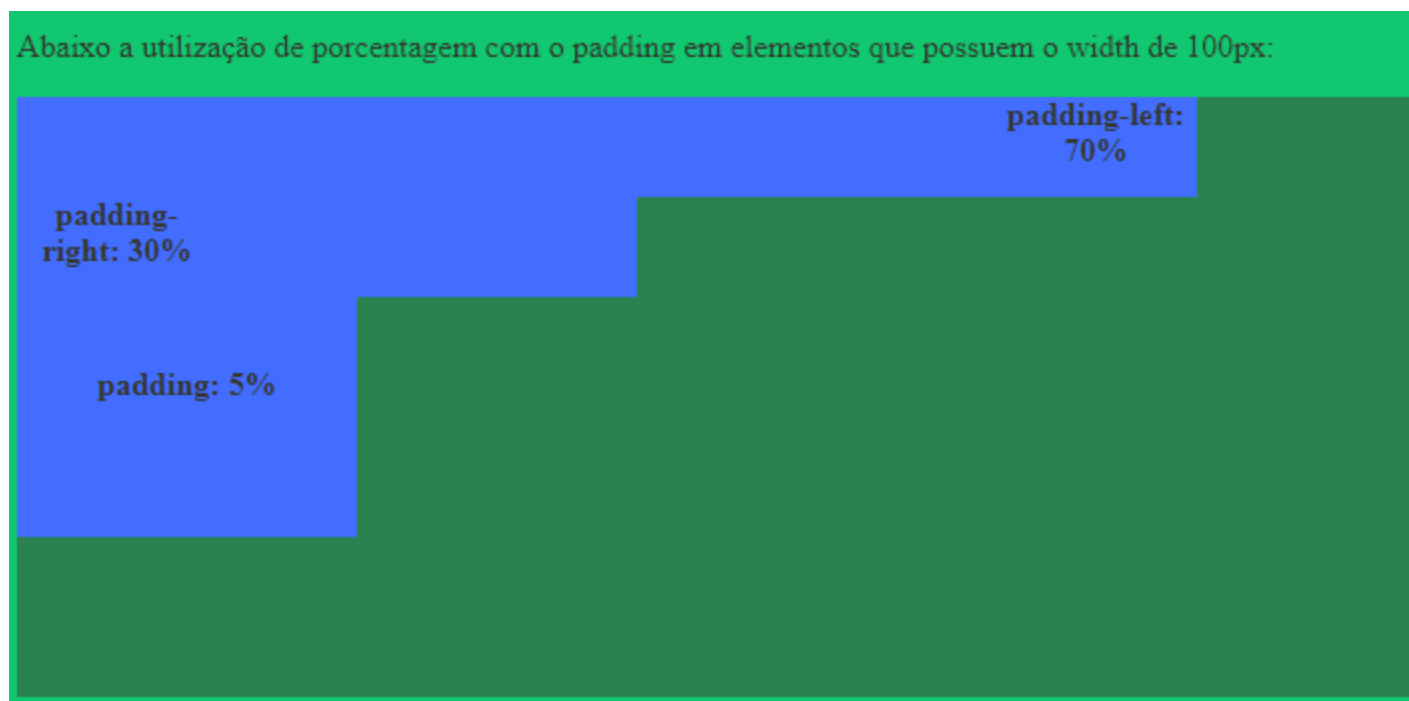
Porcentagem (%): margin

É possível utilizar a porcentagem para a delimitação de margens. Nesses casos é bom que se tenha cuidado, principalmente, na utilização da *shorthand* margin, uma vez que o tamanho da janela do navegador pode ser alterado:



Porcentagem (%): padding

É possível utilizar a porcentagem para a delimitação de paddings. Nesses casos é bom que se tenha cuidado, principalmente, na utilização da *shorthand* padding, uma vez que o tamanho da janela do navegador pode ser alterado:



Unidades Absolutas:

As unidades absolutas são aquelas que não são relativas a nenhuma outra unidade. A documentação MDN apresenta essas unidades e suas equivalências:

| Unidade | Nome | Equivalência |
|---------|---------------------|---------------------|
| cm | Centimeters | 1cm = 37.8px |
| mm | Millimeters | 1mm = 0.1cm |
| Q | Quarter-millimeters | 1Q = 1/40cm |
| in | Inches | 1in = 2.54cm = 96px |
| pc | Picas | 1pc = 1/6in |
| pt | Points | 1pt = 1/72in |
| px | Pixels | 1px = 1/96in |

