tsi tecnologia em sistemas para internet

Curso Superior de Tecnologia em Sistemas para Internet
Desenvolvimento de Sites para Web (DSW)
Rafael Cunha Cardoso
rafaelcardoso@ifsul.edu.br

Unidades de medida CSS

INSTITUTO FEDERAL
Sul-rio-grandense
Câmpus Pelotas

Unidades de medida

Ao longo de nossa vida, utilizamos **diferentes unidades de medida** nas mais diversas situações.

Para medir o tamanho, utilizamos **metros**.

Para declarar distâncias geográficas, utilizamos **quilômetros**.

Para aferir pesos, adotamos **quilos**, e assim por diante.

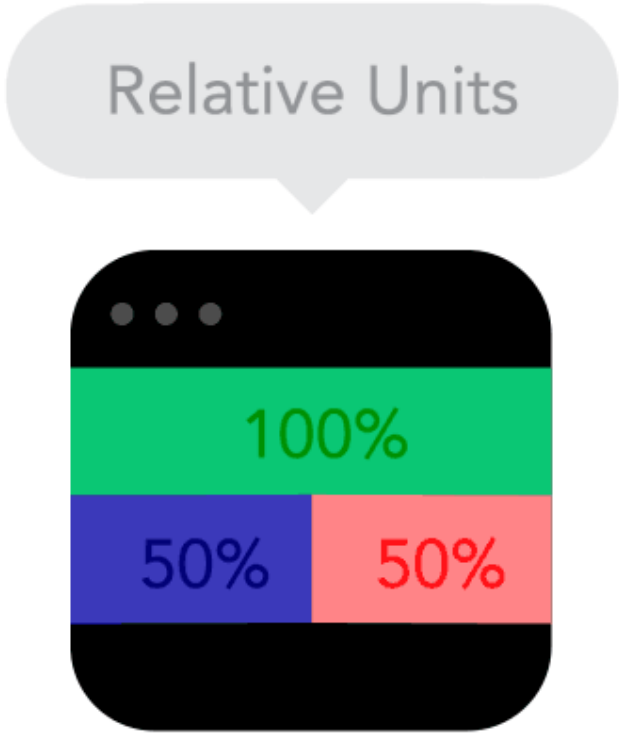
Em programação isto não é diferente.

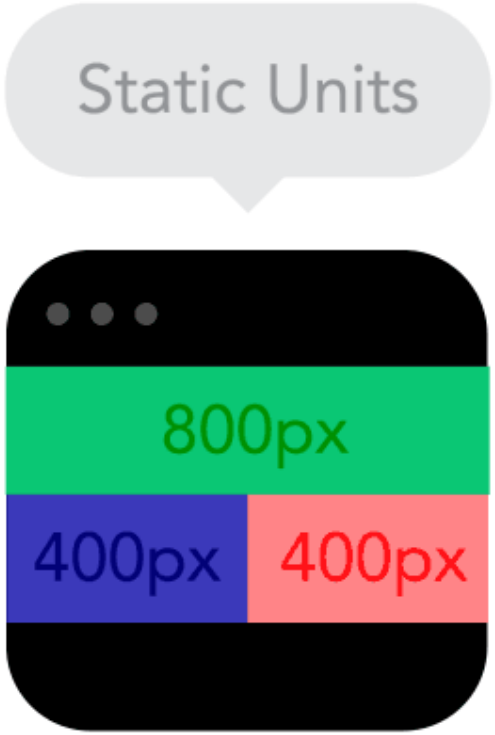
Para posicionar elementos de um website em tela, é possível definir **larguras**, **distâncias** e **alturas** dos **elementos**.

Para configurar essas propriedades adequadamente (de forma que alcancem a estilização desejada), é necessário o uso de **unidades de medida**.

Medidas Relativas x Medidas Absolutas

Uma unidade de medida CSS determina o tamanho de uma propriedade que você está definindo para um elemento ou seu conteúdo.





Unidades de medida **Absolutas**

As unidades **absolutas** são aquelas que se baseiam em medidas físicas reais.

Exemplos: **`px`** (píxeis), **`cm`** (centímetros).

