

CSS: ementa

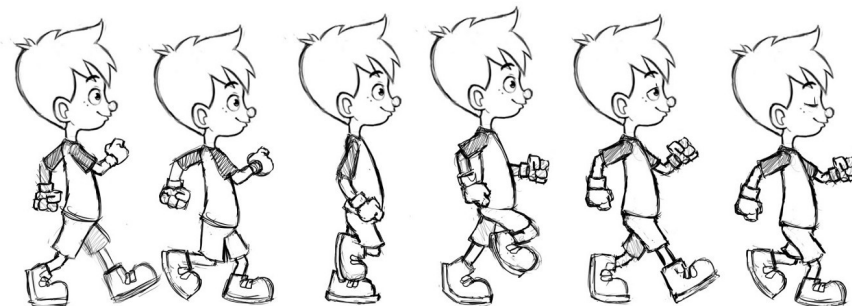
- Animação
- Transição
- Gradiente
- PARTES típicas de uma página web
- Design Responsivo



CSS: animação

Propriedades:

- @keyframes
- animation-name
- animation-duration
- animation-delay
- animation-iteration-count
- animation-direction
- animation-timing-function
- animation



Veja: https://www.w3schools.com/css/css3_animations.asp

CSS: animação

```
<head>
<style>
div {
  width: 100px; height: 100px; background-color: red; position: relative;
  animation-name: sem-loop; animation-duration: 4s;
}
@keyframes sem-loop {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; left:200px; top:0px;}
  50% {background-color:blue; left:200px; top:200px;}
  75% {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<h1>Animação com CSS </h1>
<div></div>
<p><b>Animação com CSS - propriedade @keyframes.</b></p>
</body>
```

animation: sem-loop 4s;

CSS: animação

```
<head>
<style>
div {
  width: 100px; height: 100px; background: red; position: relative;
  animation: com-loop 4s; animation-iteration-count: infinite;
}
@keyframes com-loop {
  0% {background-color:red; left:0px; top:0px;}
  25% {background-color:yellow; left:200px; top:0px;}
  50% {background-color:blue; left:200px; top:200px;}
  75% {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<h1>Animação com CSS </h1>
<div></div>
<p><b>Animação com CSS - propriedade @keyframes.</b></p>
</body>
```

CSS: animação

Outra forma:

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  position: relative;  
  animation: com-loop 4s infinite;  
}
```

Observação: o uso do comando

animation-delay: 2s;

Faz a animação esperar 2s para começar.

CSS: animação

Para a animação funcionar em sentido inverso:

animation-direction: reverse;

Para a animação alternar nos sentidos (de cima para baixo):

animation-direction: alternate;

Para a animação alternar nos sentidos (de baixo para cima):

animation-direction: alternate-reverse;

CSS: animação

A propriedade **animation-timing-function** pode ter os seguintes valores:

- **ease** - Especifica uma animação com início lento, rápido e final lento (este é o padrão)
- **linear** - Especifica uma animação com a mesma velocidade do início ao fim
- **ease-in** - Especifica uma animação com um início lento
- **ease-out** - Especifica uma animação com um final lento
- **ease-in-out** - Especifica uma animação com início e fim lentos
- **cubic-bezier(n,n,n,n)** - Permite definir seus próprios valores em uma função cubic-bezier

CSS: animação

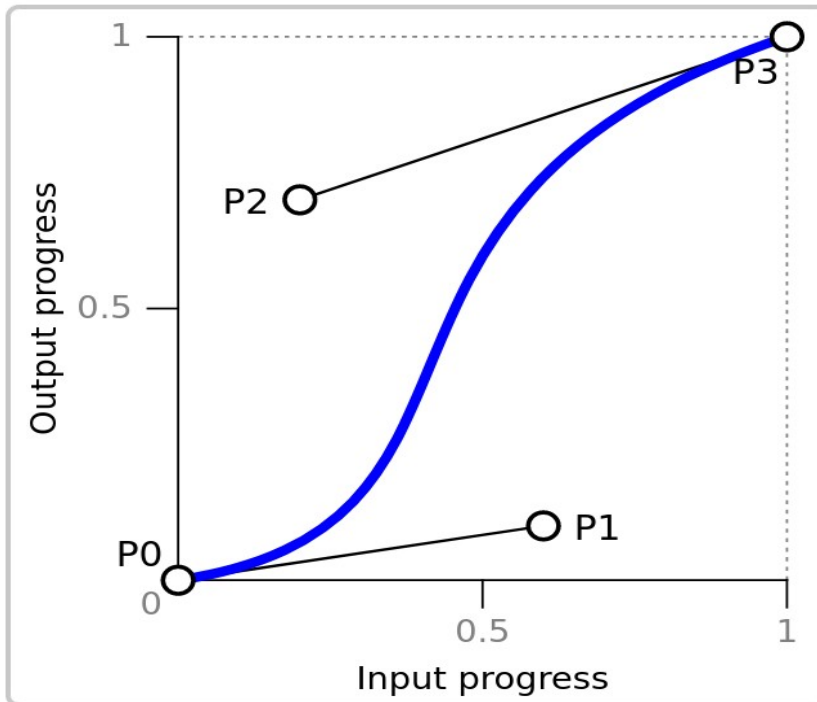
```
<head>
<style>
div {
  width: 100px; height: 100px;
  background: red; position: relative;
  animation: na-reta 5s infinite;
  animation-timing-function: linear;
}
@keyframes na-reta {
  from {left: 0px;}
  to   {left: 300px;}
}
</style>
</head>
<body>
<h1>Animação com CSS </h1>
<div></div>
<p><b>Animação com CSS - propriedade @keyframes.</b></p>
</body>
```


CSS: animação