Unidades Absolutas:

As unidades absolutas são aquelas que não são relativas a nenhuma outra unidade. A documentação MDN apresenta essas unidades e suas equivalências:

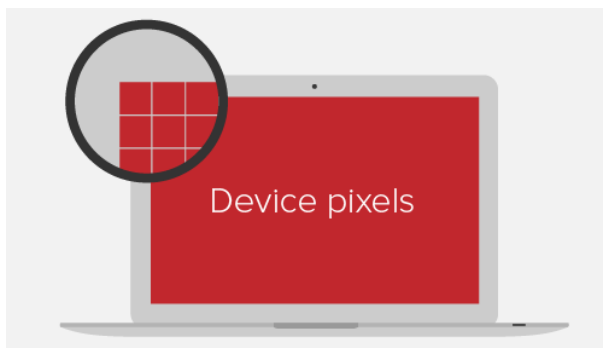
| Unidade | Nome | Equivalência |
|---------|---------------------|---------------------|
| cm | Centimeters | 1cm = 37.8px |
| mm | Millimeters | 1mm = 0.1cm |
| Q | Quarter-millimeters | 1Q = 1/40cm |
| in | Inches | 1in = 2.54cm = 96px |
| pc | Picas | 1pc = 1/6in |
| pt | Points | 1pt = 1/72in |
| px | Pixels | 1px = 1/96in |

As unidades *cm*, *mm*, *Q* e *in* (polegadas) são conhecidas por sua utilização no dia a dia, dependendo do sistema de medidas de cada país. Já as medidas *pc* e *pt* são advindas da tipografia, associando-se, usualmente, a características textuais. A documentação da W3C sugere que para o trabalho com mídias impressas deve-se utilizar essas unidades, também chamadas de unidades físicas (<https://www.w3.org/TR/css-values-4/#physical-unit>)

Unidades Absolutas: pixel (px)

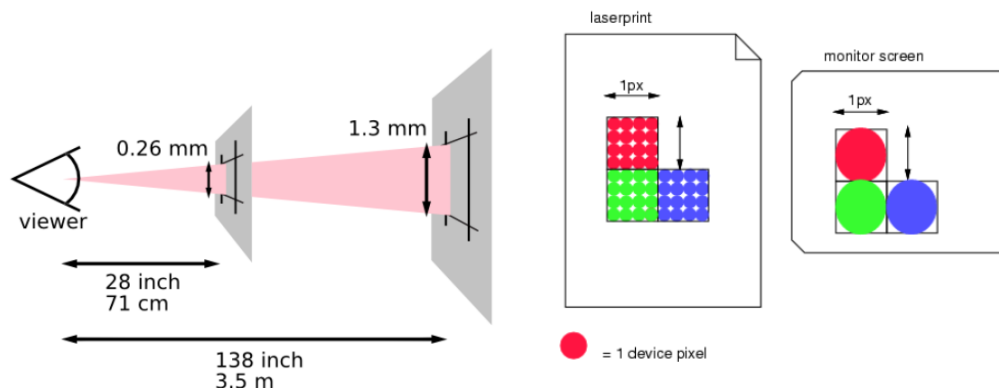
Para o CSS o pixel (px) é compreendido como uma unidade de medida, diferente da ideia de pixel físico. Considere a comparação abaixo:

Pixel Físico (hardware)



Fonte:
<https://www.smashingmagazine.com/2012/08/towards-retina-web/>

Pixel CSS – px (software)



Fonte: <https://www.w3.org/TR/css-values-4/>

Um **pixel físico** (hardware) é basicamente um **ponto de luz na tela** enquanto um **pixel CSS** (software) refere-se a uma **unidade de medida absoluta**. O fabricante de um dispositivo determina quantos pixels físicos equivalem a um pixel de software, conceito chamado de **device pixel ratio (DPR)**. A resolução de um dispositivo se relaciona com a quantidade de pixels que ele utiliza para cobrir uma área, por exemplo de 1px por 1px, mostrado na figura a direita.