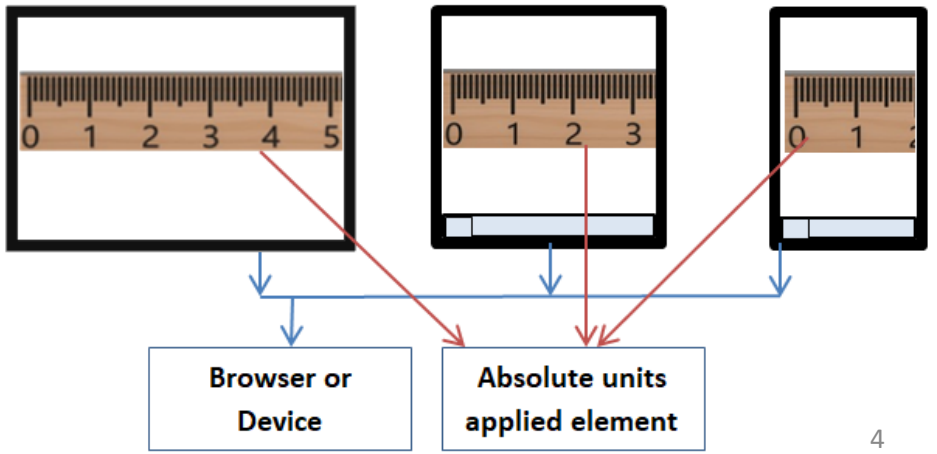
São medidas fixas e **não mudam de acordo com as especificações do dispositivo**.

Não existe nenhum elemento ou referência que influencie no seu tamanho. Ao se declarar um valor em medida absoluta, o **tamanho final da propriedade será o mesmo declarado**.

Indicadas para situações em que se conhece as características físicas e as configurações das mídias onde serão exibidos os projetos.

No desenvolvimento frontend, é necessário **tornar as aplicações responsivas e acessíveis em qualquer tamanho de tela**.

Portanto, é preciso ter cuidado ao utilizar essas medidas para não quebrar o layout do site.



Unidades de medida **Relativas**

Unidades **relativas** são aquelas cujo valor depende de um fator externo.

Exemplos: a **%** que depende do valor do elemento pai; e do **vh** que varia de acordo com a altura do **viewport**. Etc.

Por serem medidas que são calculadas pelo browser (baseando-se em outra unidade), elas tendem a ser flexíveis. Ou seja, permitem resultados diferentes de acordo com o ambiente onde o projeto é executado.

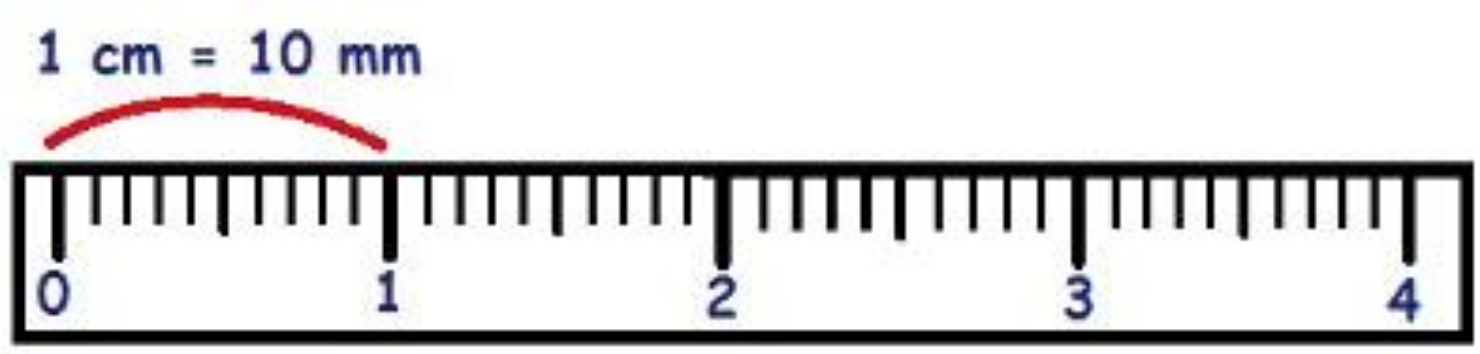
São medidas que têm um comportamento diferente dependendo de algum fator previamente definido e também tem fácil adaptação a diferentes tamanhos de tela.

