



CENTRO DE TECNOLOGIA DA INFORMAÇÃO
DESENVOLVIMENTO WEB BÁSICO

Introdução a layouts com responsividade

Geucimar Brilhador
geucimar@up.edu.br

Atualizado em 02/2021

Tutoriais CSS (Mozilla.org)



► Technologies

► References & Guides

► Feedback

Search MDN



Sign in

Tecnologia Web para desenvolvedores ► CSS



Change language | View in English

Table of contents

Tutoriais

Veja também


Related Topics

CSS

CSS Reference

CSS Tutoriais

CSS (*Cascading Style Sheets* ou **Folhas de Estilo em Cascata**) é uma linguagem de [estilo \(en-US\)](#) usada para descrever a apresentação de um documento escrito em [HTML](#) ou em [XML](#) (incluindo várias linguagens em XML como [SVG](#), [MathML](#) ou [XHTML](#)). O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias.

CSS é uma das principais linguagens da **open web** e é padronizada em navegadores web de acordo com as [especificação da W3C](#) . Desenvolvido em níveis, o CSS1 está atualmente obsoleto, o CSS2.1 é uma recomendação e o [CSS3](#), agora dividido em pequenos módulos, está progredindo para a sua padronização.

<https://developer.mozilla.org/pt-BR/docs/Web/CSS>

Um pouco de história...

- No início da web os layouts disponíveis eram **fluídos** ou de **medidas fixa**. Por um período de tempo utilizou-se o layout baseado em tabelas;
- Em 2004, Cameron Adams escreveu um post intitulado **layout dependente de resolução**, descrevendo um método de criação layout para se adaptar às diferentes resoluções de tela. Esta abordagem usava Javascript para detectar a resolução da tela e carregar o CSS correto;
- Em 2010, Ethan Marcotte cunhou o termo **layout responsivo** que é baseado em três técnicas: grades fluídas, imagens fluídas e media query.
- Em 2011, Mark Otto e Jacó Thornton desenvolveram o **framework Bootstrap** que passou a ser uma das ferramentas mais utilizadas nos últimos anos para criação de layout de aplicações web.

Técnicas de layout baseado em CSS

- **Fluxo normal:** segue a ordem que os elementos são colocados no HTML;
- **Layout com tabelas:** é baseado na propriedade `display table`, mas não é mais recomendado para layout de páginas, deve ser utilizado apenas para formatação de conteúdos tabulares;
- **Elementos flutuantes (float):** foi utilizado para substituir o layout baseado em tabelas, mas não é recomendado para isso. Só deve ser utilizado em casos específicos;
- **Propriedade de posicionamento (position):** permite colocar os elementos em posições específicas da página (viewport). Só deve ser utilizado em casos específicos;
- **Flexbox, Grid e Multi-column:** são os recursos adequados para a formatação de layout de páginas web.

Propriedade flutuante (**float**)

- **Float** altera o comportamento de um elemento e dos elementos de nível de bloco que estejam no fluxo normal. O elemento é movido para a esquerda ou para a direita e removido do fluxo normal, e o conteúdo flutua ao redor do elemento;
- Aceita os valores:

float: left;

float: right;

float: none;

float: inherit;

Quando usar float?



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Técnicas de posicionamento (**position**)

- A propriedade **position** possui os valores static e relative que mantêm o elemento no fluxo normal da página e outros valores absolute, fixed e sticky que movem o elemento para outra camada de visualização no eixo-z retirando-o do fluxo normal da página;
- Aceita os valores:

position: static;