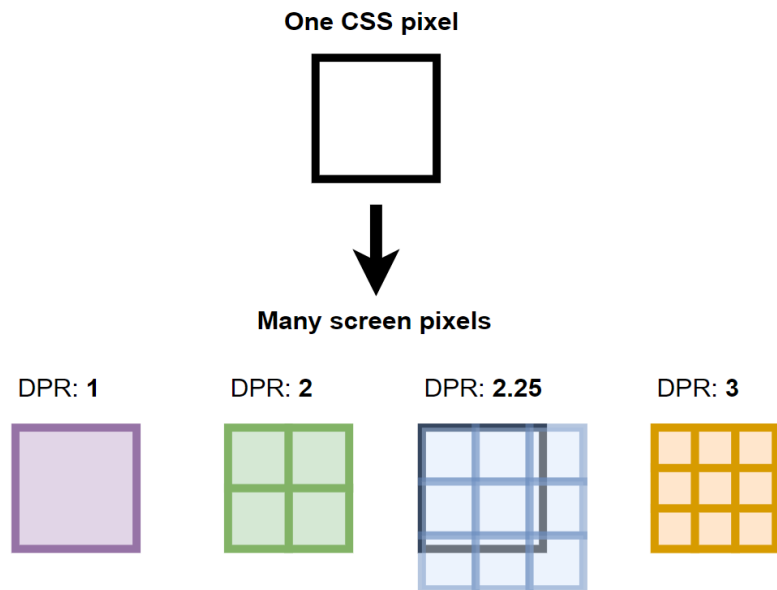
GERMINARE
Escola de Negócios

Uma década germinando líderes.

PicPay

Unidades Absolutas: pixel (px) – DPR

O DPR pode ser compreendido como “número de pixels físicos necessários em uma direção para cobrir um pixel CSS”.



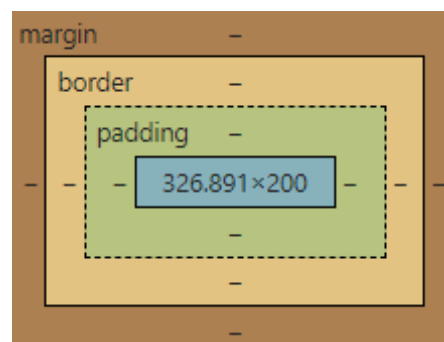
Fonte: <https://tomroth.com.au/dpr/>

Cada dispositivo possui um DPR diferente. O site <https://www.mydevice.io/> permite com que sejam consultadas essa especificação de alguns dispositivos. De acordo com seus registros, um Iphone 3 possui um DPR de 1, enquanto um LG G5 possui um DPR de 4.

Em 2010 a Apple fez história ao lançar o Iphone 4, sendo o primeiro celular com uma tela de retina, possuindo um DPR de 2.

Unidades Absolutas: pixel (px)

Quando inspecionamos as dimensões de um determinado elemento, ou até mesmo sua fonte, os valores apresentados são em px. Isso não quer dizer, porém, que sua estilização foi feita dessa forma:



```
#externo2 {  
  height: 200px;  
  width: 70%;  
  background-color: #2a8250;  
}
```

É importante ressaltar que o valor final apresentado pelo inspecionar é **apenas o valor final calculado**. Nesse caso a largura foi definida como 70% e o resultado apresentado pelo *inspect* foi de 326px

Unidades Absolutas: pixel (px)

Dimensões como height e width podem ser definidas com px, gerando um elemento que não varia suas dimensões de acordo com a mudança da janela do navegador. Considere abaixo um quadrado com 200px de width e height:



```
#bloco {  
  margin: 0 auto;  
  height: 200px;  
  width: 200px;  
  background-color: #2a8250;  
}
```

Independente do tamanho da tela esse quadrado será apresentado com essas dimensões especificadas. Para alguns casos isso pode ser problemático.

Unidades Absolutas: pixel (px)

Considere que se deseja adaptar essa forma geométrica a largura do corpo, delimitando seu width em 70%. Agora, conforme a dimensão da janela do navegador se altera, a largura dessa figura é redefinida:



```
#bloco2 {  
  margin: 0 auto;  
  height: 200px;  
  width: 50%;  
  background-color: #2a8250;  
}
```

Unidades Absolutas: pixel (px)

Considere, no entanto, que serão colocados outros blocos aninhados a essa figura, sendo eles definidos com altura e largura fixas (px). Para alguns cenários onde a tela do navegador não está pequena ou grande demais, a visualização não fica comprometida, mas para outros cenários isso ocorre:



É importante compreender que o redimensionamento da janela do navegador deve gerar alterações visuais no layout, e representar com unidades absolutas, nesse caso px, pode apresentar alguns problemas. Nesse caso o conteúdo aninhado ao bloco verde colapsou a margem direita da figura. O mesmo problema pode ser enfrentado ao dimensionar margin e padding com px.