Unidades de medidas absolutas

cm (centímetros) e mm (milímetros)

Por mais que essas medidas sejam muito conhecidas em países que utilizam o sistema métrico (como o Brasil), elas são muito mais utilizadas no mundo real do que no desenvolvimento front-end.

Por exemplo, o uso de cm é inviável para se definir o tamanho de um texto ou título, pois 1cm pode ser muito grande para a tela de um celular.



in (inches/polegadas)

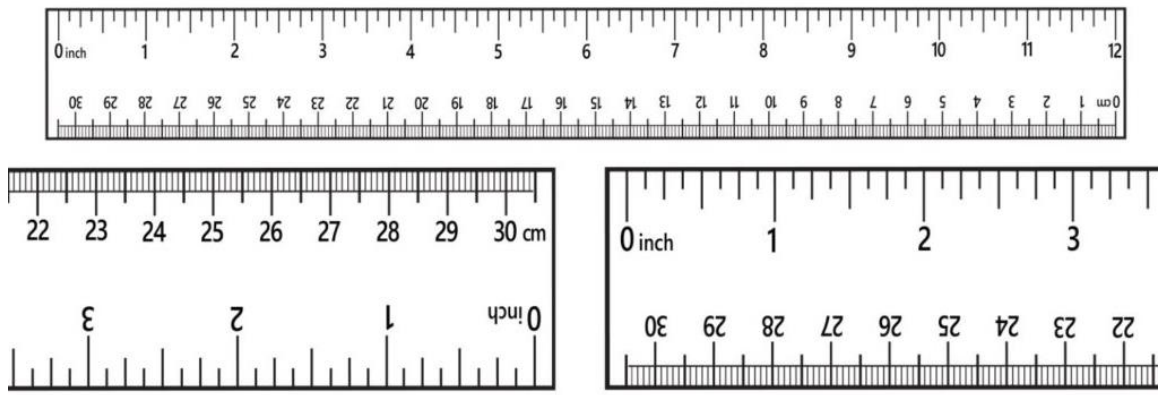
Outra unidade de medida que conhecemos do mundo físico.

Geralmente nos deparamos com elas quando efetuamos compras de dispositivos como televisores ou monitores.

Apesar de existirem, elas não costumam ser utilizadas em projetos, uma vez que não existem um uso prático para elas.

É possível atingir os mesmos resultados utilizando outras unidades.

1 in é equivalente a 2.54 cm



pt (pontos)

Unidade de medida bem antiga, usada para padronização do **tamanho das fontes impressas em papel**.

Unidade é mais conhecida pelos designers, principalmente os que estudam **tipografia**.

Medida pode ser adotada em propriedades relacionadas a fonte de um projeto.

Geralmente espera-se que essa medida seja utilizada em folhas de estilo para **impressões**, quando se precisa ter **certeza do tamanho da fonte utilizada**.

Não é recomendada para a estilização em tela!

`1pt é equivalente a 1.66 pixels`

pc (paica)

Assim como o pt, a **Paica** também é herdada da tipografia.

Outra unidade pouco usada no mundo web.

Todavia é sempre importante conhecer todas as ferramentas disponíveis.

A relação entre as unidades absolutas é:

1pt é equivalente a 1.66 pixels

1pc é equivalente a 12 pts.

px (pixels)

Um pixel é um conjunto de três pontos minúsculos, um vermelho, um verde e um azul (RGB). Diversos pixels combinados compõe as imagens que vemos em uma tela.

É uma medida muito adotada por web designers e grande parte dos desenvolvedores web, que o utilizam como unidade principal de seus projetos.

O píxel é o menor elemento em um dispositivo de exibição!

