Flexbox

Flexbox

display: flex

```
elemento {
   display: flex;
   }
```

Definição: Torna o elemento um flex container automaticamente transformando todos os seus filhos diretos em flex itens.

Flexbox

flex-direction

```
elemento {
    display: flex;
    flex-direction: column;
    }

    [row, row-reverse, column-
    reverse]
```

Definição: Define a direção dos flex itens. Por padrão ele é row (linha), por isso quando o display: flex; é adicionado, os elementos ficam em linha, um do lado do outro.

Flexbox

justify-content

```
elemento {
    display: flex;
    justify-content: flex-start;
    }

    [flex-end, center, space-between, space-
    around]
```

Definição: Alinha os itens flex no container de acordo com a direção. A propriedade só funciona se os itens atuais não ocuparem todo o container. Isso significa que ao definir flex: 1; ou algo similar nos itens, a propriedade não terá mais função.

Flexbox

flex-wrap

```
elemento {
    display: flex;
    flex-wrap: wrap;
    }

    [nowrap, wrap-reverse]
```

Definição: Define se os itens devem quebrar ou não a linha. Por padrão eles não quebram linha, isso faz com que os flex itens sejam compactados além do limite do conteúdo.

Flexbox

align-items

```
elemento {
    display: flex;
    align-items: center;
    }

    [flex-end, flex-start, stretch]
```

Definição: O align-items alinha os flex itens de acordo com o eixo do container. O alinhamento é diferente para quando os itens estão em colunas ou linhas.

Flexbox

align-content

```
elemento {
    display: flex;
    align-content: center;
    }

    [flex-end, flex-start, stretch]
```

Definição: Alinha as linhas do container em relação ao eixo vertical. A propriedade só funciona se existir mais de uma linha de flex-itens. Para isso o flex-wrap precisa ser wrap.

Flexbox

align-self

```
elementoPai {
  display: flex;
  align-items: flex-start;
}
elementoFilho {
align-self: flex-end;
}

[flex-end, flex-start, stretch]
```

Definição: O align-self aceita os mesmos valores que a propriedade align-items mas alinha somente o item selecionado.

Referências

flexbox
https://origamid.com/projetos/flexbox-guia-completo/

http://www.flexboxdefense.com/

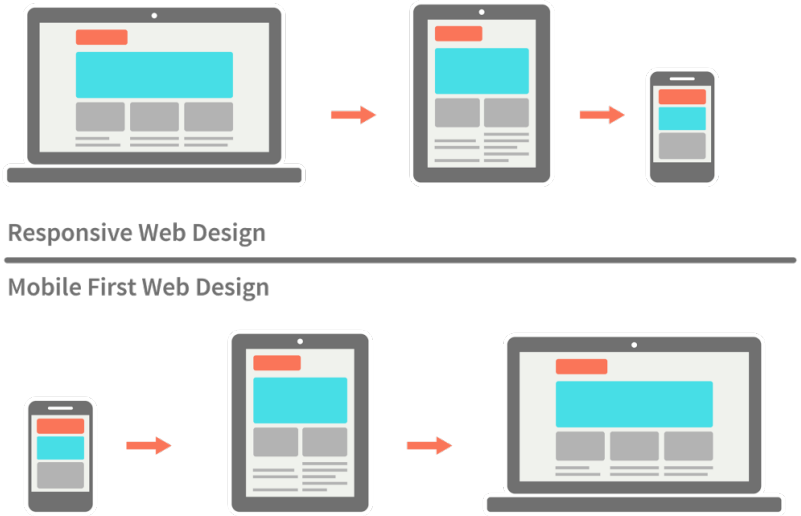
https://flexboxfroggy.com/#pt-br

Unidades relativas e media-queries

O que aprendemos na aula anterior?

Flexbox

Mobile first



Estruturar a página **pensando primeiro** em um **dispositivo móvel**.

Unidades de medida relativas

Medidas relacionadas ao container pai direto.