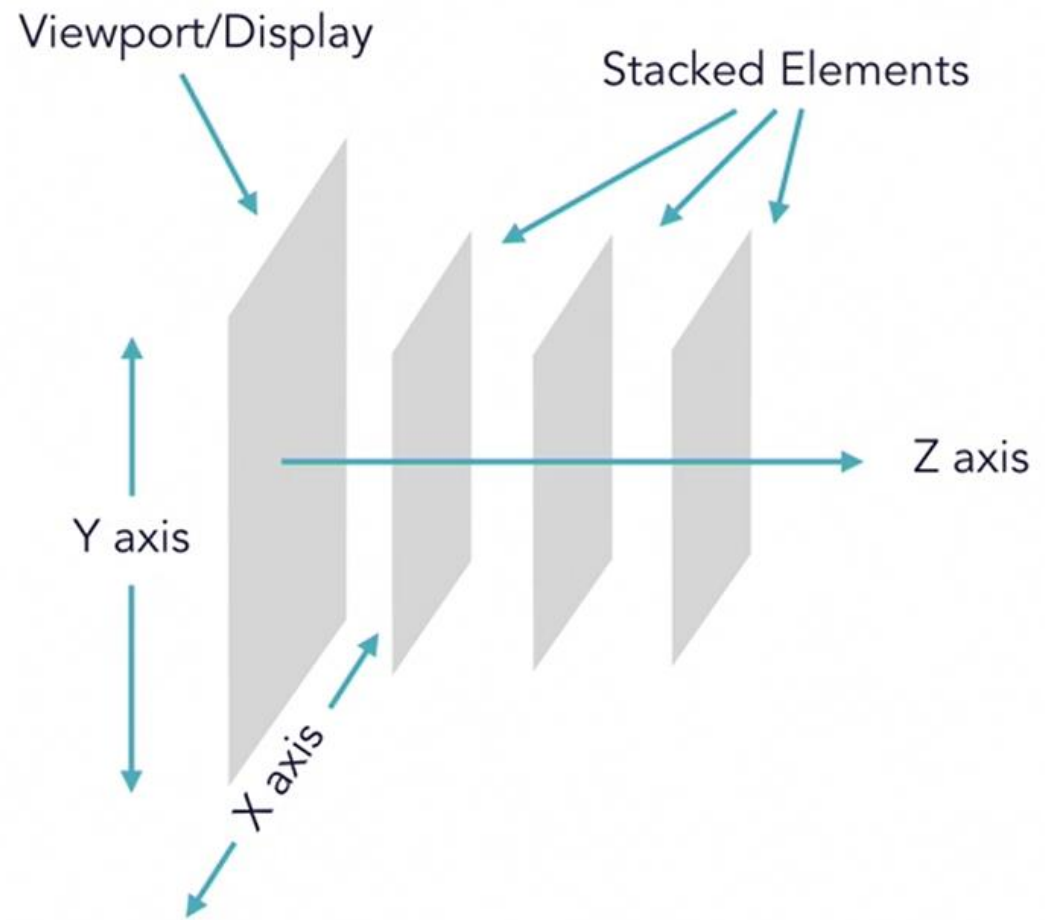
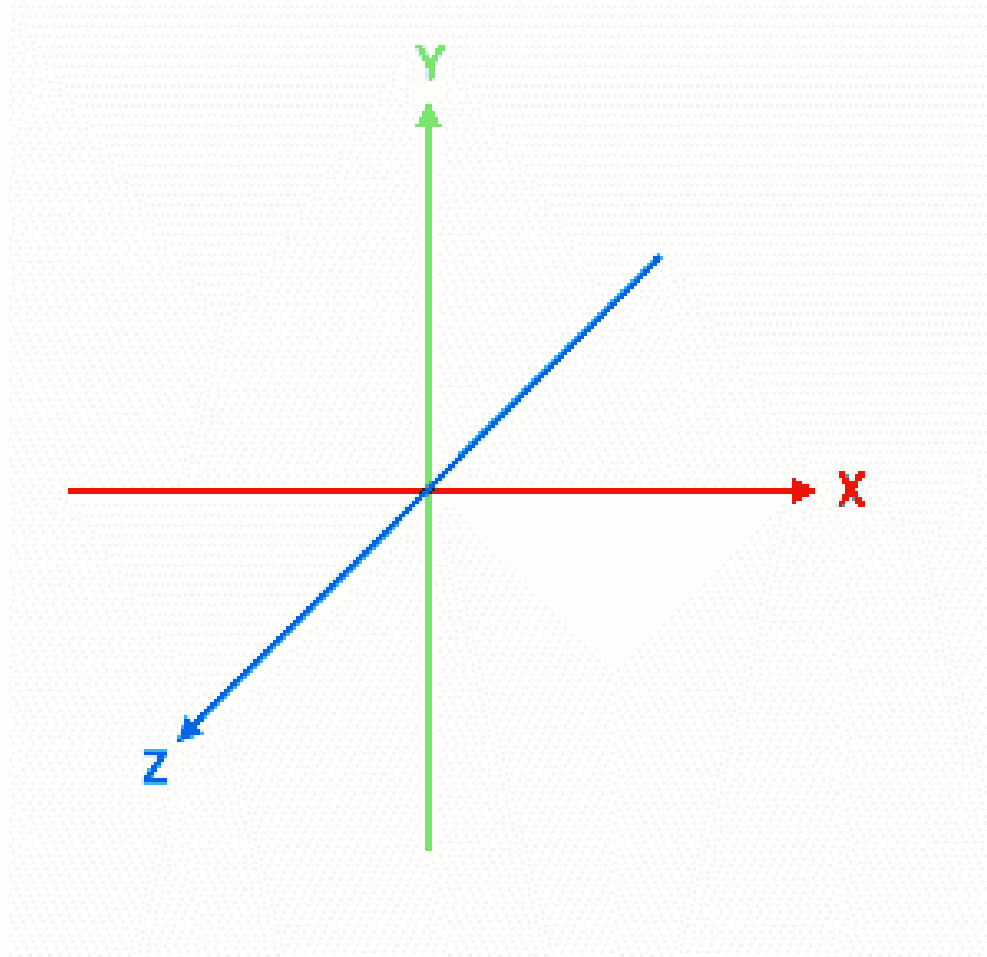
position: relative;

position: absolute;

position: fixed;

position: sticky;

Position + z-index



Técnicas de posicionamento

position: ?