```
<head>
<style>
div {
  width: 100px; height: 100px;
  background: red; position: relative;
  animation: na-reta 5s infinite;
  transition-timing-function: cubic-bezier(0.3, 0.7, 1.0, 0.1);
}
@keyframes na-reta {
  from {left: 0px;}
  to   {left: 300px;}
}
</style>
</head>
<body>
<h1>Animação com CSS </h1>
<div></div>
<p><b>Animação com CSS - propriedade @keyframes.</b></p>
</body>
```

CSS: animação

// A cubic Bézier easing function is a type of [easing function](#) defined by four real numbers that specify the two control points, $P1$ and $P2$, of a cubic Bézier curve whose end points $P0$ and $P3$ are fixed at $(0, 0)$ and $(1, 1)$ respectively. The x coordinates of $P1$ and $P2$ are restricted to the range $[0, 1]$.



<https://css-tricks.com/advanced-css-animation-using-cubic-bezier/>

CSS: transição

Propriedades

- transition
- transition-delay
- transition-duration
- transition-property
- transition-timing-function

CSS: transição

```
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: red;
  transition: width 2s;
  transition-timing-function: ease-out;
}
div:hover {
  width: 400px;
}
</style>
</head>
<body>
<h1>Transição com CSS </h1>
<div></div>
<p><b>Exemplo de transição.</b></p>
</body>
```

```
transition-property: width;
transition-duration: 2s;
```

CSS: transição

```
<head>
<style>
div {
  border-style: solid; border-width: 1px; display: block;
  width: 100px; height: 100px; background-color: darkblue;
  -webkit-transition: width 2s, height 2s, background-color 2s, -webkit-transform 2s;
  transition: width 2s, height 2s, background-color 2s, transform 2s;
}
div:hover {
  background-color: yellow; width: 300px; height: 300px;
  -webkit-transform: rotate(180deg);
  transform: rotate(180deg);
}
</style>
</head>
<body>
<h1>Transição com CSS </h1>
<div></div>
<p><b>Exemplo de Transição</b></p>
</body>
```

CSS: transição

```
div {
    border-style: solid;
    border-width: 1px;
    display: block;
    width: 100px;
    height: 100px;
    background-image: url("girassol.png");
    -webkit-transition: width 2s, height 2s, background-image 2s, -
webkit-transform 2s;
    transition: width 2s, height 2s, background-image 2s, transform
2s;
}
div:hover {
    background-image: url("tulipa.png");
    width: 300px;
    height: 300px;
    -webkit-transform: rotate(180deg);
    transform: rotate(180deg);
}
```

CSS: transição

```
<head>
<style>
p {
  background-color: lightblue;
}
p:hover {
  background-color: yellow;
  -webkit-transition: background-color 1000ms linear;
  -ms-transition: background-color 1000ms linear;
  transition: background-color 1000ms linear;
}
</style>
</head>
<body>
<h1>Transição com CSS </h1>
<div></div>
<p><b>Exemplo de Transição</b></p>
</body>
```

CSS: transição

```
<head>
<style>
img {
  padding: 50px;
  transition: transform .2s;
  width: 200px; height: 200px; margin: 0 auto;
}
img:hover {
  -ms-transform: scale(2.0);    /* IE 9 */
  -webkit-transform: scale(2.0); /* Safari 3-8 */
  transform: scale(2.0);
}
</style>
</head>
<body>
<h1>Zoom na imagem</h1>
<div align="center">
  
</div>
</body>
```


CSS: transição

Bordas arredondadas

```
img {  
  padding: 50px;  
  width: 200px;  
  height: 200px;  
  margin: 0 auto;  
}  
img:hover {  
  border-radius: 50%;  
}
```

CSS: gradiente

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: blue;
  background-image: linear-gradient(blue, yellow);
}
</style>
</head>
<body>
<h1>Gradiente</h1>
<p>Teste de gradiente</p>
</body>
</html>
```

CSS: gradiente

Outros Exemplos:

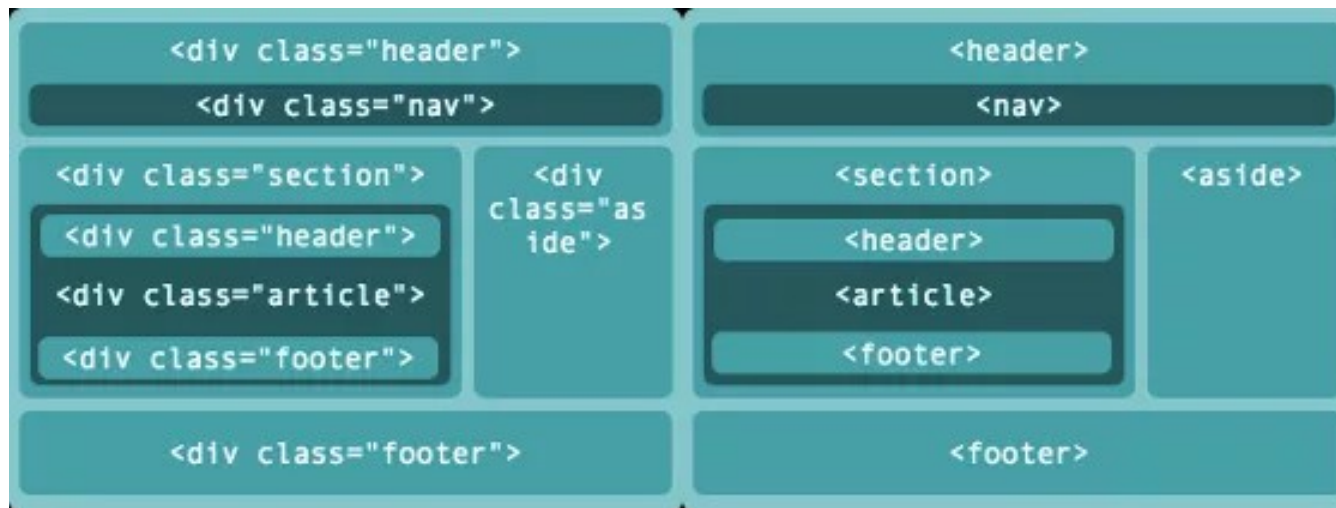
- background-image: linear-gradient(red, orange, yellow, green, blue, indigo, violet);
- background-image: linear-gradient(red 10%, green 85%, blue 90%);
- background-image: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
- background-image: linear-gradient(to right, red, orange, yellow, green, blue, indigo, violet);
- background-image: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));
- background-image: repeating-linear-gradient(45deg, red 10px, yellow 30px);

Partes típicas de uma página de um site

Na HTML 4, usava-se **id/class** para nomear os elementos de estilo: header, top, bottom, footer, menu, navigation, main, container, contente, article, sidebar, topnav, etc.