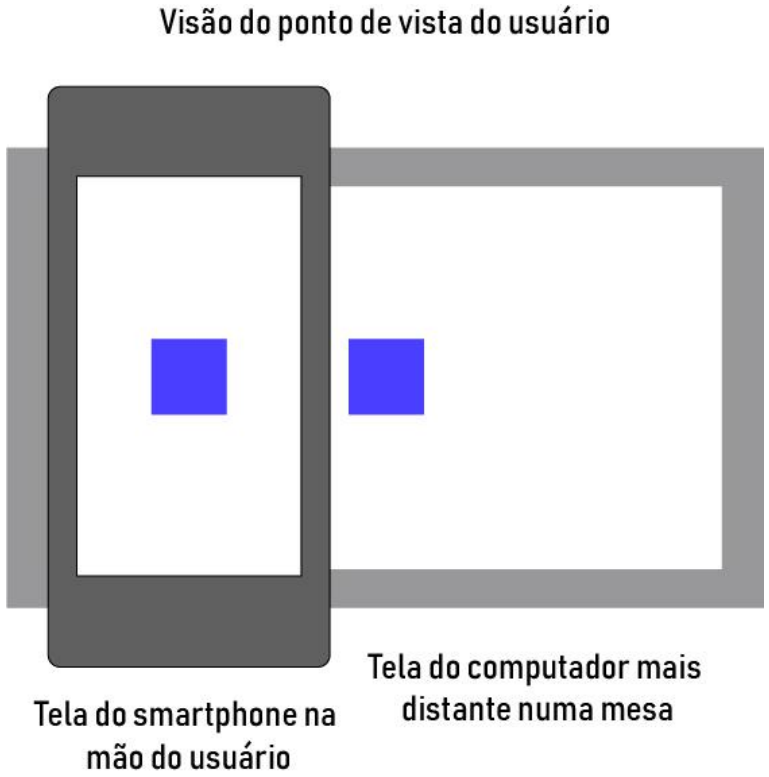
Um detalhe sobre o PX é que na verdade, o pixel do CSS **NÃO é realmente um pixel da tela do dispositivo**, e sim o que chamamos de **pixel de referência**, que geralmente é **maior do que um pixel real**.

Isso torna essa medida ainda mais abstrata.

Sua utilização é bem comum por trazer uma proximidade do tamanho dos elementos entre telas diferentes.

Exemplo

O benefício desse pixel de referência é que ele leva em consideração a proximidade da tela.



Assim, ao utilizar um celular que está próximo a si, o pixel de referência terá o tamanho semelhante ao de um monitor mais distante teria, por exemplo.

Os pixels são considerados unidades absolutas, embora sejam **relativos ao DPI e à resolução do dispositivo de visualização**.

Mas no próprio dispositivo, a unidade PX é fixa e não muda com base em nenhum outro elemento.

O uso de PX pode ser problemático para sites responsivos, mas eles **são úteis para manter o dimensionamento consistente de alguns elementos**.

Se tivermos elementos que não devem ser redimensionados, usar PX é uma boa escolha.

Absolute Unit	Description	Example
px	1/96 of 1 inch (96px = 1 inch)	font-size: 12px;
pt	1/72 of 1 inch (72pt = 1 inch)	font-size: 12pt;
pc	12pt = 1pc	font-size: 1.2pc;
cm	centimeter	font-size: 0.6cm;
mm	millimeter (10 mm = 1 cm)	font-size: 4mm;
in	inches	font-size: 0.2in;


tsi

Unidades de medidas relativas

Unidades de medida relativas

Ao contrário da unidades absolutas, as unidades relativas **são mais adequadas ao design responsivo** e também **ajudam a atender aos padrões de acessibilidade**.

Unidades relativas **escalam melhor em diferentes dispositivos** porque podem **aumentar e diminuir de acordo com o tamanho de outro elemento**.

em

Unidade relativa é bastante famosa no mundo CSS.

- Dificilmente algum navegador que não tem suporte para essa medida, que está presente desde os primórdios.
- Outro ponto é que faz o **em** tão popular a **facilidade de criar layouts fluídos e responsivos**.

Unidade de tamanho relativa ao tamanho da fonte do elemento pai.

Ou seja, esta unidade muda para os elementos filhos de acordo com o tamanho da fonte (`font-size`) do elemento pai.