Elemento com posicionamento absoluto (absolute) sai do fluxo.

Elemento com posicionamento estático (static) fica no fluxo normal.

Elemento com posicionamento estático (static) fica no fluxo normal.

Elemento com posicionamento estático (static) fica no fluxo normal.

Elemento com posicionamento relativo (relative) fica no fluxo normal.

Elemento com posicionamento estático (static) fica no fluxo normal.

Elemento com posicionamento fixo (fixed) sai do fluxo.

Elemento com posicionamento estático (static) fica no fluxo normal.

Elemento com posicionamento estático (static) fica no fluxo normal.

Elemento com posicionamento pegajoso (sticky), depende.

Elemento com posicionamento estático (static) fica no fluxo normal.

Alterações no fluxo normal (**display**)

- Muda o comportamento de apresentação de um elemento no fluxo normal:

display: block;

display: inline;

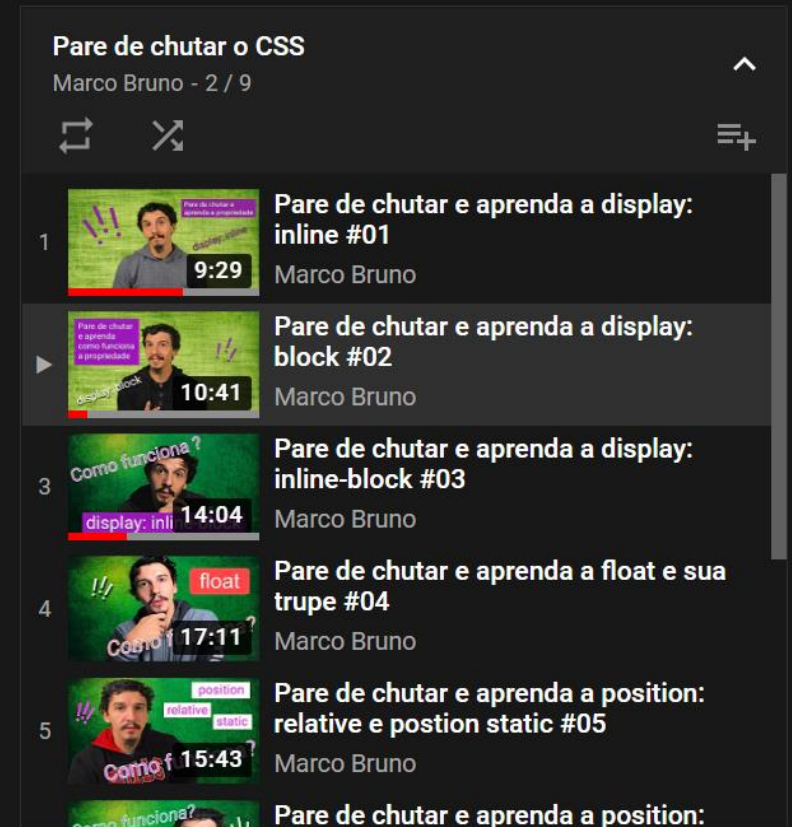
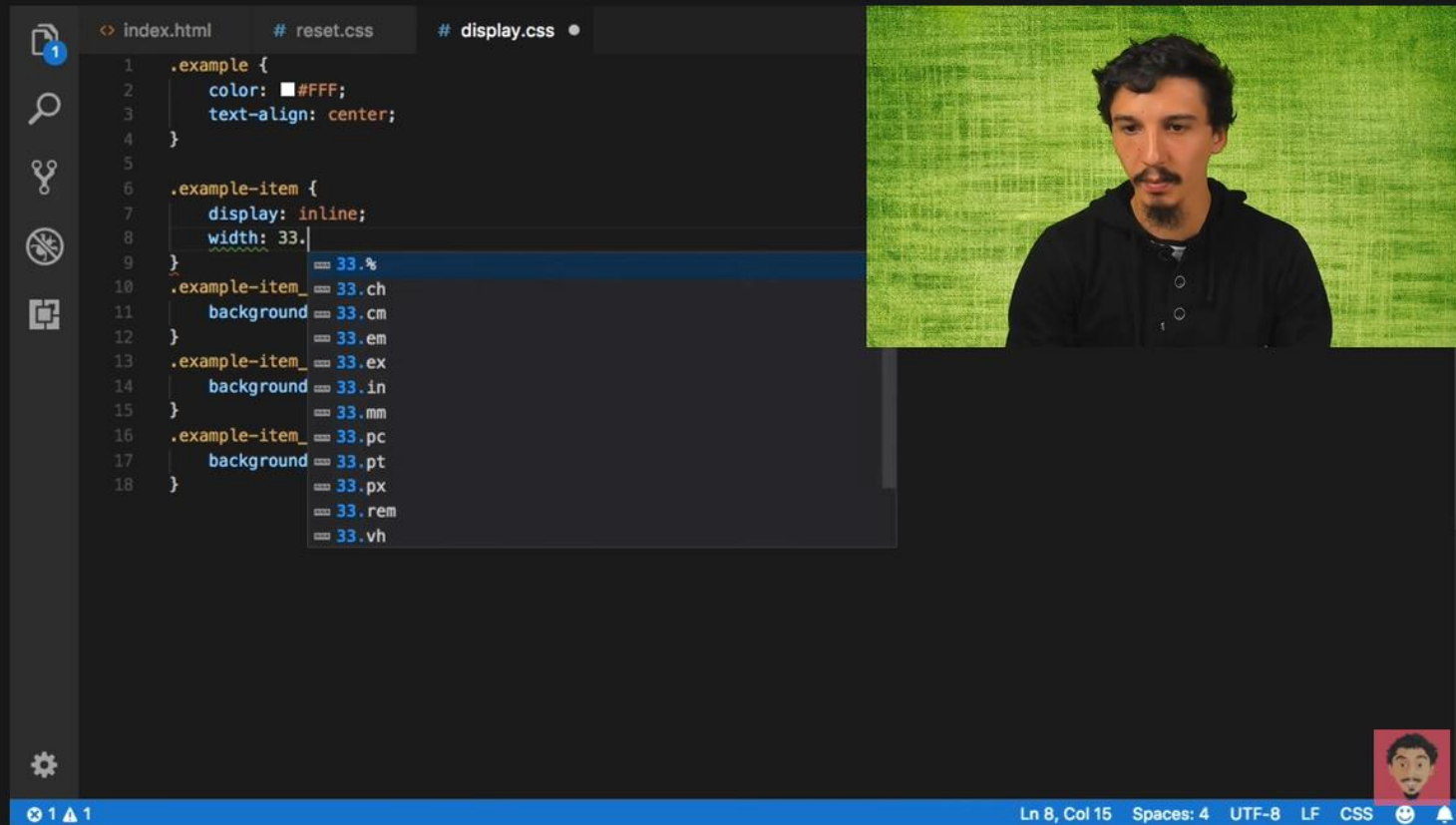
display: inline-block;