HTML 5 - Elementos de Semântica:

Tornou impossível para os motores de busca identificar o conteúdo da página web correta.



<http://www.uxdesign.blog.br/front-end/html5-estrutura-semantica/>

Partes típicas de uma página de um site

Typical HTML4

`<div id="header">`

`<div id="menu">`

`<div id="content">`

`<div class="article">`

`<div id="footer">`

Typical HTML5

`<header>`

`<nav>`

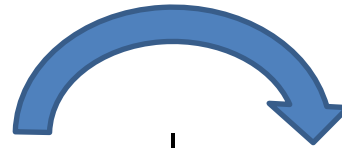
`<section>`

`<article>`

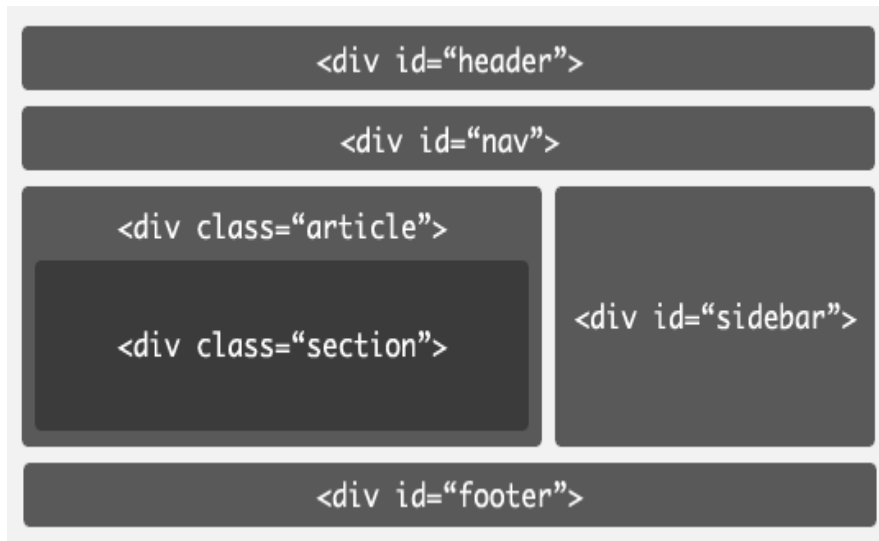
`<footer>`

Partes (SECTIONS): elementos introduzidos no HTML 5

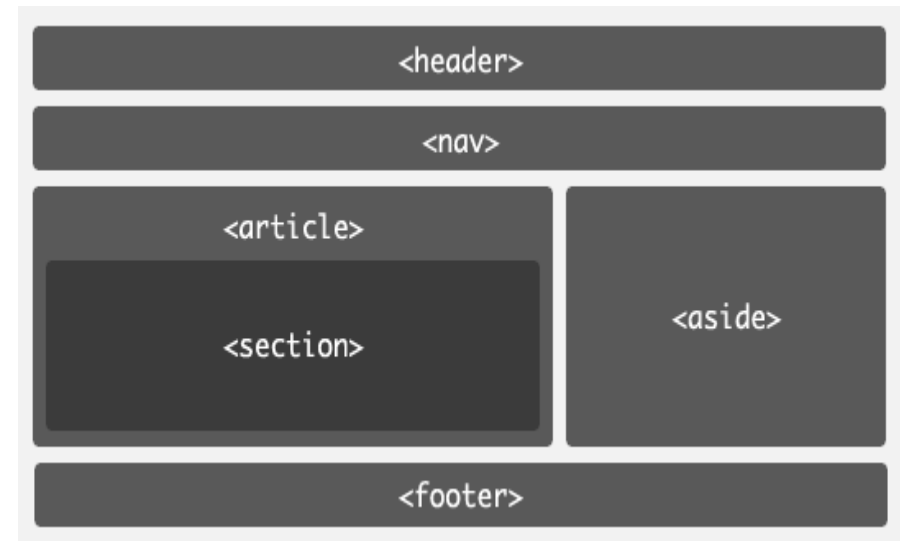
Por quê?



XHTML, HTML 4.01 e anterior



HTML 5



Partes típicas de uma página de um site

<article> identifica um conteúdo independente, autossuficiente

<aside> define um conteúdo lateral

<details> define detalhes adicionais que pode ficar visível ou oculto

<figcaption> adiciona uma legenda a uma imagem

<figure> agrupa imagens e legendas

<footer> elemento que especifica o rodapé do documento ou seção

<header> especifica o cabeçalho do documento ou seção

<main> especifica o conteúdo principal de um documento

<mark> define textos realçados

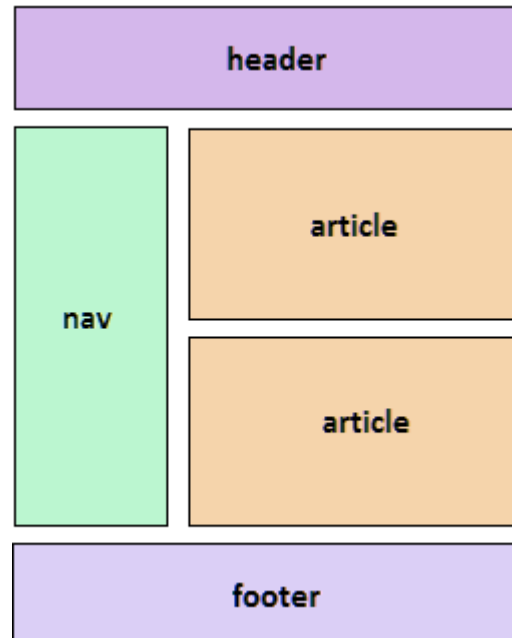
<nav> elemento que define um conjunto de links de navegação

