

Linguagens de Marcação e Scripts

Prof. Aníbal Cavalcante de Oliveira

UFC - QXD0164 - 2019.2

Agenda - Aula 11

- HTML semântico.
- Css seletores avançados e pseudo-elementos.

HTML 5 e as novas tags semânticas (que possuem significado)

HTML semântico é o uso de marcação HTML para reforçar a semântica, ou o significado, das informações em páginas da Web, em vez de simplesmente definir sua apresentação ou aparência.

Em outras palavras, como a chegada do HTML 5, o papel do HTML deixou de ser o de apenas estruturar os documentos e passou a dar significado aos conteúdos através das tags semânticas.



Problema do HTML4

Utilizava-se a tag `<div>` como um **coringa** do HTML para definir quase toda a estrutura das páginas, desde o cabeçalho ao rodapé.



Problemas do uso indiscriminado da tag <div>

1 - Usuários deficientes, que utilizam leitores de tela para navegação, não conseguem identificar cada conteúdo da tela e nem sua organização. O leitor interpreta a página como um grande aglomerado de tags <div>.

2 - Mecanismos de busca, como o do Google, não conseguem identificar qual é o conteúdo principal do seu site. Para eles, o id que é dado às tags <div> não possui significado algum.

3 - Para programadores, manter um site que possui um emaranhado de tags <div> aninhadas, umas dentro das outras, é extremamente trabalhoso, principalmente para entender onde começa e termina cada <div>, e saber o papel de cada uma no documento.

Solução

Com isso, o HTML5 traz alguns elementos para suprir essa falta de tags mais semânticas e descritivas para alguns elementos clássicos em diversos cenários, como cabeçalhos, menus de navegação e áreas de conteúdo secundário. Elas não alteram a visualização da página, apenas adicionam significado aos blocos de conteúdo.

Solução

`<header>`

O elemento `<header>` representa um contêiner para um conteúdo introdutório ou um conjunto de links de navegação. (Elemento do tipo bloco)

Um elemento `<header>` geralmente contém:

- 1 - um ou mais elementos de cabeçalho (`<h1>` - `<h6>`)
- 2 - logotipo ou ícone
- 3 - informação de autoria

* Você pode ter vários elementos `<header>` em um documento.

Obs1.: A tag `<header>` não pode ser colocada dentro de um `<footer>`, `<address>` ou outro elemento `<header>`.

Obs2.: O ideal é que a página possua apenas um elemento `<header>` principal. Porém podemos ter tags `<header>` dentro de tags `<section>`.

Solução

`<section>`

O elemento `<section>` define uma seção em um documento.

De acordo com a documentação HTML5 do W3C: "Uma seção é um agrupamento temático de conteúdo, normalmente com um cabeçalho".

Uma página normalmente pode ser dividida em várias seções como: introdução, conteúdo e links.

Obs1.: A tag `<section>` parece muito com uma tag `<div>` normal, mas não é.

Obs2.: Dica, antes de substituir uma tag `<div>` por uma tag `<section>`, pergunte-se sempre: "Todo o conteúdo dessa `<div>` está relacionado, possui o mesmo tema ou trata do mesmo assunto?" Se a resposta for verdadeira, você pode substituir a `<div>` pela `<section>`.

Solução

`<footer>`

A tag `<footer>` define um rodapé para um documento ou para uma section.

Um elemento `<footer>` geralmente contém:

- 1 - informação de autoria
- 2 - Informações sobre direitos autorais
- 3 - informações de contato
- 4 - o mapa do site
- 5 - links para redes sociais
- 6 - links para documentos relacionados

Obs.: O ideal é ter um elemento `<footer>` principal para sua página. Porém, você pode ter elementos `<footer>` dentro de elementos `<section>`.

Solução

`<nav>`

A tag `<nav>` define um conjunto de links de navegação da página.

O elemento `<nav>` destina-se apenas ao bloco principal de links de navegação.

Navegadores, como leitores de tela para usuários com deficiência, podem usar esse elemento para determinar se omitem a renderização inicial desse conteúdo.

Obs.: Observe que nem todos os links de um documento devem estar dentro de um elemento `<nav>`. Use apenas para os links de navegação da página.