- Transforma a tag HTML em um contêiner cujos elementos filhos poderão ser organizados ou na forma de linha, coluna ou grade:

display: flex;

display: grid;

Pare de chutar o CSS

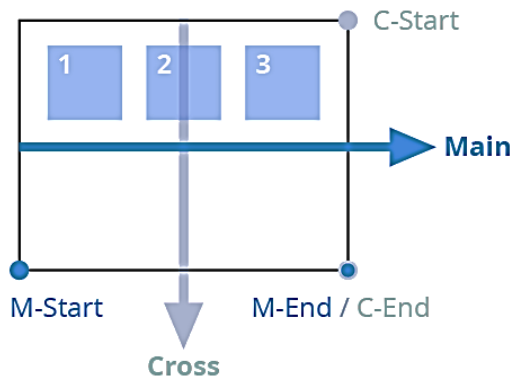


https://youtu.be/HWfhwokS_qg

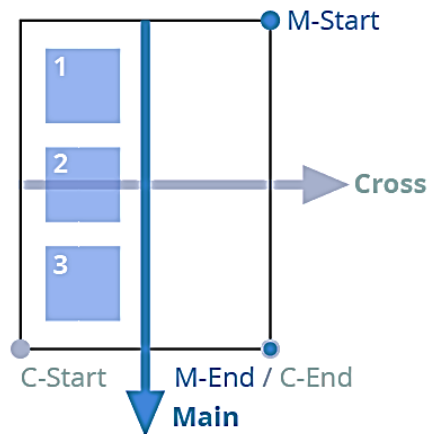
Modelo de layout Flexbox

- A propriedade **display: flex** transforma o elemento pai em um contêiner que possibilita a organização dos elementos filhos em uma única dimensão;
- No modelo de layout flex os filhos podem ser dispostos em linhas ou colunas e tornam-se flexíveis para alterações no tamanho, distribuição e alinhamento.