<section> define uma seção de um documento

<summary> define um título visível para um elemento <details>

<time> define data/hora

Partes típicas de uma página de um site



Partes típicas de uma página de um site

“The HTML **<main>** tag is used to represent the main content area within an HTML document.

The **<main>** tag surrounds the main content of the page - content that is unique to that document and is obviously the "main" content for that page. This excludes any content that is repeated across multiple pages (such as navigation bars, headers, footers, etc).

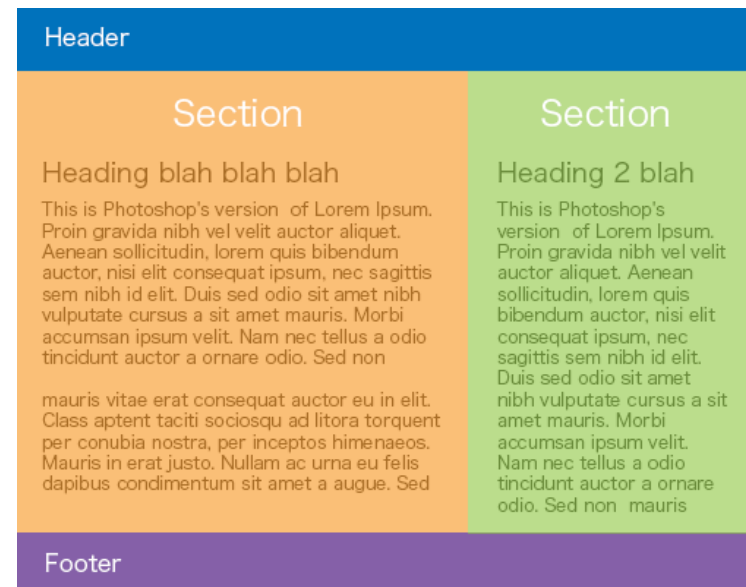
An HTML document can have a **maximum of one <main>** element. It must not appear within the <article>, <aside>, <footer>, <header> or <nav> tags.”

Fonte: https://www.quackit.com/html_5/tags/html_main_tag.cfm

Partes típicas de uma página de um site

Elemento <section>

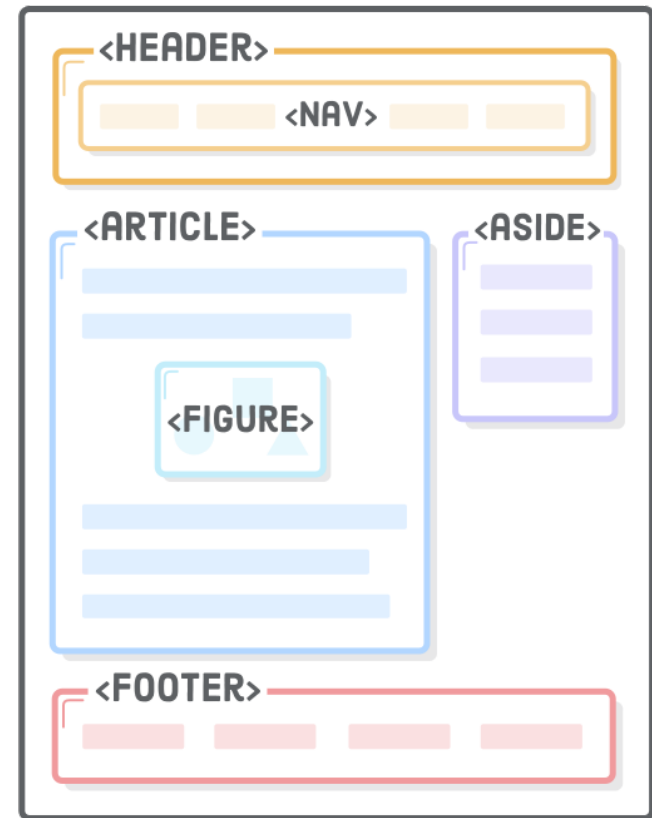
- Define uma seção no documento
- Para o W3C, uma seção é um agrupamento temático de um conteúdo tipicamente com um cabeçalho.
- Exemplos: introdução, conteúdo, contato, etc.