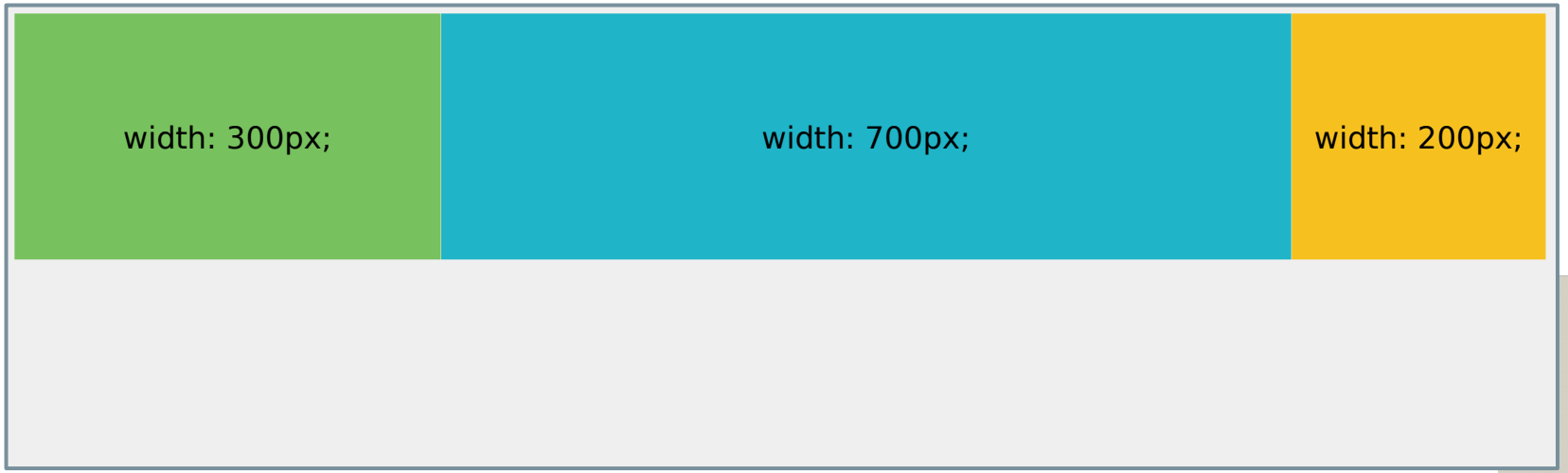
Porcentagens - %

Uma medida expressa em porcentagem **SEMPRE** está relacionada à medida (no mesmo eixo) do **elemento pai** que a contém.

```
.box{
    width: 25%;
}
```

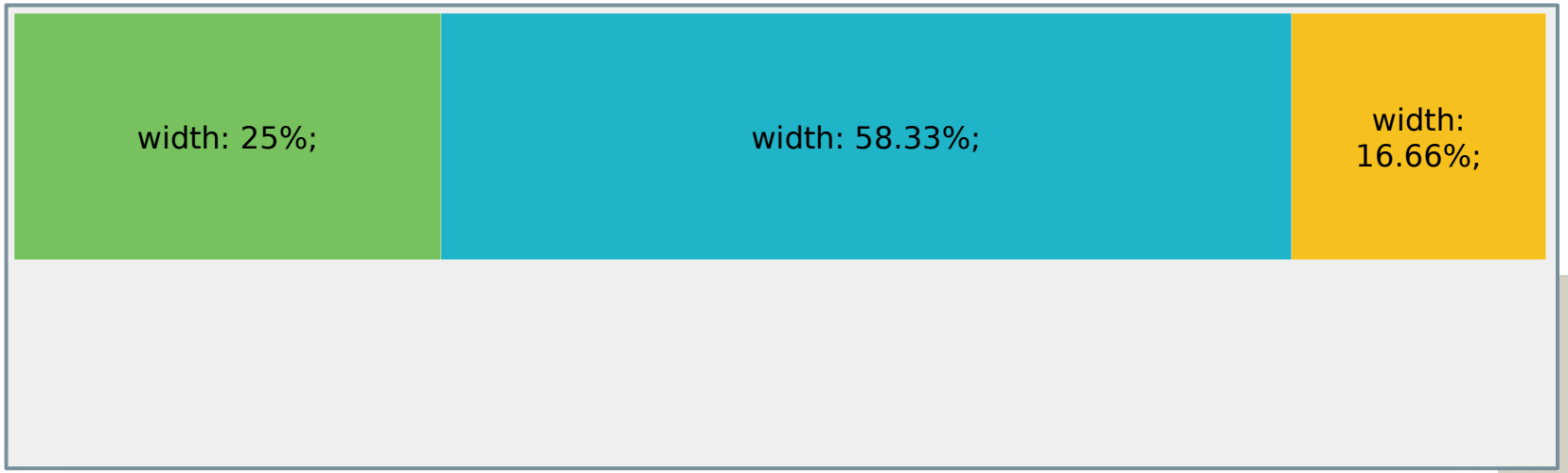
Geralmente, não usamos porcentagens para a altura.