Unidades Relativas:

Essas unidades especificam um comprimento levando em conta outro comprimento. Normalmente esse tipo de unidade é mais facilmente escalável de um ambiente a outro. Abaixo algumas unidades de acordo com a documentação (https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units):

| Unidade | Relativo a |
|---------|------------------------------------|
| em | Font-size do elemento |
| ex | x-height da fonte do elemento. |
| ch | Largura do 0 na fonte do elemento |
| rem | Tamanho da fonte do elemento raiz. |
| lh | Altura da linha do elemento. |
| vw | 1% do viewport width |
| vh | 1% do viewport height |
| vmin | 1% da menor dimensão do viewport |
| vmax | 1% da maior dimensão do viewport |

Dentre essas unidades destacadas pela tabela ao lado, vamos estudar especificamente: *vw*, *vh*, *rem* e *em*. Para o contato inicial com a estilização em CSS elas são fundamentais. As demais unidades que estão na tabela, e outras mais podem ser consultadas na documentação oficial da W3C (<https://www.w3.org/TR/css-values-4/#relative-lengths>)

Unidades Relativas: vw

A unidade vw refere-se a menor a 1% da largura da janela do navegador. É importante destacar que, a princípio, pode parecer similar a porcentagem (%), porém, deve-se lembrar que a porcentagem baseia-se na característica do pai, diferente da vw. Abaixo um exemplo em que aninhadas a tag <body> implementam-se dois retângulos, um com 50% de width e outro com 50vw de width:



```
#porcentagem {  
  font-size: 24px;  
  font-weight: bold;  
  height: 30px;  
  width: 50%;  
  border: 2px solid #426dff;  
  background-color: #2a8250;  
}
```

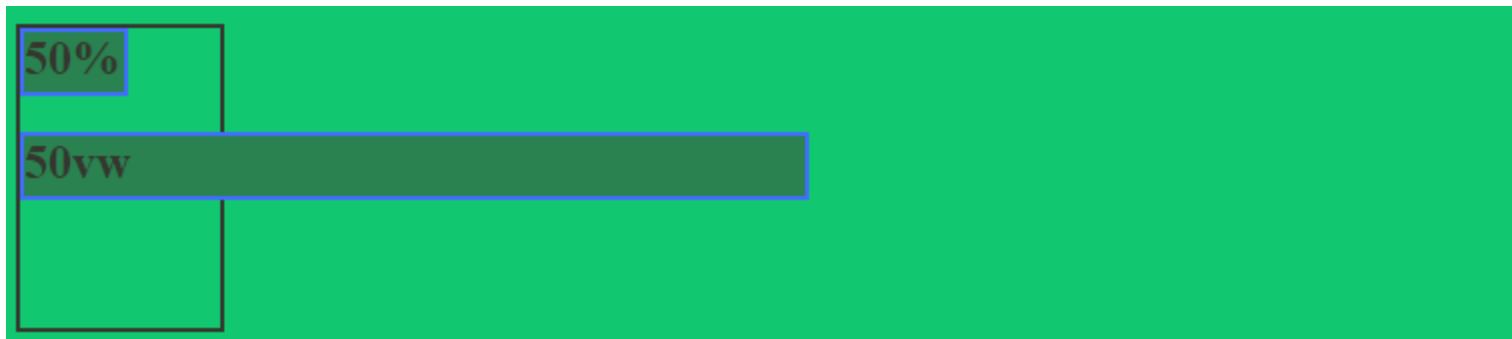
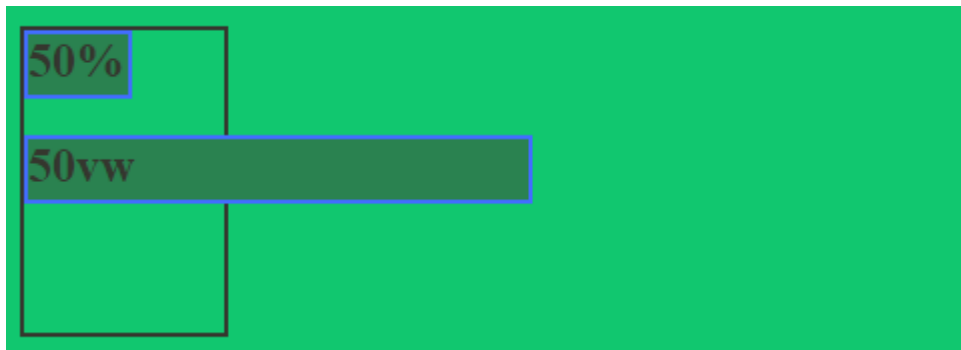
```
#viewport-width {  
  font-size: 24px;  
  font-weight: bold;  
  height: 30px;  
  width: 50vw;  
  border: 2px solid #426dff;  
  background-color: #2a8250;  
}
```