Partes típicas de uma página de um site

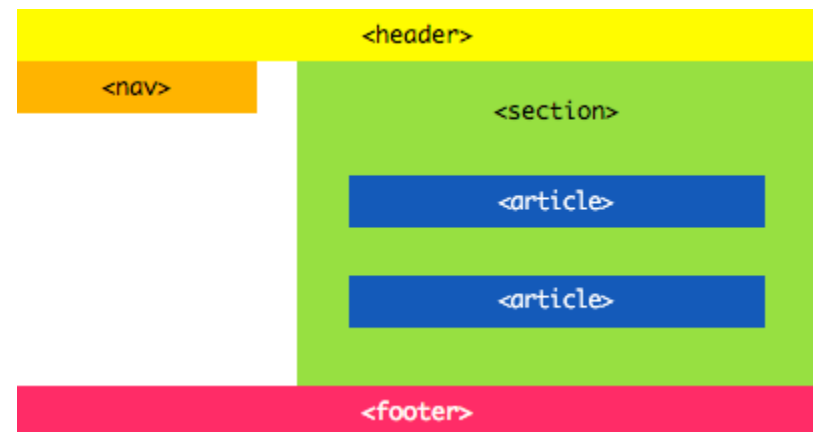
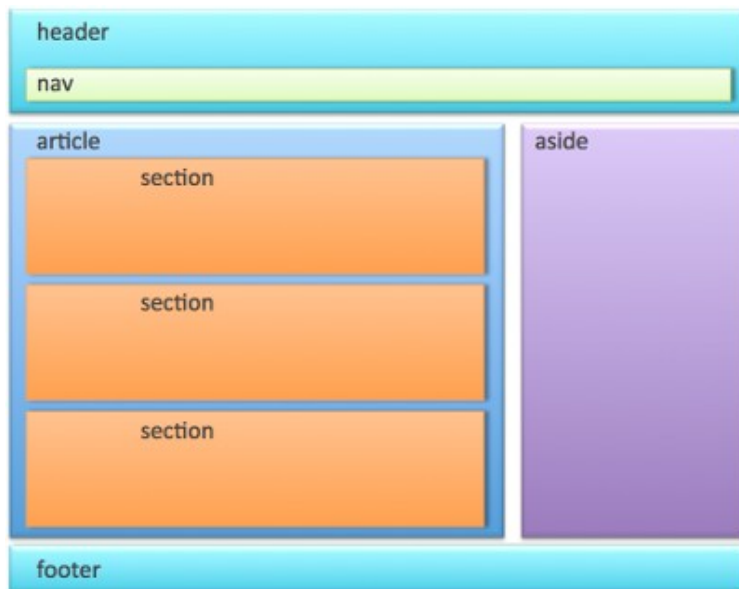
Elemento `<article>`

- Especifica conteúdo autônomo e independente (é compreensível quando lido sozinho).
- Exemplos: fórum, blog, artigo de jornal, etc.



Partes típicas de uma página de um site

Como agrupar `<section>` e `<article>` ?

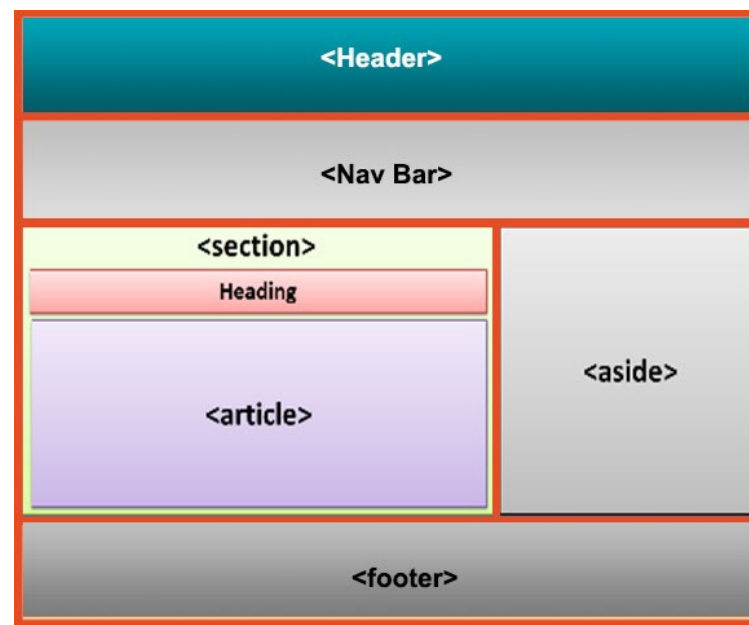


- Um elemento `<section>` pode ter diferentes elementos `<article>`
- Um elemento `<article>` pode ter diferentes elementos `<section>`

Partes típicas de uma página de um site

Elemento <header>

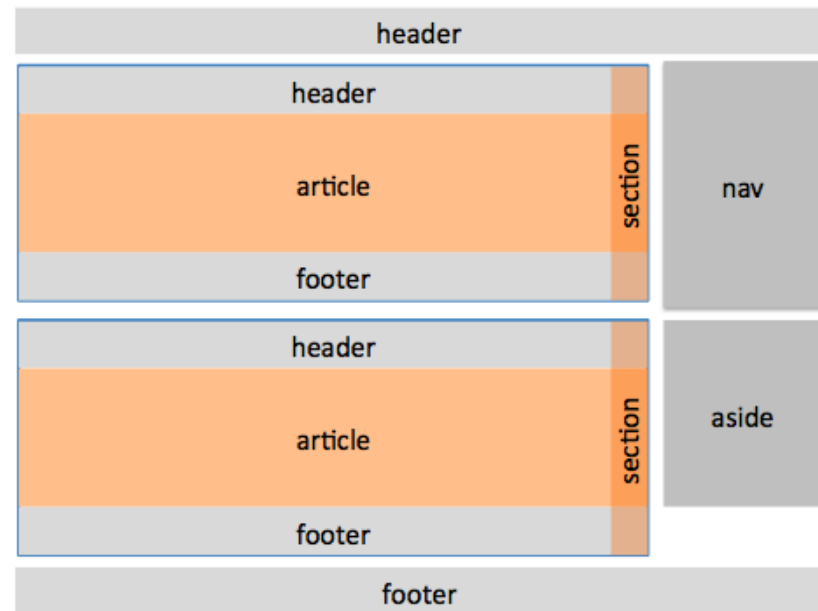
- Especifica um cabeçalho para um documento ou uma seção.
- Deve ser usado como um contêiner para um conteúdo introdutório.
- Pode-se ter vários elementos <header> em um documento.