Dependendo do tamanho da fonte utilizada em determinado elemento, os elementos **filhos serão redimensionados para obedecer a referência a esse tamanho**.

```
<style>
    #div {
        font-size: 16px;
    }

    #filho {
        font-size: 2em;
    }
</style>
```

div pai com um **font-size** de **16px.**

A **div** mais interna, **1em** será igual a **16px**, seguindo a lógica, **2em será igual a 32px**, e assim por diante.

```
<body>

    <div id="pai">
        div pai
        <div id="filho">
            div filho
        </div>
    </div>

</body>
```

Importante: quando essa medida é usada, é preciso considerar o **font-size** de todos os elementos pai.

Exemplo: se existisse uma terceira div mais interna e o tamanho da fonte fosse **2em**... Nesse caso esses **2em seriam 64px**, uma vez que o **font-size** do elemento pai foi definido sendo **32px**(2em)!

Isso tende a se complicar quando temos 5, 6, 7 **divs** aninhadas.

Exemplo

rem (root em)

O **rem** surgiu a partir do **em**.

As suas utilizações são bem semelhantes, porém, ao utilizar o **rem**, dizemos que valor é igual ao tamanho da fonte da **raiz do documento**.

Geralmente, como padrão dos navegadores, esse font-size do root é de 16px.
Logo, **1rem** é igual a **16px**.

Uma vez que o **rem** é relativo ao **elemento raiz** do documento resolve o problema relacionado a elementos aninhados, uma vez que não ocorrerá essa "herança" de tamanhos entre eles.

Permite evitar a realização de cálculos de tamanhos, uma vez que essa técnica permite basear a configuração dos tamanhos utilizando a tag raiz.

```
<style>
        html{
            font-size: 32px;
        }

        #div {
            font-size: 2rem;
        }

        #filho {
            font-size: 2em;
        }
    </style>
</head>

<body>

    <div id="pai">
        div pai
        <div id="filho">
            div filho
        </div>
    </div>

</body>
```

A tag **<html>** é a raiz de todo documento html.

Ao definir um font-size **<html>** para esse elemento de 32px, faz com que **1rem = 32px**, **2rem = 64px** e assim por diante...

Normalmente os browsers especificam o tamanho default da fonte do elemento raiz sendo 16px.

Exemplo

em x rem

Basicamente eles diferem com base na herança.

em é baseado no tamanho da fonte do elemento pai.

Se um elemento pai tiver um tamanho diferente do elemento raiz, o cálculo EM será baseado no elemento pai e não no elemento raiz.

rem é baseado no elemento raiz (HTML).

Todo elemento filho que usa REM usará o tamanho da raiz HTML como seu ponto de cálculo, independentemente de um elemento pai ter ou não tamanhos diferentes especificados.

% (porcentagem)

Apesar de não ser uma unidade de medida, a porcentagem costuma ser bastante utilizada quando trata-se de layout responsivo e fluido.

No desenvolvimento front-end, é corriqueiro utilizar porcentagens para definir valores de propriedades CSS.

Isso ocorre devido a sua flexibilidade em relação ao elemento pai, fazendo com que o elemento filho se adapte melhor a diferentes situações e tamanhos de tela.

Por exemplo: ao definir um módulo com tamanho de **50%**, independente do dispositivo, ele **ocupará a metade do espaço** que lhe cabe (caso esteja dentro de algum outro elemento).

% (porcentagem)

```css
.container {
        width: 800px;
        height: 800px;
        background-color: grey;
    }

    #metade {
        width: 50%;
        height: 100%;
        background-color: blue;
    }
```

```html
<div class="container" id="modulo">
    <div id="metade">

    </div>
</div>
```