Solução

`<aside>`

A tag `<aside>` pode ser utilizada para representar uma seção de conteúdo secundário ou um conteúdo auxiliar que completa outro conteúdo de maior importância. Exemplos:

- 1 - Citações;
- 2 - Links de referência;
- 3 - Notas adicionais

Obs.: Só porque algum conteúdo aparece à esquerda ou à direita do conteúdo principal não é motivo suficiente para usar o elemento `<aside>`. Pergunte a si mesmo se o conteúdo do `<aside>` pode ser removido sem reduzir o significado do conteúdo principal. Se a resposta for verdadeira ele é do tipo `<aside>`.

Solução

`<article>`

A tag `<article>` especifica um conteúdo independente.

Um `<article>` deve fazer sentido por si só e deve ser possível distribuí-lo independentemente do resto do site.

Fontes potenciais para o elemento `<article>`:

- 1 - um post em um fórum
- 2 - um post em blog
- 3 - uma notícia de um portal
- 4 - um comentário em uma lista de comentários

Obs1.: Um `article` deve fazer sentido por conta própria. Antes de substituir `<div>` por elementos do tipo `<article>`, pergunte-se sempre: "É possível lê-lo independentemente do resto do site?"

Obs2.: Podemos ter uma tag `<section>` com vários elementos `<article>` dentro dela.

Solução

`<main>`

A tag `<main>` deve circundar o conteúdo principal da página.

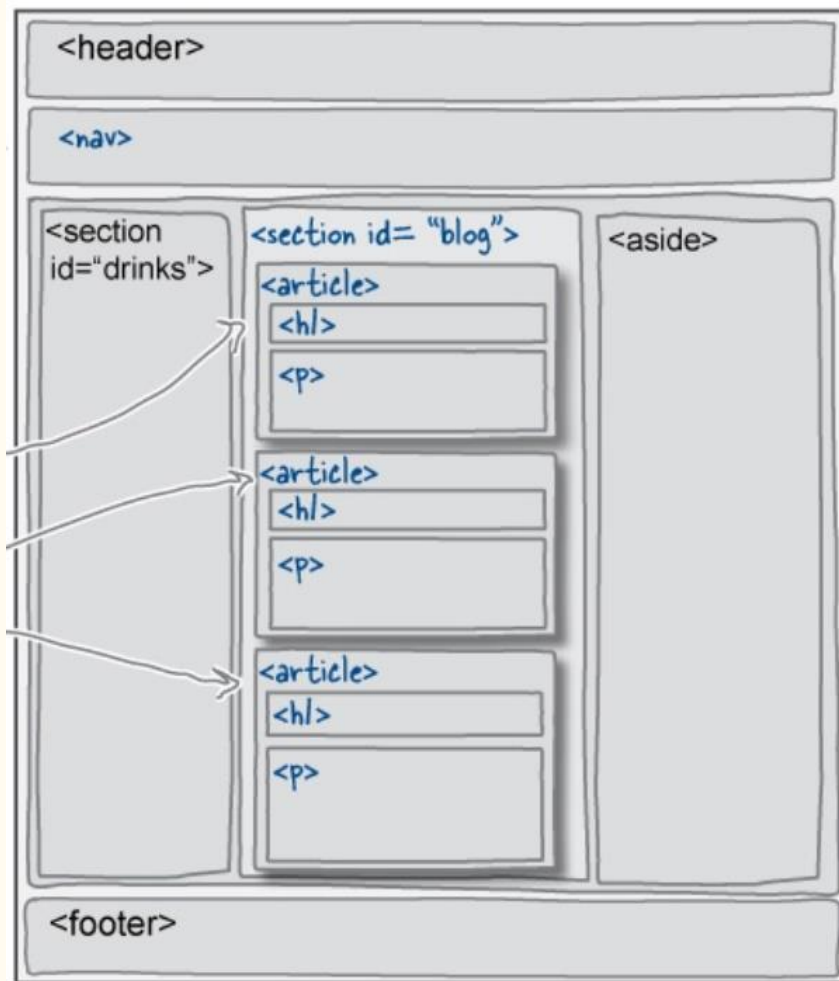
Ela tem uma única função semântica: definir o conteúdo principal da página ou da aplicação.