Pixels em percentual



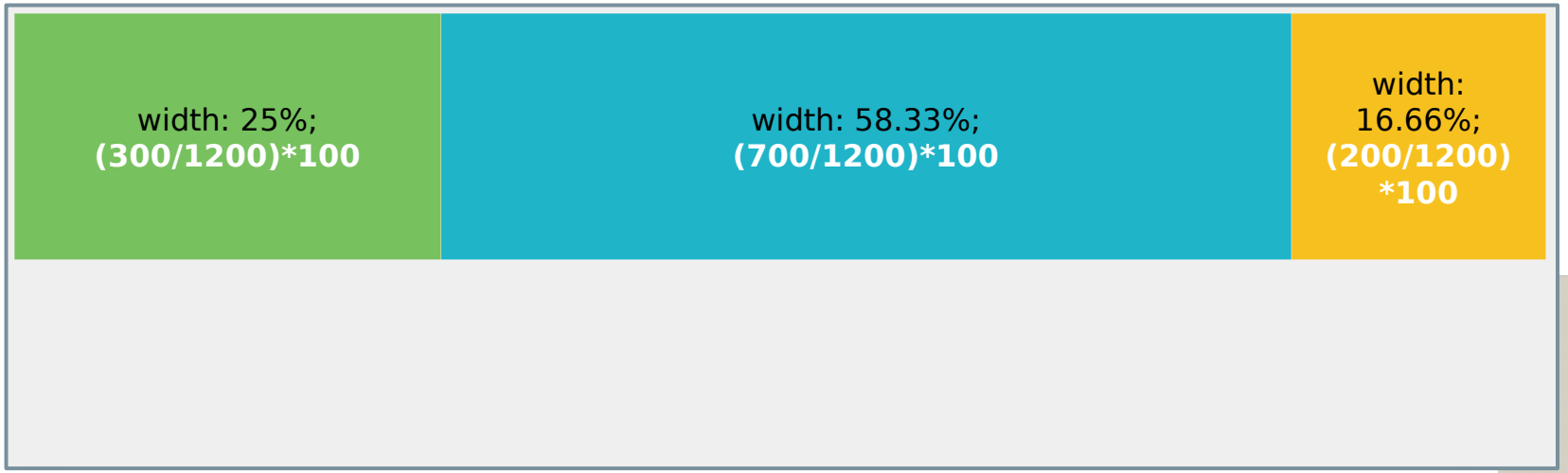
container pai com **1200px** em width

Pixels em percentual



container pai com **1200px** em width

Pixels em percentual



Nesse caso, nossa “base” é **1200px**.