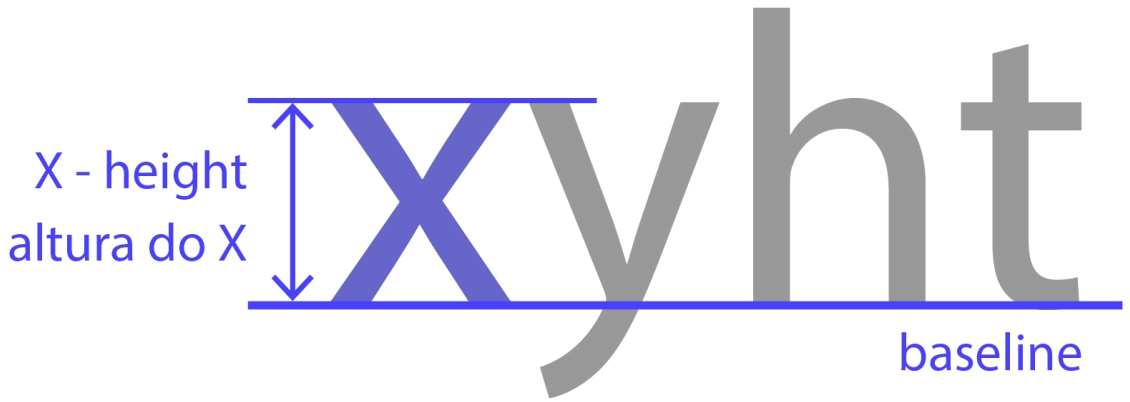
Exemplo

ex

Diferente da maneira que **em** e **rem** funcionam, essa unidade não se relaciona com o tamanho da fonte (**font-size**) de outro elemento.

Ela é relativa a fonte que está sendo utilizada (**font-family**)

Relaciona-se especificamente ao tamanho do caractere x dessa fonte em questão (**x-height**).



Como o navegador sabe esse valor?

- Ele pode vir diretamente com a fonte;
- O navegador pode medir o caractere em caixa baixa
- Se os anteriores não funcionarem, o navegador estipula um valor de **0.5em** para **1ex**.

ch

Na documentação é definida como a "medida avançada" da largura do caractere zero ("0").

Exemplo: A ideia é que um elemento com **`100ch`** de largura comportará uma **`string`** de 100 caracteres dessa determinada fonte, caso essa fonte seja **`monospace`** (todos os caracteres com o mesmo tamanho).

Novamente... A regra de **`1ch = 1 caractere`** se aplica apenas se a fonte usada for **`monospace`**.

Fontes com a largura variável, qualquer caractere pode ser mais largo ou menos largo que o zero ("0")

Courier

Look, 20 characters.
abcdefghijklmnopqrst
12345678901234567890
iiiiiiiiiiiiiiiiiiii
mmmmmmmmmmmmmmmmmmmm

Helvetica

Look, 20 characters.
abcdefghijklmnopqrst
12345678901234567890
iiiiiiiiiiiiiiiiiiii
mmmmmmmmmmmmmmmmmmmm

Georgia

Look, 20 characters.
abcdefghijklmnopqrst
12345678901234567890
iiiiiiiiiiiiiiiiiiii
mmmmmmmmmmmmmmmmmmmm

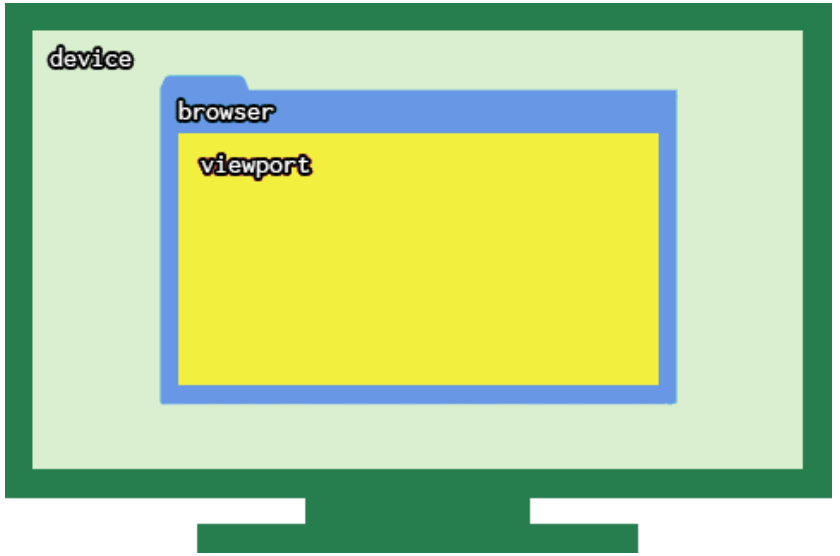
vw (viewport width)

Viewport é a área visível de uma página web para o seu usuário.

A **viewport** pode variar de acordo com o dispositivo, sendo menor em celulares e maior em desktops.

