0.1 Links HTML

Quando precisamos indicar que um trecho de texto se refere a um outro conteúdo, seja ele no mesmo documento ou em outro endereço, utilizamos a tag de âncora <a> . Existem dois diferentes usos para as âncoras. Um deles é a definição de links:

```
<p>Visite o site do <a href="http://www.google.co">
   Google</a></p>
```

Note que a âncora está demarcando apenas a palavra "Google"de todo o conteúdo do parágrafo exemplificado. Isso significa que, ao clicarmos com o cursor do mouse na palavra "Google", o navegador redirecionará o usuário para o site do Google, indicado no atributo href. Outro uso para a tag de âncora é a demarcaçãoo de destinos para links dentro do próprio documento, o que chamamos de **bookmark**. Por exemplo:

```
<p>Mais informacoes aqui <a href="#info">aqui</a></
   p>

<p>Conteudo da pagina...</p>

<h2 id="info">Informacoes: </h2>
<p>Bolsa de valores sobe 0,4%.</p>
```

De acordo com o exemplo acima, ao clicarmos sobre a palavra "aqui", demarcada com um link, o usuário será levado à porção da página onde o bookmark "info"é visível. Bookmark é o elemento que tem o atributo id.

0.2 Elementos Estruturais

Como foi visto, o conjunto de tags do HTML é bem vasto mas é também limitado. Não existe uma tag diferente para cada coisa do universe. Mais cedo ou mais cerdo você vai cair num cenárui onde não conseque achar a tag certa para aquele conteúdo. Nesse caso, existem duas tags que são utilizadas como "coringas", sendo elas: * * e **. São tags sem nenhum significado especial mas que podem servir para agrupar um certo conteúdo, tanto um bloco de página quanto um pedaço de texto. Por padrão, estão tags não têm estilo aglum. Contudo, podemo aplicar CSS customizado a esses elementos.

0.3 Seletores de ID e Filho no CSS

Como visto até o corrente momento do curso podemos selecionar elementos no CSS usando simplesmente o nome da tag:

```
1  p {
2    color: yellow;
3  }
```

Apesar de simples, é uma maneira muito limitada de selecionar. Às vezes não queremos pegar todos os parágrafos da página, mas apenas algum determinado. Existem maneiras mais avançadas de selecionarmos um ou mais elementos do HTML usando os seleteroes CSS.

0.3.1 Seletor de ID

É possível aplicar propriedades visuais a um elemento selecionado pelo valor de seu atributo id. Para isso, o seletor deve iniciar com o caractere "#"seguido do valor correspondente. Utilizando exemplo a definição da seção que demarca o conteúdo principal da página criado do curso (`https://github.com/heberthbraga/ifal/blob/master/intweb/app/css/index.css`)

```
1  #main {
2    position: relative;
3        ...
4  }
```

O seletor acima fará com que o elemento do nosso HTML que tem o atributo id com valor "main"tenha sua posição definida como *relative*, por exemplo. Note que não há nenhuma indicação para qual tag a propriedade será aplicada. Pode ser tanto uma quanto um <p> , até mesmo tags sem conteúdo como uma <img>, desde que essa tenha o atributo id com o valor "main".

Como o atributo id deve ter valor único no documento, o seletor deve aplicar suas propriedades declaradas somente àquele único elemento e, por cascata, a todos os seus elementos filhos.

0.3.2 Seletor Hierárquico

Podemos ainda utilizar um seletor hierárquico que permite aplicar estilos aos elementos filhos de um elemento pai:

```
1  #main p {
2    text-align: left;
3  }
```

No exemplo anterior, o elemento pai "main"selecionado pelo seu id. O estilo será aplicado apenas nos elementos p filhos do elemento com id=main.

0.4 Estilização com Classes no CSS

Para podermos estilizar os elementos que criamos, vamos precisar de uma forma de selecionarmos no CSS cada coisa. Já foi visto o seletor de tag e por ID. Tomando como exemplo o menu de navegação da página `https://github.com/heberthbraga/ifal/blob/master/intweb/app`, poderíamos fazer:

```
1  nav {
2    ...
3  }
```

Entretanto, imagem que a página venha a ter muitos NAV e queremos ser mais específicos. Poderíamos utilizar o ID como solução:

```
1  <nav id="menu"></nav>
2
3  CSS:
4
5  #menu {
6    ...
7  }
```

Porém, existe uma terceira form, com o usuo de *classes*. O código é semelhante de quando o usamos o atributo *id*, mas agora vamos usar o atributo **class** no HTML e o *ponto* no CSS:

```
1  <nav class="menu"></nav>
2
3  CSS:
4
5  .menu {
6    ...
7  }
```

Mas aí vocês se perguntam: Quando usar ID ou CLASS?