Partes típicas de uma página de um site

Elemento <footer>

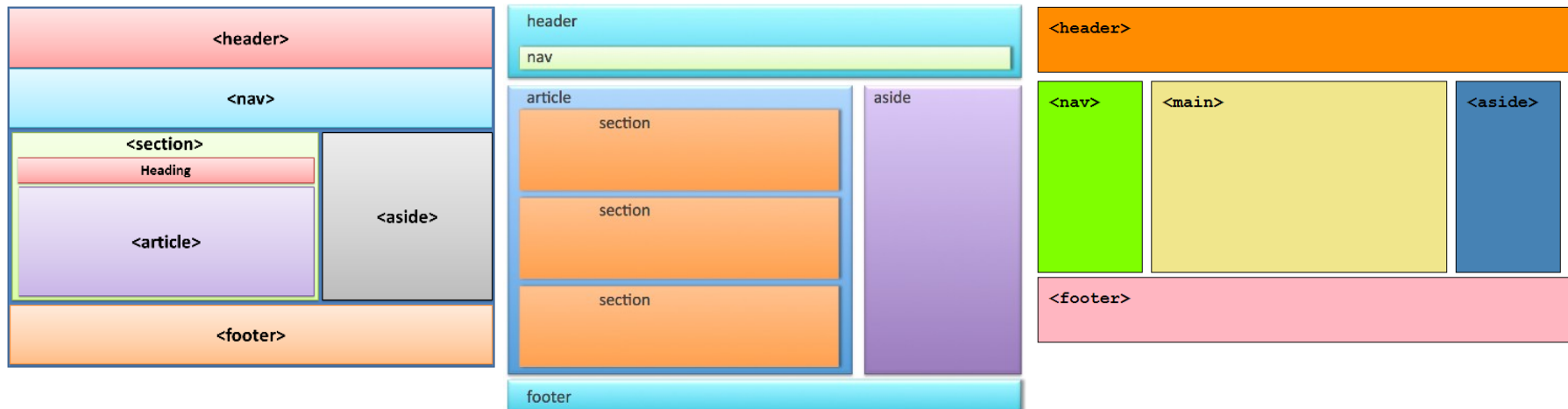
- Especifica um rodapé para um documento ou uma seção.
- Tipicamente contém informações sobre o autor, copyright (direitos autorais), termos de uso, contato, etc.
- Pode-se usar vários elementos <footer> em um documento.



Partes típicas de uma página de um site

Elemento `<nav>`

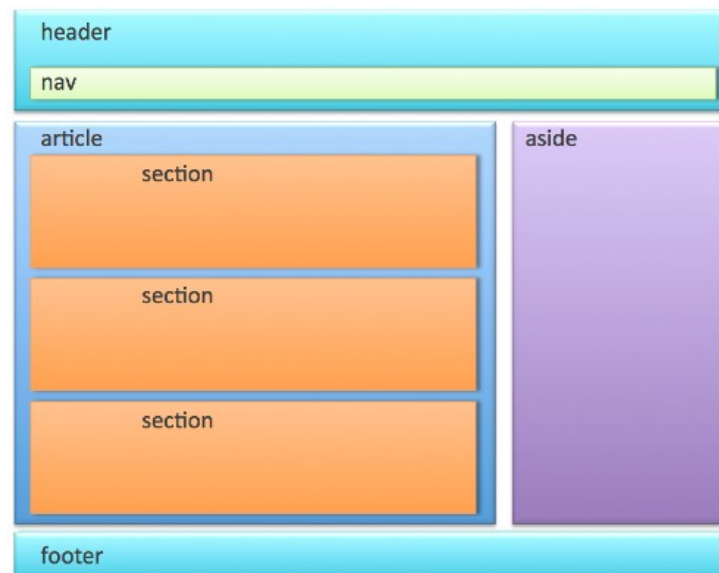
- Define um conjunto de links de navegação.
- Não precisa conter todos os links de navegação do documento.



Partes típicas de uma página de um site

Elemento `<aside>`

- Conteúdo colocado na lateral da página.