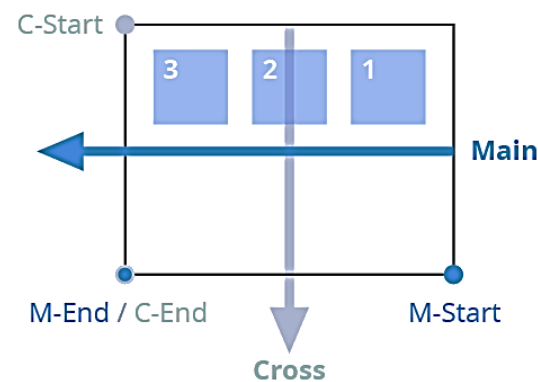
flex-direction: row



flex-direction: column



flex-direction: row-reverse



flex-direction: column-reverse

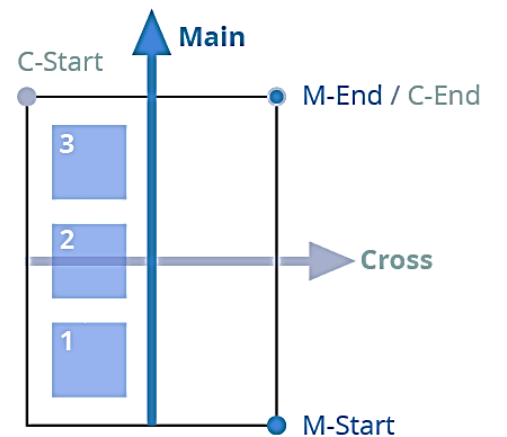




Table of contents

[Os eixos do *flexbox*](#)[Linhas de Início e Fim](#)[Contêiner *Flex*](#)[Quebra de linha com "*flex-wrap*"](#)[Propriedade abreviada *flex-flow*](#)[Expansão, encolhimento e tamanho dos elementos *flex*](#)[Alinhamento, justificação e distribuição de espaço livre entre os elementos](#)[Próximos passos](#)

Conceitos básicos de flexbox

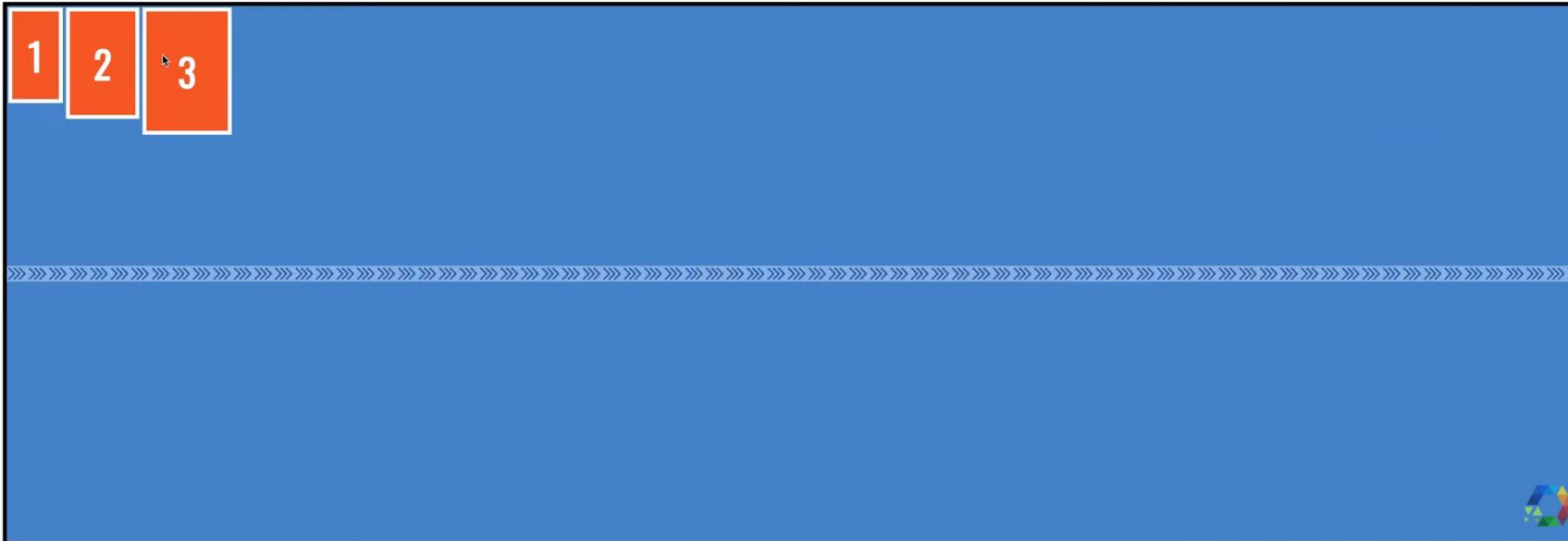
O *Flexible Box Module*, geralmente chamado de *flexbox*, foi projetado tanto como um modelo de *layout* unidimensional quanto como um método capaz de organizar espacialmente os elementos em uma interface, além de possuir capacidades avançadas de alinhamento. Este artigo fornece um resumo das principais funcionalidades do *flexbox*, as quais exploraremos com mais detalhes no restante deste guia.

Quando se descreve o flexbox como sendo unidimensional, enfatiza-se o fato de que ele lida com o *layout* em uma dimensão de cada vez - seja uma linha ou uma coluna. Isto pode ser comparado com o modelo bidimensional de [CSS - Layout de Grade](#), que permite o controle simultâneo das colunas e linhas.

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Flexible_Box_Layout

Entendendo o Flexbox...