Ambos tem sua largura alterada conforme a janela do navegador é redimensionada, porém a porcentagem considera as dimensões de seu pai, que é visivelmente menor do que o viewport width. Isso se deve as propriedades padrão do <body>

Unidades Relativas: vw

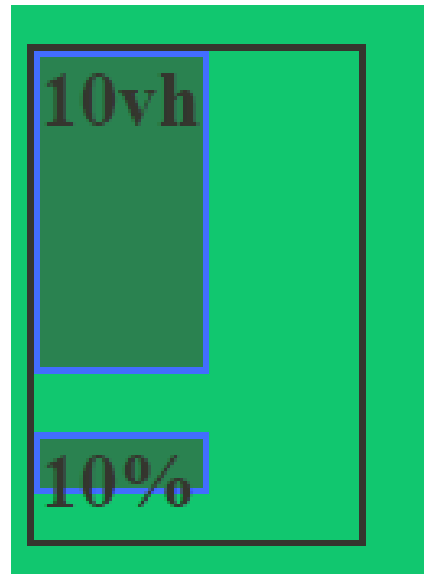
A diferença entre **vw** e porcentagem (%) é mais visível quando os elementos são envolvidos em uma `<div>` com dimensões fixas. Considere a mesma implementação anterior, porém, com os elementos envolvidos pela `<div>`:



A largura do primeiro elemento leva em conta a largura da `<div>`, enquanto o segundo apenas a do **viewport**, colapsando a margem direita.

Unidades Relativas: vh

A mesma ideia se aplica ao **vh** (*viewport height*), sendo que essa unidade considera a altura da janela do navegador:



```
#porcentagem-altura {  
  font-size: 24px;  
  font-weight: bold;  
  height: 10%;  
  width: 50%;  
  border: 2px solid #426dff;  
  background-color: #2a8250;  
}  
  
#viewport-height {  
  font-size: 24px;  
  font-weight: bold;  
  height: 10vh;  
  width: 50%;  
  border: 2px solid #426dff;  
  background-color: #2a8250;  
}
```

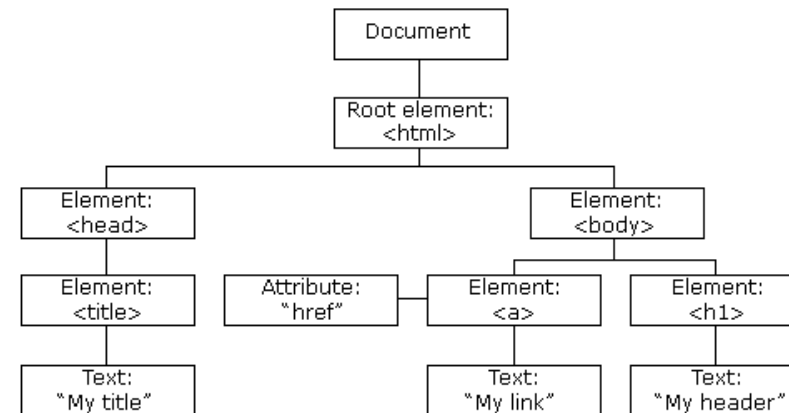
Unidades Relativas: rem e em

Outras unidades de medidas relativas importantes referem-se ao *em* e ao *rem*. A primeira unidade vai ter como referência sempre o seu pai, enquanto a segunda unidade vai ter como referencia o elemento raiz (<html>).