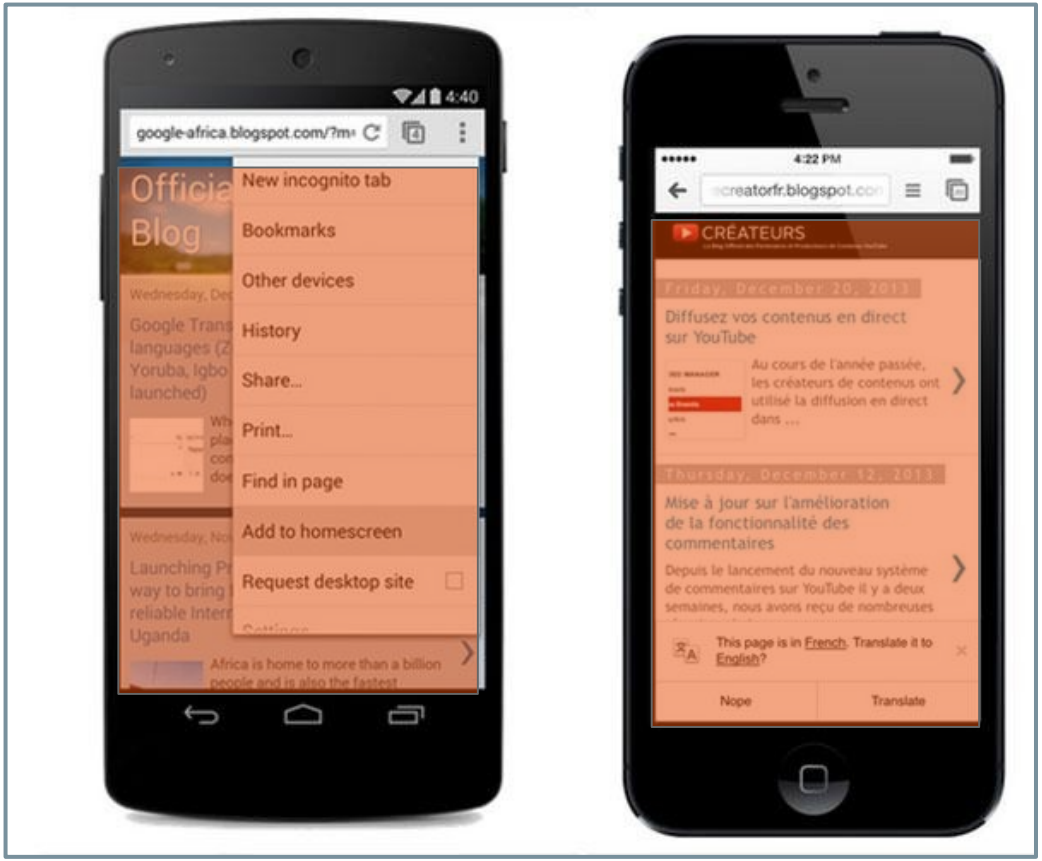
Medida em

Os em’s são **unidades** de medida que **devem ser usadas** para tudo o que tiver relação com **tipografias**.

1em começará sendo igual a 16px (a menos que configuremos outra coisa)

```
p {
  font-size: 1em;
  line-height: 1.5em;
}
```

O Viewport


4:40
google-africa.blogspot.com/?m
New incognito tab
Bookmarks
Other devices
History
Share...
Print...
Find in page
Add to homescreen
Request desktop site
4:22 PM
creatorfr.blogspot.com
CRÉATEURS
Diffusez vos contenus en direct sur YouTube
Au cours de l'année passée, les créateurs de contenus ont utilisé la diffusion en direct dans ...
Mise à jour sur l'amélioration de la fonctionnalité des commentaires
Depuis le lancement du nouveau système de commentaires sur YouTube il y a deux semaines, nous avons reçu de nombreuses
This page is in French. Translate it to English?
Nope
Translate

Tag para RWD

A tag <meta> viewport dá instruções ao navegador sobre **como dimensionar** e escalar a página web durante o carregamento.

```
<meta name="viewport"
content="width=device-width, initial-scale=1">
```

Esta é a estrutura **básica** desta tag, às vezes ela pode ter mais informações.
Saiba mais: Responsive Web Design