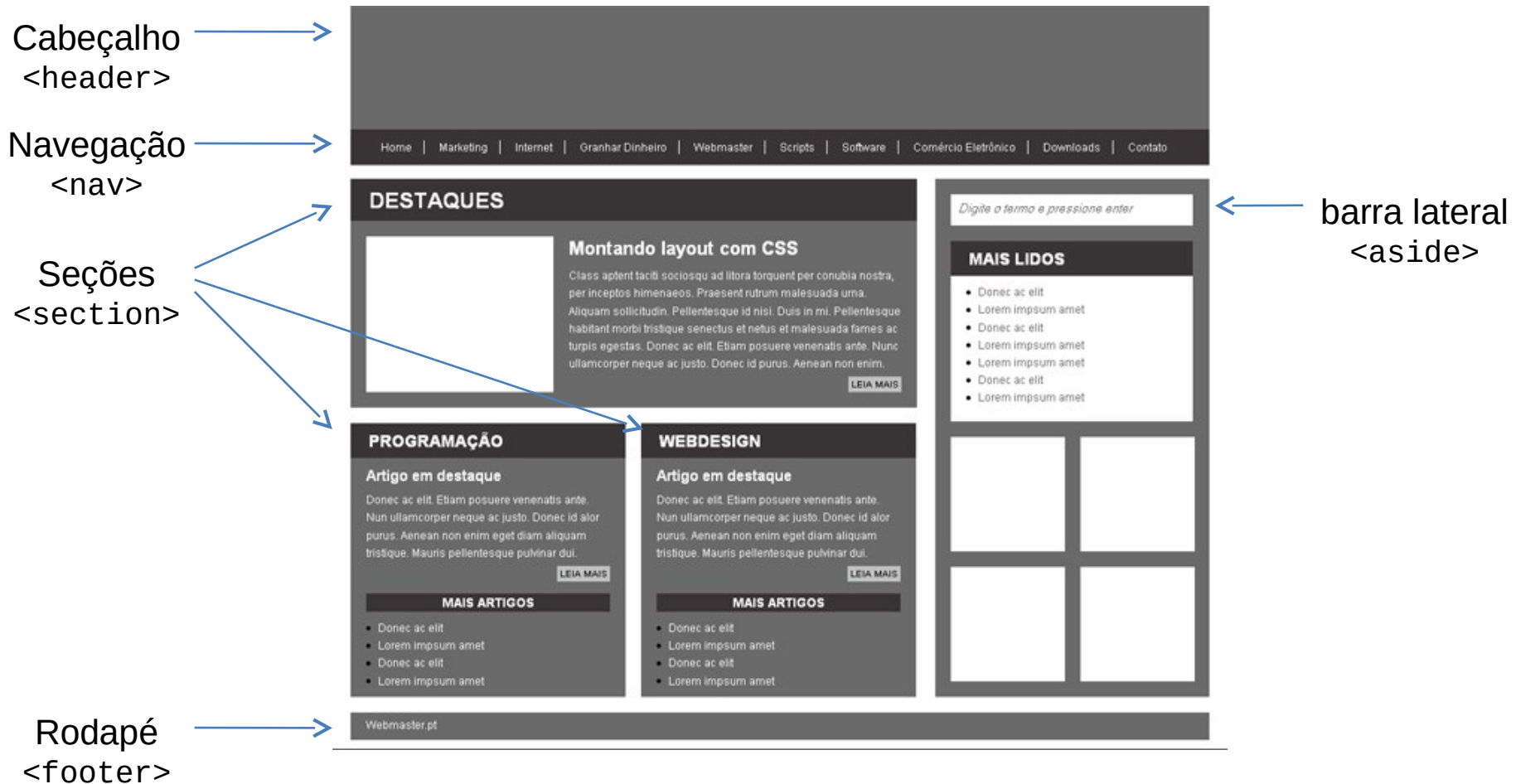
Partes típicas de uma página de um site

Elementos `<figure>` e `<figcaption>`

- O elemento `<figcaption>` adiciona um título a uma figura.
- A imagem e o elemento `<figcaption>` são agrupados dentro de um elemento `<figure>`.

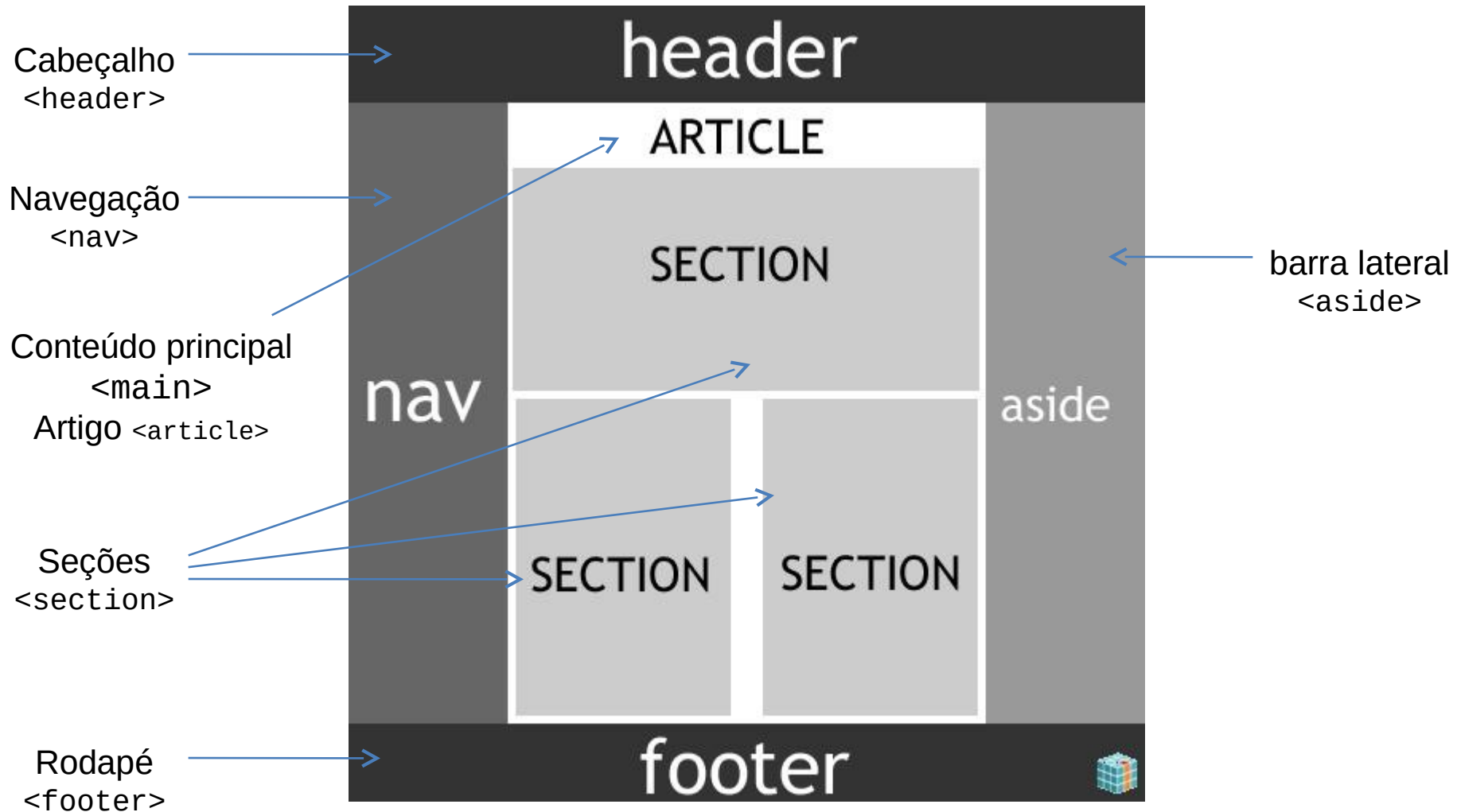
```
<figure>  
    
  <figcaption>GDG Feira de Santana</figcaption>  
</figure>
```

Partes típicas de uma página de um site



<http://www.webmaster.pt/html5-parte1-4955.html>

Partes típicas de uma página de um site



http://imasters.uol.com.br/artigo/16598/webstandards/html5_semantica_seo_e_organizacao

Design Responsivo

Definição

- Otimização estrutural e de design de sites para diferentes tipos de tela de dispositivos.
- O usuário pode navegar independente do dispositivo (computador, tablet ou celular).