Root Element

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

DOM nodes



Por padrão o elemento raiz <html> apresenta font-size de 16px, portanto a unidade rem será um múltiplo desse número.

Unidades Relativas: rem x em

Outras unidades de medidas relativas importantes referem-se ao *em* e ao *rem*. A primeira unidade vai ter como referência sempre o seu pai, enquanto a segunda unidade vai ter como referencia o elemento raiz (<html>). Considere abaixo alguns parágrafos e suas respectivas *font-size*, sabendo que não estão envelopados em nenhuma tag:



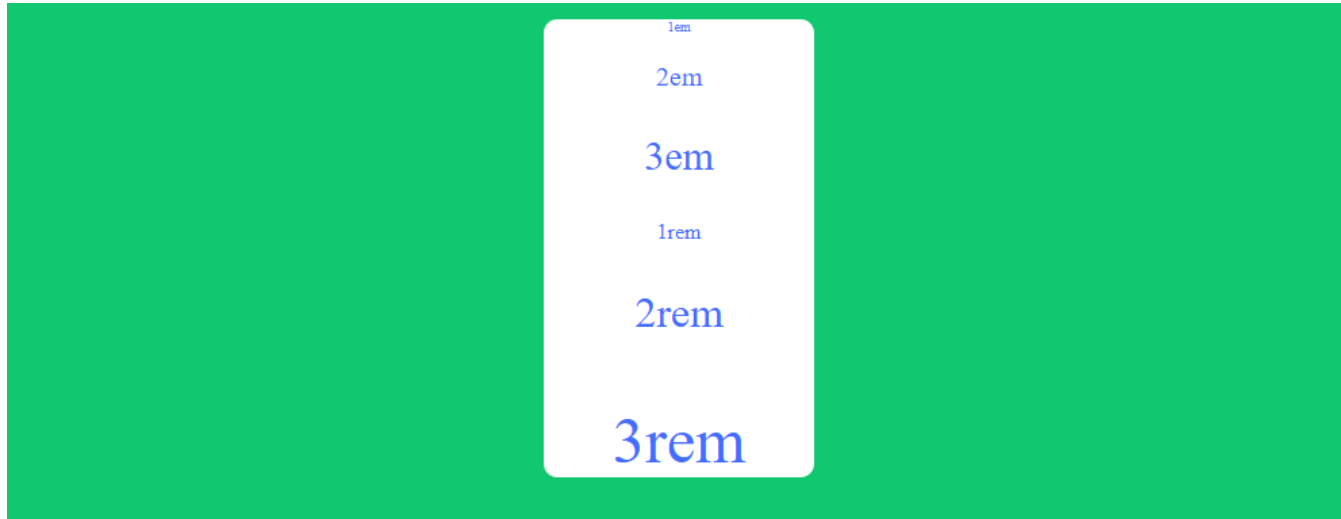
Nesse caso, observa-se que os parágrafos com font-size definido por em e rem possuem o mesmo tamanho de fonte, ou seja:

- 1em = 1rem = 16px;
- 2em = 2rem = 32px;
- 3em = 3rem = 48px;

Isso ocorre pois ambos se pautam na mesma referencia de 16px.

Unidades Relativas: rem x em

Envolvendo os mesmos parágrafos em uma `<div>` em que se delimita um *font-size* de 10px irá alterar a apresentação visual dos elementos que estão definidos com *em*:



Nesse caso, a referência do rem não foi alterada (16px = 1rem), já a referência do em foi alterada (1em = 10px), acarretando na diferença de tamanho das fontes.

Unidades Relativas: rem

Uma maneira mais fácil de se trabalhar com rem é modificar o padrão de 16px para 62,5% que representa 10px. Dessa maneira simplifica-se a representação de unidades de px considerando rem:

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



```
html {  
  font-size: 62.5%;  
}  
  
body {  
  font-size: 1.6rem;  
}
```

= 10px

= 16px

Para representar 24px por exemplo, basta considerar 2.4 rem, imaginando a multiplicação $2.4 \times 10 = 24\text{px}$.

Referências

DUCKETT, Jon. HTML & CSS: design and build websites. Indianapolis, IN: Wiley, 2011.