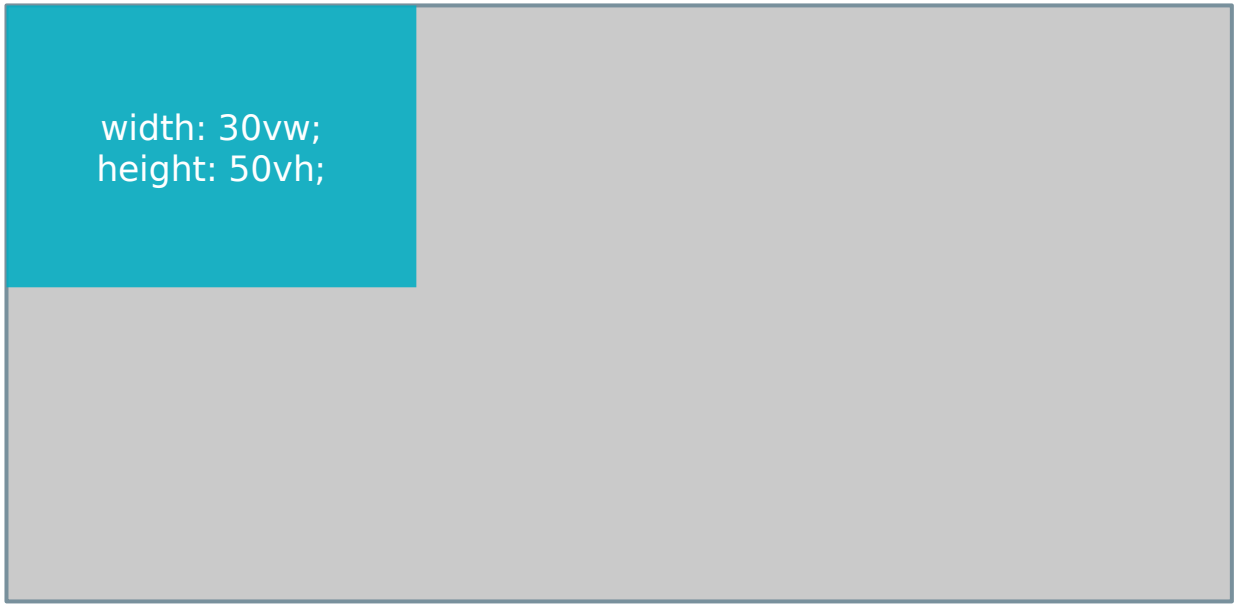
Viewport Measures - vw / vh

Toda medida expressa em **vw/vh** terá **SEMPRE** como eixo de referência o **viewport** do documento.

```
.box {
    width: 25vw;
    height: 50vh;
}
```

Viewport faz referência à “caixa visível” de conteúdo dentro de um navegador.

Neste caso, a caixa terá 0,3 vezes a largura do viewport **(96px)** e 0,5 vezes a altura do viewport **(225px)**



Neste caso, a caixa terá 0,3 vezes a largura do viewport **(198px)** e 0,5 vezes a altura do viewport **(125px)**

Media-queries

Conjunto de regras CSS que permitem reorganizar o conteúdo de acordo com as condições de visualização do documento.

Sempre devem ser escritos no final da folha de CSS.

Media-queries

para Mobile **First**

```
@media (min-width: 460px){
   /* regras de CSS */
}
```

Especificar a **min-width** é como dizer: “se o mínimo é **N**px de largura, aplicar isto” ou “deste ponto para cima”.

Media-queries

para Mobile **Last**

```
@media (max-width: 960px){
    /* regras de CSS */
}
```

Especificar a **max-width** é como dizer: “se o máximo é Npx de largura, aplicar isto” ou “deste ponto para baixo”.

Media-queries

definindo a
orientação

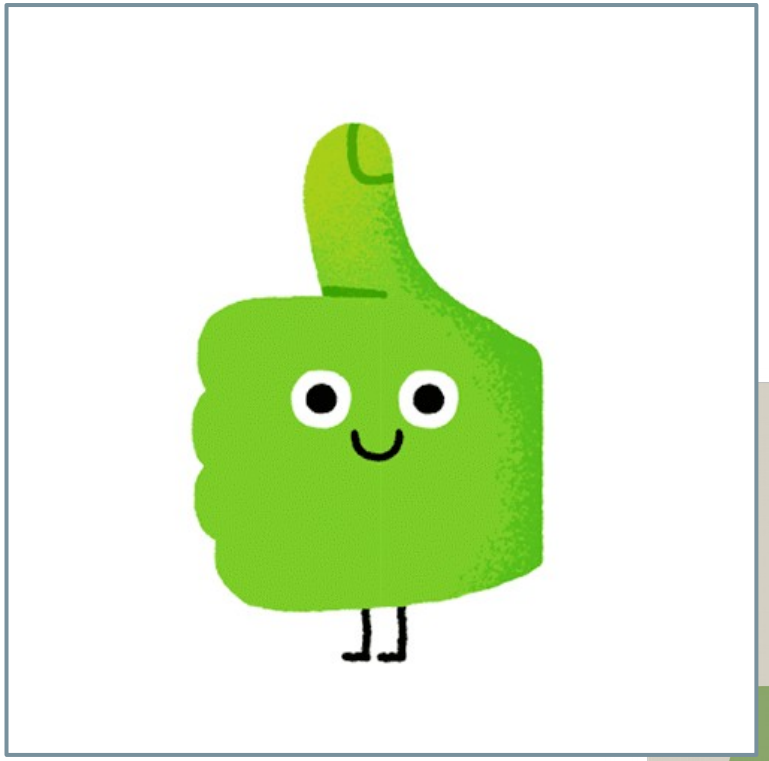
```
@media (max-width: 768px) and (orientation: portrait){
      /* regras de CSS */
}
```

Especificar **a orientação (portrait ou landscape)**, é como dizer
“se a largura máxima é Npx e o dispositivo está em posição vertical/horizontal, aplicar isto”.