Ambos fariam seu trabalho nesse caso. Mas é bom lembrar que ids são mais fortes, devem ser únicos na página (sempre). Embora esse nosso menu seja único agora, imagine se, no gfuturo, quisermos ter o mesmo menu em outro ponto da página (outra seção). Nesse caso, o uso de classes irá facilitar o **reuso do código** e permitir maior **flexibilidade**.

Além disso, é possível que um elemento possua mais de uma classe ao mesmo tempo, aplicando estilos de várias regras do CSS ao mesmo tempo:

```
<nav class="menu menu-items"></nav>

CSS:

.menu {
  // Estilos especificos do menu em si.
}

.menu-items {
  // Estilos especificos para os items do menu.
}
```

0.5 Fluxo do Documento e Float

A propriedade **float** permite que tiremos um certo elemento do fluxo vertical do documento, o que faz com que o conteúdo abaixo dele flua ao seu redor. Tomando como exemplo `https://github.com/heberthbraga/ifal/tree/master/intweb/fluxo`, se vocês executarem index.html no navegador, perceberão que o conteúdo do parágrafo tentará fluir ao redor da nossa imagem com float, pois agora a nossa imagem parece existir fora do fluxo. Como exercício, tentem modificar o fluxo dessa imagem modificando seu estilo em *css/index.css*.

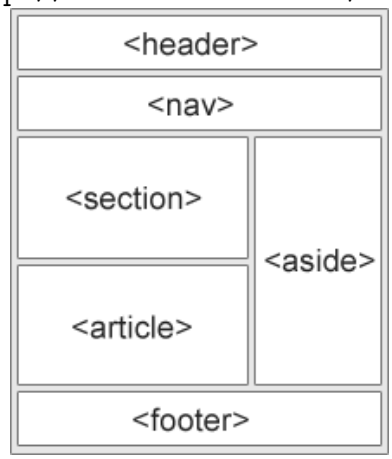
0.6 HTML semântico

O HTML5 oferece elementos semânticos que definem diferentes partes de uma página Web. Elementos semânticos são usados para "dizer"o que é cabeçalho, o que é barra de navegação, o que é conteúdo da página, o que é uma barra lateral, etc.

Sendo:

- <header> - Define um cabeçalho para um documento ou seção;
- <nav> - Define um container para links de navegação (como visto no exemplo da seção anterior);
- <section> - Define uma seção em um documento;
- <article> - Define uma seção independente;

Figura 1: Fonte: `http://www.w3schools.com/html/html_layout.asp`



- <aside> - Define uma área lateral para inserir conteúdo (pode ser um menu lateral);
- <footer> - Define um rodapé para um documento ou seção.

Antigamente era comum utilizarmos somente DIV para separar o conteúdo de uma página. Entretanto:

Qual a diferença entre colocar div e essas novas tags do HTML5?

Visualmente e funcionalmente, nenhuma. A única diferença é o nome da tag e o significado que elas carregam. E isto tem um impacto importante como será visto mais adiante.

Como foi dito, a função do HTML é fazer a marcação do conteúdo da página, representar sua estrutura da informação. Não é papel do HTML, por exemplo, cuidar da apresentação final e dos detalhes de design, pois isto é o papel do CSS. O HTML precisa ser claro e limpo, focado em marcar o conteúdo.

As novas tags do HTML5 trazem novos significados semânticos para usarmos em elementos HTML. Em vez de simplesmente agrupar os elementos do cabeçalho em um div genérico e sem significado, usamos uma tag header que carrega em si o significado de representar um cabeçalho, por exemplo.

Com isso, temos um HTML com estrutura baseada no significado de seu conteúdo, o que traz uma série de benefícios, como a facilidade de manutenção e compreensão do documento. Mas e se desabilitarmos o CSS e as regras visuais? Como distinguir esses conteúdos?

Um HTML semântico carrega significado independente da sua apresentação visual. Isso é particularmente importante quando o conteúdo for consumido por clientes não visuais. Há vários desses cenários. Um usuário cego

poderia usar um leitor de tela para ouvir sua página. Neste caso, a estrutura semântica do HTML é essencial para ele entender as partes do conteúdo. Outro ponto importante, robês de busca como os do Google também são leitores visuais de uma página. Sem um HTML semântico, o Google não consegue, por exemplo, saber que aquilo é menu e que deve seguir seus links.

Vocês podem usar como exemplo, o código do app da disciplina: `https://github.com/heberthbraga/ifal/blob/master/intweb/app`.

Referências Bibliográficas

[1] HTML5 e CSS3: Domine a web do futuro - Lucas Mazza, editora Casa do Código.

[2] Desenvolvimento Web com HTML, CSS e JavaScript - Caelum ensino e inovação, Curso WD-43.

[3] HTML Documentação: `https://devdocs.io/html/`

[4] W3C – CSS documentação: `https://www.w3.org/Style/CSS/#specs`