Ele representa o conteúdo mais importante da página, que está diretamente relacionado ao tópico central do documento ou a funcionalidade principal de uma aplicação.

Obs1.: Só pode ser utilizada uma única vez em cada documento html.

Obs2.: Não pode ser inserida dentro das tags `<article>`, `<aside>`, `<footer>`, `<header>`, ou `<nav>`

Exemplos



CSS Avançado

Como já vimos utilizamos seletores em CSS são padrões usados para selecionar elementos na página HTML que queremos estilizar. Agora, vamos ver a lista de seletores em CSS.

1 - Seletor Universal (*) - É utilizado quando queremos aplicar um estilo a todos os elementos. Podemos utilizá-lo para zerar os estilos nativos de cada elemento.

```
* {  
  margin: 0;  
  padding: 0;  
}
```

CSS Avançado

2 - Seletor de Elemento - Usado para estilizar os elementos da marcação que são de um mesmo tipo. Por exemplo: elemento do tipo <p>, <div>, , e assim por diante.

Obs.: Podemos adicionar o mesmo estilo a mais de um elemento, separando por vírgula os elementos.

```
a {  
    text-decoration: none;  
}  
  
h1, p {  
    background-color: yellow;  
}
```


CSS Avançado

3 - Seletor de Identificador - Usado para estilizar os elementos únicos na página. O elemento deverá ter o atributo “id”. É possível colocar o nome do elemento junto ao seu id no Css.

```
#container {  
    width: 960px;  
    margin: auto;  
}
```

```
div#container {  
    width: 960px;  
    margin: auto;  
}
```

CSS Avançado

4 - Seletor de Classe - Use uma classe quando você precisar estilizar um grupo de elementos. Todos os elementos desse grupo terão o atributo “**class**”. É possível colocar o nome do elemento junto ao nome da classe no Css.

```
.error {  
  
    color: red;  
}  
  
span.texto {  
    display: inline-block;  
    width: 100px;  
    height: 100px;  
}
```

CSS Avançado

5 - Seletor de Descendência - Permite ser mais específico, ou seja, selecionado apenas os elementos filhos de um outro elemento.

```
li a {  
    text-decoration: none;  
}  
  
#drinks h1 {  
    font-size: 120%;  
}  
  
.beanheading img {  
    display: block;  
}
```

CSS Avançado

6 - Seletor de irmão imediatamente adjacente (+) - Permite selecionar o elemento imediatamente após o primeiro elemento. No exemplo abaixo, selecionará só o primeiro parágrafo após cada ul na página, e fará que ele tenha a cor vermelha.

```
ul + p {  
    color: red;  
}  
  
<ul>  
    <li> List Item </li>  
    <li> List Item </li>  
</ul>  
<p> Lorem ipsum dolor sit amet. </p>
```

CSS Avançado

7 - Seletor descendente direto (>) - Semelhante ao seletor descendente, porém só seleciona os filhos diretos, e não seleciona netos, bisnetos e outros descendentes.

```
div#container > ul {  
    border: 1px solid black;  
}  
  
<div id="container">  
    <ul>  
        <li> Item da lista  
            <ul>  
                <li> Filho </li>  
            </ul>  
        </li>  
        <li> Item da lista </li>  
    </ul>  
</div>
```

CSS Avançado

8 - Seletor de irmão geral (~) - Semelhante ao seletor adjacente, porém seleciona qualquer elemento que apareça após o primeiro elemento.

```
ul ~ p {  
    color: red;  
}
```

```
<div id="container">  
    <ul>  
        <li> List Item </li>  
    </ul>  
    <p> Lorem ipsum dolor sit amet.</p>  
    <p> Lorem ipsum dolor sit amet.</p>  
    <p> Lorem ipsum dolor sit amet.</p>  
</div>
```

9 - Seletor de atributos - Seleciona apenas as tags que possuem um determinado atributo. No exemplo abaixo, ele só selecionará aquelas âncoras com um atributo title.

```
a[title] {  
    color: green;  
}
```