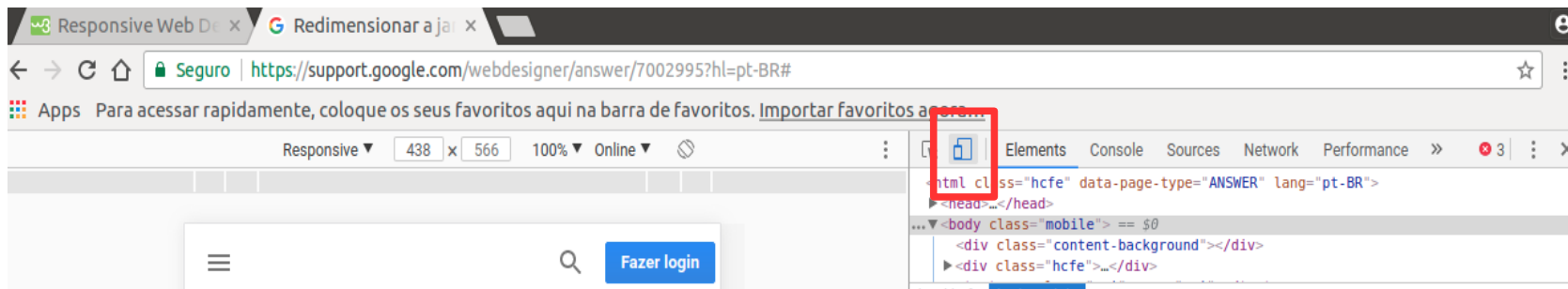


Design Responsivo

Atenção

Para poder redimensionar a tela:

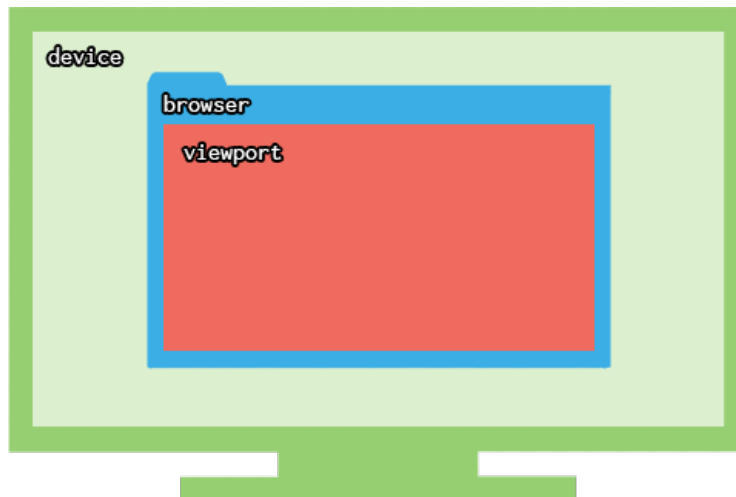
- No Firefox: ctrl + shift + m (Modo de Design Adaptável)
- No Chrome: 1) F12 + ctrl + shift + m; 2) Mais ferramentas → Ferramentas do desenvolvedor → toggler device toolbar



Design Responsivo

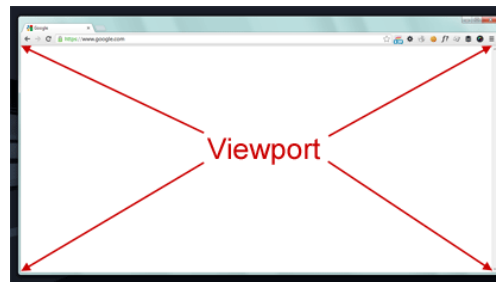
Viewport

- É a área visível de uma página web.
- O tamanho do *viewport* depende do equipamento e pode varia do tamanho de uma tela do celular a uma grande tela de desktop.
- Anteriormente, as páginas web tinha layout e tamanhos fixos.



Design Responsivo

- O HTML5 introduziu um método para controlar o *viewport*.
- Deve-se incluir o seguinte elemento em todas as páginas:
`<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- A parte "width=device-width" especifica que a página deve adotar o tamanho da tela como padrão.
- A parte "initial-scale=1.0" define que nenhum zoom será dado quando a página é inicialmente carregada.
- Esse elemento prepara a página para os diferentes dispositivos.



Design Responsivo

```
<!DOCTYPE html>
<html>
<head>
<title>Teste de HTML</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>



<p>Altere o tamanho da tela e veja o que acontece</p>

</body>
</html>
```


Design Responsivo

```
<head>
<title>Teste de HTML</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
.centralizar {
  display: block; margin-left: auto; margin-right: auto;
}
</style>
</head>
<body>
<picture>
  <source srcset="margarida.png" media="(max-width: 500px)">
  <source srcset="tulipa.png" media="(max-width: 700px)">
  
</picture>
</body>
```

- A ordem importa!
- max-height

Exercícios

- 1) O cor de background da página muda do azul para o amarelo e depois do amarelo para o azul em loop infinito.
- 2) Quando o mouse passa em cima de uma imagem; a imagem é trocada e aumenta de tamanho.
- 3) Quando o mouse passa em cima de uma imagem, aparece a borda da imagem.