Visualização e pseudo-elementos

Barras de navegação

As barras de navegação devem ser feitas de forma semânticas para SEO



Barras de navegação

Semanticamente, as barras de navegação devem ser feitas com listas, de preferência desordenadas, e links:

```
<ul>
   <li><a href=”inicio.html”> Início </a></li>
   <li><a href=”quem.html”> Quem somos </a></li>
   <li><a href=”servicos.html”> Serviços </a></li>
   <li><a href=”lojas.html”> Lojas </a></li>
   <li><a href=”contato.html”> Contato </a></li>
</ul>
```

Barras de navegação

Depois de criar a estrutura, vamos precisar:

- Retirar o estilo padrão de lista
- Posicionar <li> um ao lado do outro
- Fazer com que os <a> se comportem como blocos

Pseudo-elementos

Permitem inserir conteúdo com CSS.
São usados sobre um seletor pré-definido.
Podemos inserir conteúdo de texto, imagens ou elementos de bloco.

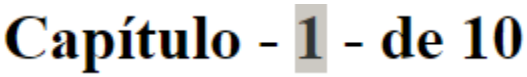
Saiba mais:
https://www.w3schools.com/css/css_pseudo_elements.asp
https://www.w3schools.com/css/css_pseudo_classes.asp
https://www.w3schools.com/cssref/sel_nth-child.asp

Pseudo-elementos

::before & ::after

```
h1::before {
    content: "Capítulo - ";
}

h1::after {
    content: " - de 10";
}
```



Pseudo-elementos

::before & ::after

```
.caixa::before {
   content: url(../images/icone-1.gif);
}

.caixa::after {
   content: url(../images/icone-2.gif);
}
```

Tabelas

Tabelas

```
<table>
    <tr>
        <td>linha 1, célula 1</td>
        <td>linha 1, célula 2</td>
    </tr>
    <tr>
        <td>linha 2, célula 1</td>
        <td>linha 2, célula 2</td>
    </tr>
</table>
```

As tabelas são definidas com a tag <table>.

Tabelas

```
<table>
    <tr>
        <td>linha 1, célula 1</td>
        <td>linha 1, célula 2</td>
    </tr>
    <tr>
        <td>linha 2, célula 1</td>
        <td>linha 2, célula 2</td>
    </tr>
</table>
```

Uma tabela é dividida em linhas com a tag <tr> (table row),

Tabelas

```
<table>
    <tr>
        <td>linha 1, célula 1</td>
        <td>linha 1, célula 2</td>
    </tr>
    <tr>
        <td>linha 2, célula 1</td>
        <td>linha 2, célula 2</td>
    </tr>
</table>
```

Em cada linha os espaços são divididos em células de dados com a tag <td> (table data).

Tabelas

Cabeçalhos

```
<table>
    <tr>
        <th>Cabeçalho coluna 1</th>
        <th>Cabeçalho coluna 2</th>
    </tr>
    <tr>
        <td>linha 2, célula 1</td>
        <td>linha 2, célula 2</td>
    </tr>
</table>
```

Os cabeçalhos em uma tabela são definidos com a tag <th> (table heading).

Tabelas

Estilos

```
table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
}
```

Podemos modificar os estilos da tabela via CSS:

border: Define uma borda
border-collapse: Entra em colapso bordas de célula
border-spacing: Define o espaçamento entre as células

Tabelas

colspan, rowspan

```
<table>
    <tr>
        <td rowspan=”2”>linha 1, célula 1</td>
        <td>linha 1, célula 2</td>
    </tr>
    <tr>
        <td>linha 2, célula 1</td>
        <td>linha 2, célula 2</td>
    </tr>
</table>
```

Definição: atributos dos elementos <td> e <th> para mesclar colunas ou linhas.