Antes de existir tablets e smartphones capazes de acessar sites, as web pages eram projetadas em função de uma tela de computadores, geralmente com **tamanho fixo e design estático.**

Com advento dos dispositivos móveis, essas páginas eram grandes demais para serem exibidas nesses aparelhos, tornando muito difícil a navegação.



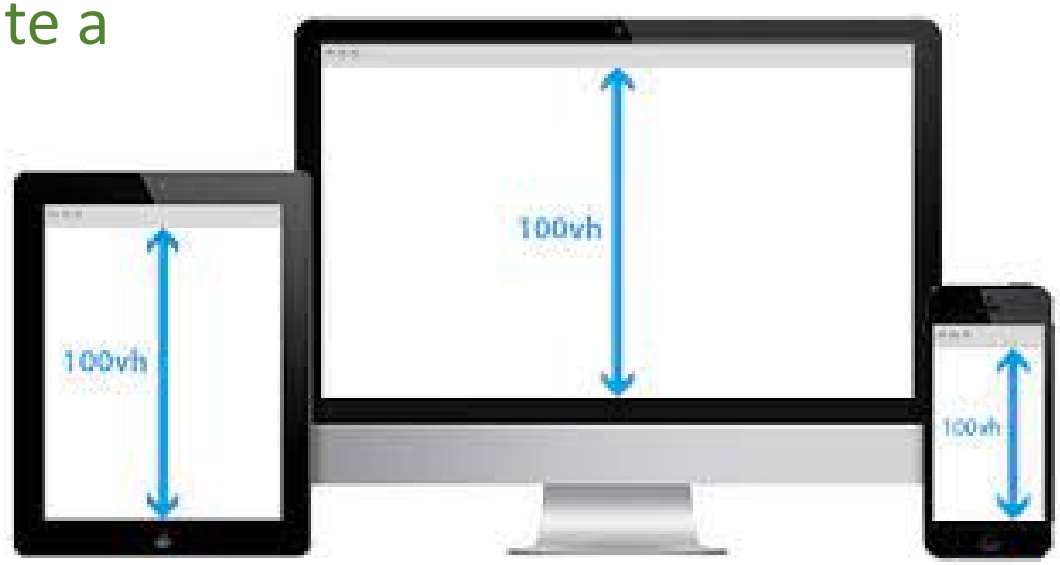
vw (viewport width)

A unidade **vw** se relaciona diretamente com a **largura da `viewport`**:

`1vw` representa **`1%`** do tamanho da largura dessa área visível.

A diferença entre **`vw`** e a **`%`** é bem semelhante a diferença entre **`em`** e **`rem`**:

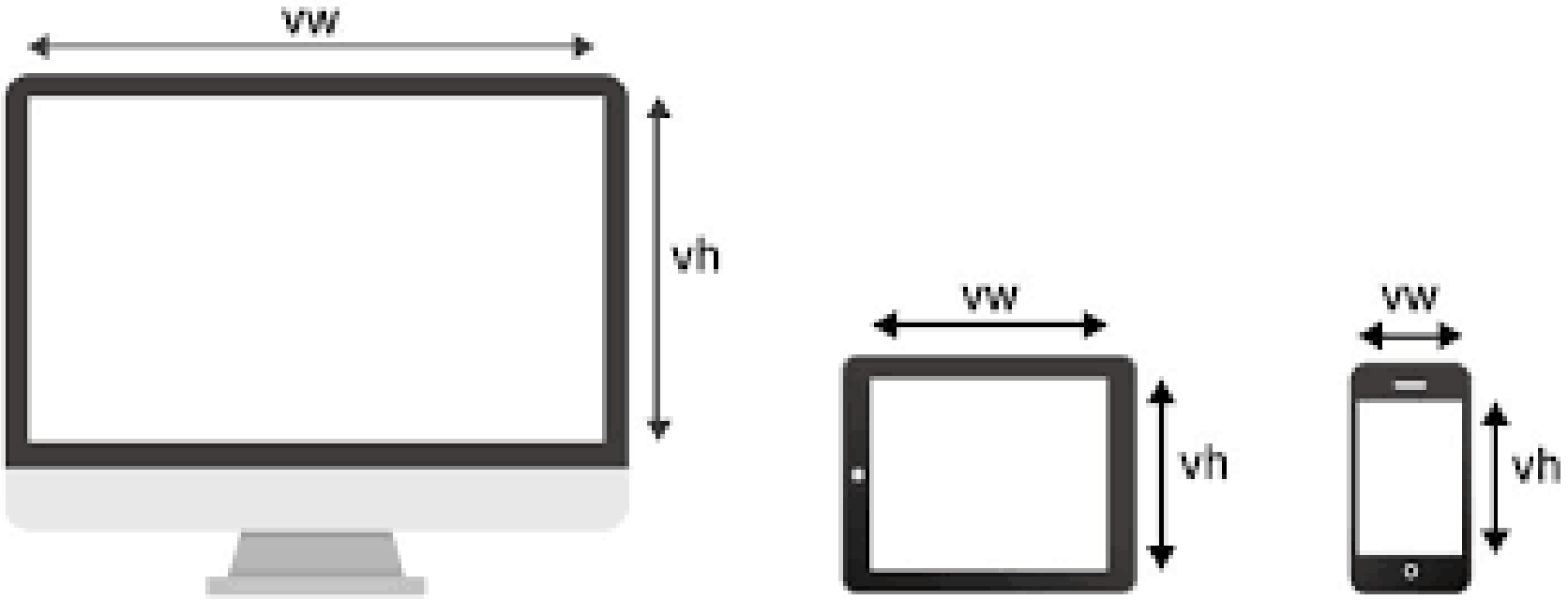
- A **`%`** é relativa ao **contexto local do elemento**
- Enquanto o **`vw`** é relativo ao **tamanho total da largura da `viewport`** do usuário.



vh (viewport height)

A unidade **vh** se relaciona diretamente com a **altura da viewport**:

1vh representa **1%** do tamanho da altura dessa área visível.



vmin (viewport minimum)

Diferentemente das anteriores, o `vmin` utilizará como base a menor dimensão da `viewport` (altura x largura)

Exemplo: se estivéssemos trabalhando com uma **`viewport`** de **`1200px`** de altura e **`700px`** de largura. Neste caso, **`1vmin`** teria o valor de **`7px`** (1% da menor dimensão). Caso tenhamos **`100vmin`**, esse será igual a **`700px`**!

No exemplo anterior, a menor dimensão foi a da **largura**.

Entretanto, se tivéssemos **`200px`** para altura e **`1000px`** para largura, nosso valor de referência seria o **`200px`**... **Ou seja, sempre a menor dimensão!**

vmax (viewport maximum)

Seguindo a lógica da unidade anterior, o **vmax** terá como valor de referência a **maior dimensão da viewport**.

Ou seja, se tivermos **1400px** de altura e **800px** de largura, **1vmax** será equivalente a **14px**.

Um segundo exemplo... Tendo **150px** para altura e **900px** para largura, **1vmax** será equivalente a **9px**.

Sempre será a maior dimensão!

Relative Unit	Description
%	Relative to the parent element's value for that property
em	Relative to the current font-size of the element
rem	Relative to the font-size of the root (e.g. the element). "rem" = "root em"
ch	Number of characters (1 character is equal to the width of the current font's 0/zero)
vh	Relative to the height of the viewport (window or app size). 1vh = 1/100 of the viewport's height
vw	Relative to the width of viewport. 1vw = 1/100 of the viewport's width.
vmin	Relative to viewport's smaller dimension (e.g. for portrait orientation, the width is smaller than the height so it's relative to the width). 1vmin = 1/100 of viewport's smaller dimension.
vmax	Relative to viewport's larger dimension (e.g. height for portrait orientation). 1vmax = 1/100 of viewport's larger dimension.
ex	Relative to height of the current font's lowercase "x".

tsi tecnologia em sistemas para internet

Curso Superior de Tecnologia em Sistemas para Internet
Desenvolvimento de Sites para Web (DSW)
Rafael Cunha Cardoso
rafaelcardoso@ifsul.edu.br

Unidades de medida CSS

INSTITUTO FEDERAL
Sul-rio-grandense
Câmpus Pelotas