Flex Container	Flex Item 1	Flex Item 2	Flex Item 3
flex-direction: row	Ativar: <input type="checkbox"/>	Ativar: <input type="checkbox"/>	Ativar: <input type="checkbox"/>
flex-wrap: nowrap	flex-grow: 0	flex-grow: 0	flex-grow: 0
justify-content: flex-start	align-self: flex-start	align-self: flex-start	align-self: flex-start
align-items: flex-start			
align-content: flex-start			

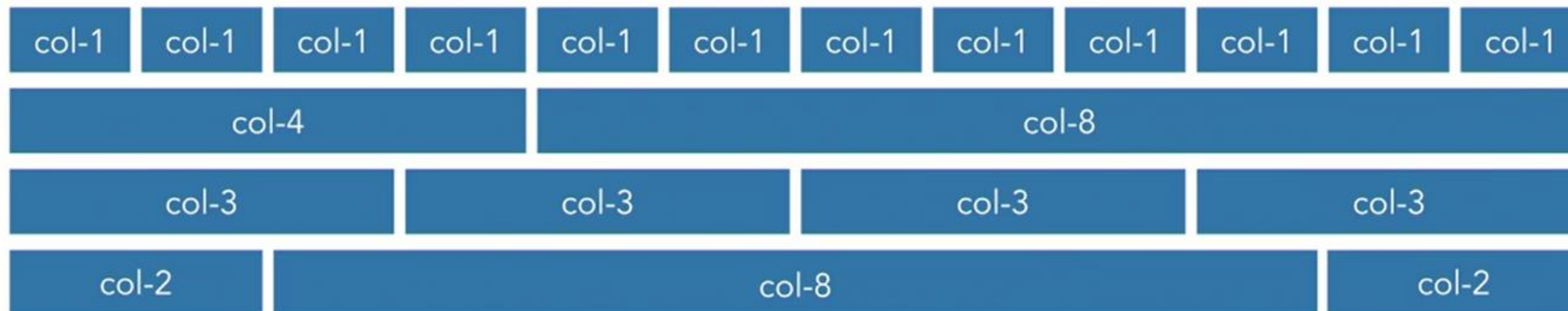


<https://youtu.be/s-CARPA01NU>

<https://github.com/cod3rcursos/youtube>

Modelo de layout Grid

- A propriedade **display: grid** transforma o elemento pai em um contêiner que possibilita a organização dos elementos filhos de forma bidimensional;
- No modelo de layout em grade os filhos podem ser dispostos no formato de linhas e colunas.