<p> Lorem ipsum dolor sit amet, consectetur. </p>

<p> Lorem ipsum dolor sit amet, consectetur elit, sed do eiusmod tempor. </p>

<p> Lorem ipsum dolor sit amet, consectetur adipisicing elit, por. </p>

<p> Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eu. </p>

CSS Avançado

10 - Seletor de atributos com valor - Seleciona apenas as tags que possuem um determinado atributo com um determinado valor. Esses valores podem ser expressões regulares.

```
a[href="https://www.amazon.com"] {/* exatamente igual ao valor */  
    color: #FEBD69;  
}  
  
a[href*="quix"] {/* a cadeia de caracteres existe em algum valor do atributo */  
    color: green;  
}  
  
a[href^="pt"] {/* o valor do atributo começa com a cadeia de caracteres */  
    color: red;  
}  
  
a[href$=".pdf"] {/* o valor do atributo termina com a cadeia de caracteres */  
    color: maroon;  
}
```


CSS Avançado

A linguagem CSS possui um grupo de seletores avançados que permitem estilizar elementos de formas distintas do que já foi visto antes. Eles são:

1 - Seletor de Pseudo-Elemento - É uma palavra-chave adicionada a um seletor que permite que você estilize uma parte específica do elemento selecionado.

2 - Seletor de Pseudo-Classe - É uma palavra-chave adicionada a seletores que especifica um estado especial do elemento selecionado. As pseudo-classes podem ser utilizadas para estilizar um elemento baseado em seu estado.

Pseudo-elementos

Sintaxe

Para **utilizar um pseudo-elemento** basta declarar o seletor seguido pelo pseudo-elemento desejado. Logo após, da mesma forma que é feito com qualquer regra CSS, podemos definir as propriedades a ele relacionadas.

Sintaxe de **declaração de um pseudo-elemento**:

```
[seletor]::pseudo-elemento { propriedade : valor; }
```

::first-line

O pseudo-elemento `::first-line` é utilizado quando desejamos customizar o conteúdo apresentado na primeira linha de um elemento. Assim, podemos estilizar esse fragmento de texto de forma independente, aplicando, por exemplo, um tamanho de fonte diferente.

```
p::first-line {  
    font-size: 32px;  
}
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Enim error commodi atque cumque  
consectetur laborum ab reprehenderit nulla voluptatem eaque aliquam est inventore saepe iusto  
sint, blanditiis at neque tempora?</p>
```

::first-letter

O pseudo-elemento ::first-letter nos permite selecionar a primeira letra do conteúdo de um elemento. Usamos esse recurso quando desejamos destacar essa letra alterando seu tamanho e sua cor, por exemplo.

```
p::first-letter {  
  color: red;  
  font-size: 32px;  
}
```

```
<p>  
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut  
labore et dolore magna aliqua.  
</p>
```

Combinando Multiplos Pseudo-elementos

Você pode combinar diversos pseudo-elementos no mesmo arquivo.

```
p::first-letter {  
  color: #ff0000;  
  font-size: xx-large;  
}  
  
p::first-line {  
  color: #0000ff;  
  font-variant: small-caps;  
}
```

<p>

Você pode combinar os pseudo-elementos `::first-letter` e `::first-line` para adicionar um efeito especial à primeira letra e à primeira linha de um texto!

</p>

::before e ::after

A declaração do pseudo-elemento `::before` e `::after` em um seletor cria um falso elemento que nos permite adicionar conteúdo antes e depois do conteúdo do elemento selecionado.

```
q::before {  
  content: "«";  
  color: blue;  
}
```

```
q::after {  
  content: "»";  
  color: red;  
}
```

`<q>Algumas citações</q>, ele disse, <q>são melhor do que nenhuma.</q>`

::selection

O pseudo-elemento **::selection** é utilizado para selecionar o conteúdo de um elemento e, a partir disso, customizar algumas propriedades dessa área, como a cor de fundo e do texto.

```
::selection {  
  color: gold;  
  background-color: red;  
}  
  
p::selection {  
  color: white;  
  background-color: blue;  
}
```

This text has special styles when you highlight it.
<p>Also try selecting text in this paragraph.</p>