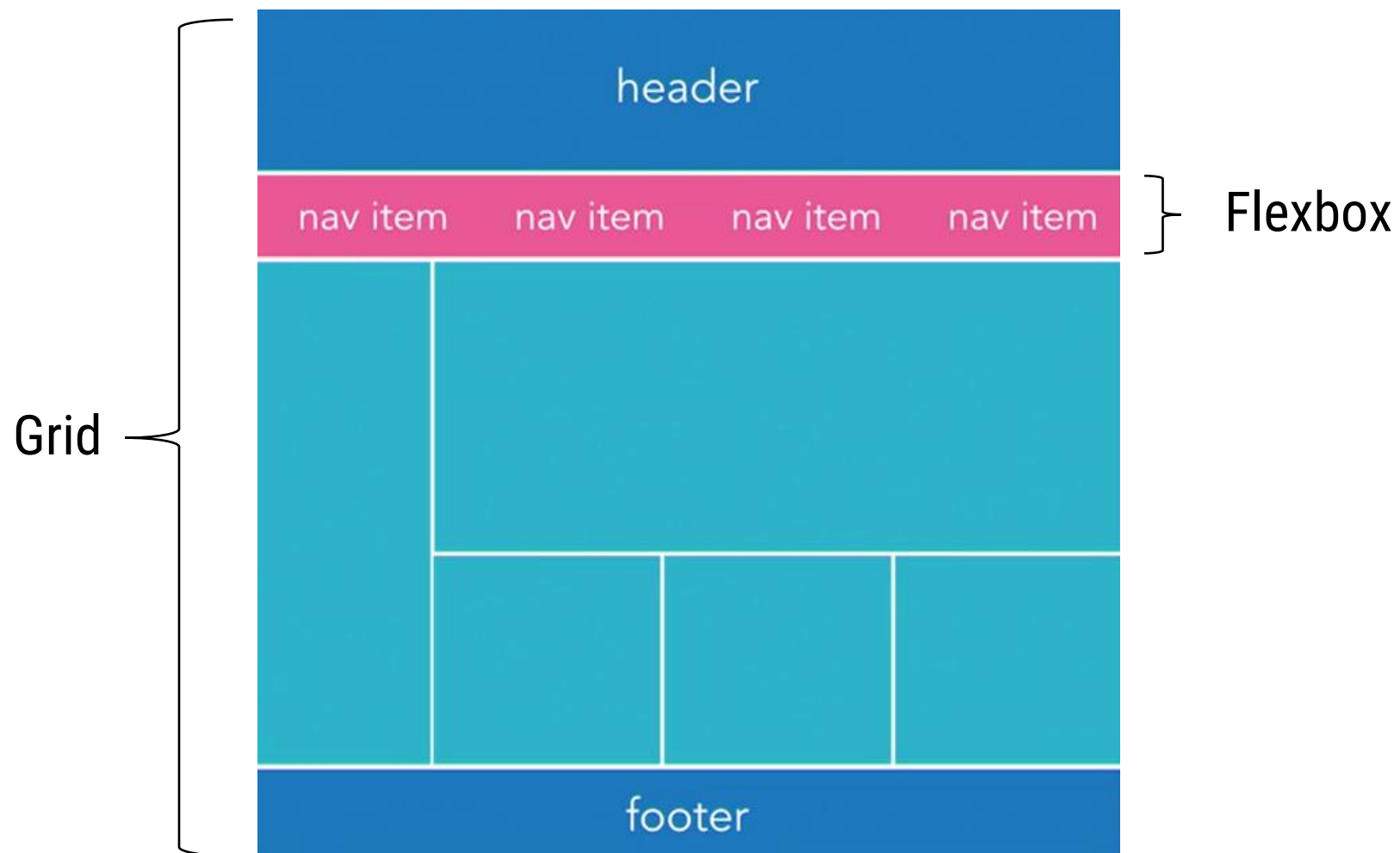


https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout

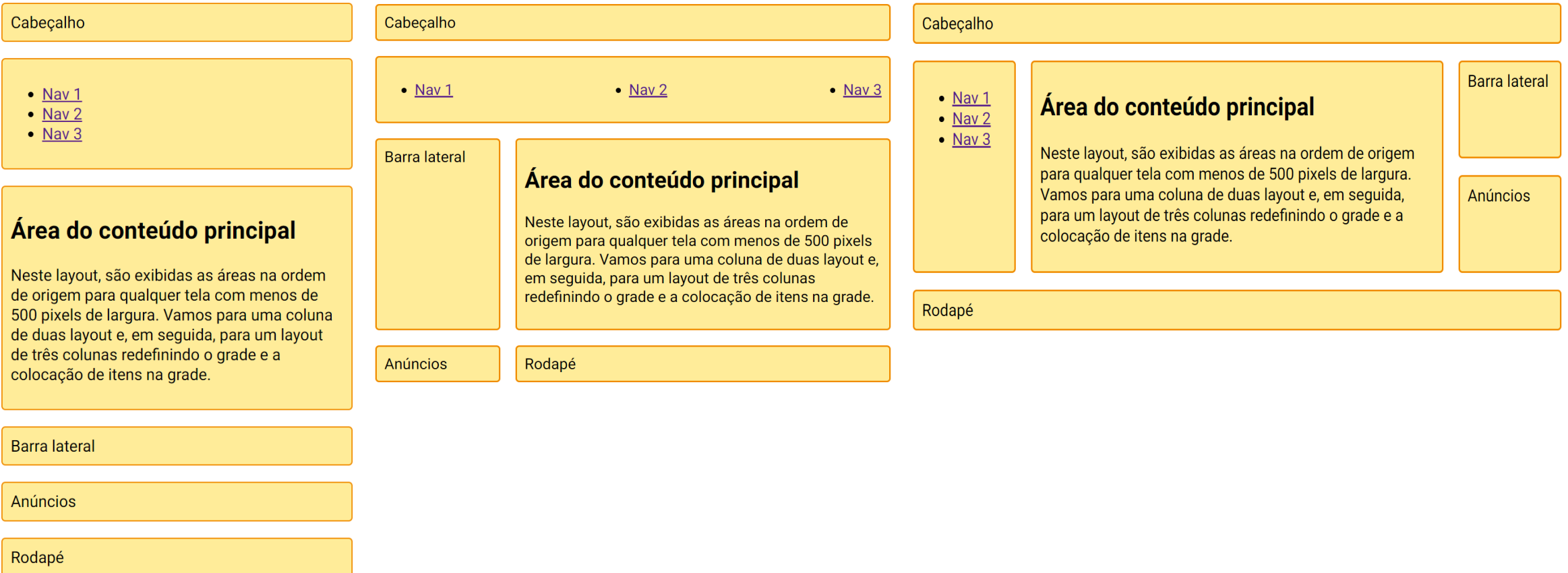
Flexbox ou grid layout?

- Para controlar layout por linha **ou** por coluna (unidimensional): **flexbox**. Um caso de uso ideal para o flexbox é quando você tem um conjunto de itens e deseja espaçá-los uniformemente em um contêiner;
- Para controlar layout por linha **e** coluna (bidimensional): **grid layout**. Ao usar grid layout, você cria um layout e então coloca os itens na posição que deseja das células da grade, ou permite que as regras de auto posicionamento coloquem os itens.

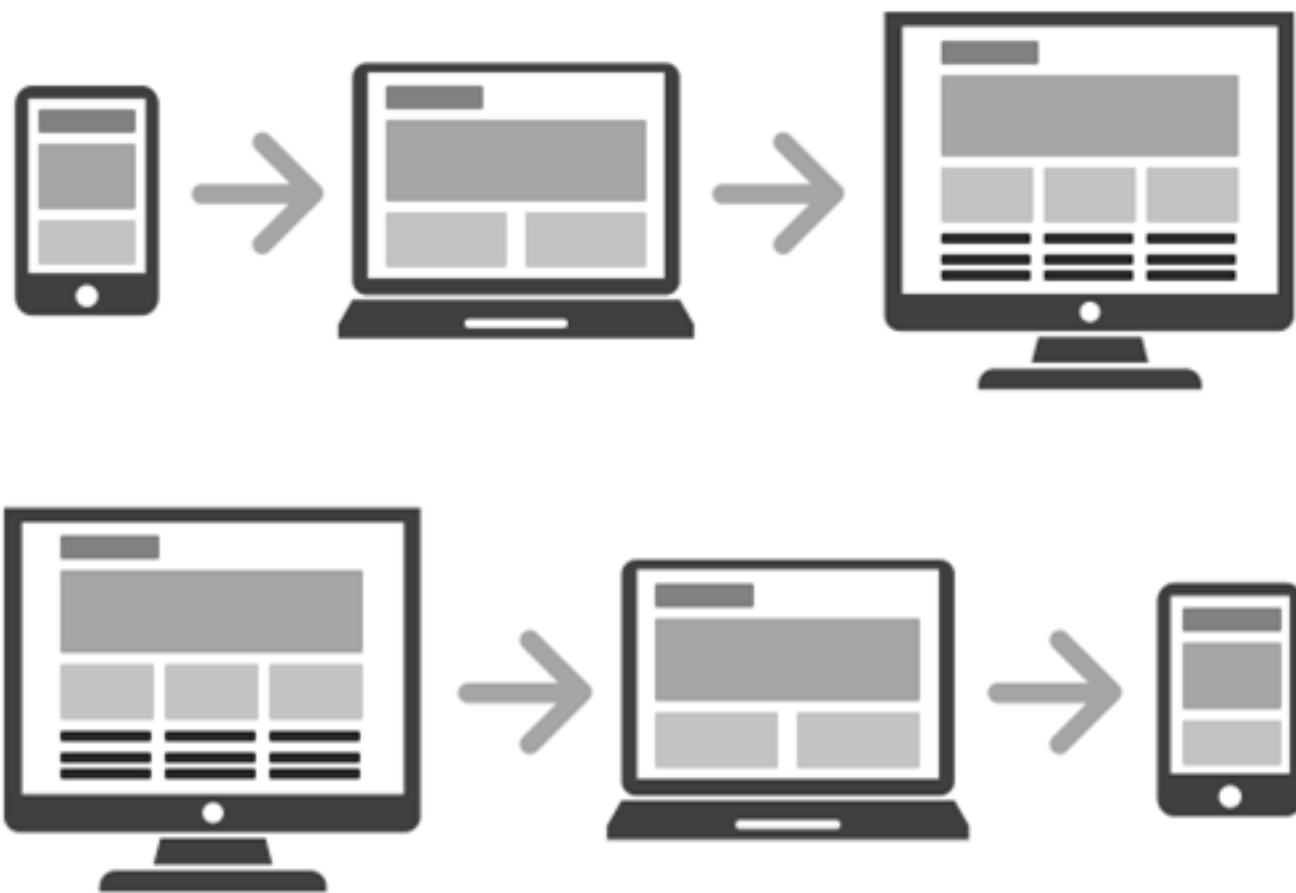
Combinação de recursos de layout



Layout **em grade** responsivo com 3 colunas fluídas



Responsividade: mobile-first ou desktop-first?



Layouts responsivos com @media query

```
/* Aplica CSS para todos os dispositivos */
```

```
@media all {...}
```

```
/* Aplica CSS para impressoras */
```

```
@media print {...}
```

```
/* Aplica CSS para leitores de texto */
```

```
@media speech {...}
```

```
/* Aplica CSS para todas as telas */
```

```
@media screen {...}
```

```
/* Aplica CSS para telas de até 1000px */
```

```
@media screen and (max-width: 1000px) {...}
```

Mobile First

```
/* Otimizado para telas pequenas */
```

```
body {...}
```

```
/* Alterações para telas maiores */
```

```
@media (min-width: 800px ) {
```

```
    body {...}
```

```
}
```

```
@media (min-width: 1200px ) {
```

```
    body {...}
```

```
}
```

Desktop First

```
/* Otimizado para telas grandes */  
body {...}  
  
/* Alterações para telas menores */  
@media (max-width: 800px ) {  
    body {...}  
}  
  
@media (max-width: 400px ) {  
    body {...}  
}
```

Vinculando diferentes folha de estilos

<!-- Usa a tag <link> para carregar a folha de estilos -->

```
<link rel="stylesheet" media="screen" href="desktop.css">
```

```
<link rel="stylesheet" media="screen and (max-width: 800px)"  
href="tablet.css">
```

```
<link rel="stylesheet" media="screen and (max-width: 400px)"  
href="mobile.css">
```