Estrutura básica

Após a introdução, passada no módulo anterior, gostaríamos de reforçar que, o HTML5 é a mais nova versão do HTML e oferece diversas novas funcionalidades, além de facilitar e potencializar as suas aplicações web, facilitando a interação com o usuário.

Além de ser uma versão mais nova do HTML, ele traz o conceito de melhorar a aplicação. Além de todas as funcionalidades e melhorias, o HTML 5 trouxe um grande recurso de armazenamento de dados, no lado cliente, que possibilita a navegação de um site offline e melhora o desempenho no servidor.

Para resumir, podemos dizer que o HTML5 nos revela um mundo de possibilidades, muito maior do que antigamente, melhorando a vida do desenvolvedor.

Para iniciarmos os estudos, vamos entender a estrutura básica para a criação de uma página em HTML5.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title of the document</title>

</head>

<body>

Content of the document...

</body>

</html>

O primeiro passo é informar ao navegador que estamos trabalhando com o HTML5, e esta á a função do **<!DOCTYPE html>**.

Em seguida, basta abrir a tag html, onde definimos a linguagem. A tag html engloba o restante da estrutura.

Dentro desta estrutura básica, temos o cabeçalho(head), onde definimos a codificação da página, que será UTF-8 e também o título.

Por último, temos o body que é responsável por englobar todo conteúdo que a página possuir.

Esta é uma estrutura bem simples. Falaremos de muitas outras tags e estruras para o desenvolvimento de uma página mais complexa. Passaremos, também, os novos conceitos.

No próximo módulo, começaremos a falar, mais detalhadamente, de cada elemento da nossa estrutura. Falaremos mais sobre os possíveis metadados que podemos utilizar, melhorando o posicionamento da página em relação aos motores de buscas, como o Google.

Trabalharemos bastante com semântica, para que os motores de busca possam indexar, da melhor forma, as nossas aplicações.

Falaremos sobre compatibilidade entre navegadores. Os navegadores mais atuais já reconhecem o HTML5, mas quando estamos desenvolvendo alguns sistemas, temos que levar em consideração as pessoas que ainda utilizam os navegadores antigos e desatualizados. Faremos com que estes navegadores, também, reconheçam o HTML 5, para que nossa aplicação não fique toda quebrada na tela do usuário.

# Compatibilidade

Como falamos, no módulo passado, a maioria dos navegadores modernos, já reconhecem o HTML 5, nativamente. Existem alguns navegadores, não atualizados, que ainda não o reconhece.

Quando um navegador não reconhece qualquer elemento que seja novo, ele trata como código inline. Podemos ensinar o navegador antigo a reconhecer estes elementos, que para ele, são novos, permitindo que o recurso do HTML 5 tenha seu papel garantido em qualquer navegador. Para isso, utilizaremos o CSS.

Para começarem a praticar, criem uma pasta para o projeto. Em seguida, criem um arquivo chamado **index.html**, dentro desta pasta. Este será nosso arquivo de exemplos. Desta forma, vocês poderão acompanhar a evolução dos módulos.

<style>

header, section, footer, nav, main, article, figure {

display: block;

}

</style>

O código acima, garante que os navegadores antigos reconheçam os novos elementos do HTML 5, como blocos e não mais in line, como anteriormente.

### Criar elementos personalizados

Podemos criar elementos personalizados, utilizando o javascript.

<script>

document.createElement("myTag");

</script>

Observem que utilizamos o método createElement para criar uma tag personalizada, onde podemos colocar o nome que quisermos. Após a criação da tag, podemos estilizar com o css. Vejam no exemplo abaixo:

<style>

header, section, footer, nav, main, article, figure {

display: block;

}

myTag{

display: block;

background-color: #ddd;

padding: 50px;

font-size: 30px;

}

</style>

Depois, basta que utilizemos, dentro da tag body, o nosso elemento criado e estilizado.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title of the document</title>

</head>

<body>

Content of the document...

<myTag></myTag>

</body>

</html>

Assim, já teremos o resultado esperado no browser. Façam as alterações acima, no arquivo index.html e depois acessem o navegador, para visualizar.

Está tudo muito lindo, tudo funcionando, mas os navegadores antigos não reconhecem esta forma de criação de novos elementos. Para isso, foi criado um arquivo de ativação, via Javascript. Vocês deverão adicionar, dentro da tag **head**, o código abaixo:

<!--[if lt IE 9]>

<script src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.js"></script>

<![endif]-->

Fazendo isso, estamos ativando o reconhecimento destes novos recursos, apenas, nos navegadores do Internet Explorer abaixo da versão 9. Isso acontece porque os novos navegadores reconhecem este código, como um comentário. Ele não executará a chamada do arquivo javascript. Os navegadores antigos não reconhecerão os comentários e incluirão o arquivo.

Nossa dica é que copiem este código, em algum local, ou decorem, para utilizarem em todos os projetos.

# Metadados

Metadados são um conjunto de informações a respeito da página e do conteúdo, nela publicada. Estas informações são utilizadas por motores de buscas ou robôs, que as utilizam para saber o conteúdo principal da página e depois exibe para os usuários que estão pesquisando.

Estes dados são invisíveis para o usuário, porque ficam dentro da tag **head**, da nossa estrutura.

Criaremos alguns dos principais metadados e depois comentaremos sobre alguns deles:

<meta charset="UTF-8">

<meta name="keywords" content="HTML, SON, School of net, HTML basic">

<meta name="description" content="Learning to work with HTML 5">

<meta name="author" content="Guilherme Ferreira">

<meta name="robots" content="index, follow">

<!-- <meta name="robots" content="noindex, nofollow"> -->

**charset**, já informamos que é o tipo de codificação da página.

**keywords**, são palavras-chave que, antigamente, mecanismos de busca dariam ênfase, durante as buscas dos usuários. Atualmente, elas não tem tanta força como antes. Isso ocorreu pelo fato de muitos usuários adicionarem palavras-chave em excesso. Muitas vezes, até palavras que não tinham relação com o setor de atuação do site. Isso fez com o que os motores de busca evoluíssem e não levassem em consideração as keywords.

Hoje em dia, os motores de busca priorizam a criação de conteúdos novos e palavras que utilizamos nos conteúdos.

**description** serve, também, para os buscadores poderem apresentar um breve resumo da página que foi pesquisada. Quando compartilhamos uma página, os recursos de compartilhamento buscarão esta informação. Os buscadores juntarão, principalmente, a tag **title** com o metadado **description**, para formar um breve resumo.

**author**, indicam o nome do autor do documento ou página.

**robots**, indicam se os motores de busca poderão indexar o seu site ou não e, se permitirão buscas dos links, presentes no site. O primeiro parâmetro é sobre indexação e o segundo é a permissão de exposição dos links do site. No exemplo, colocamos as duas opções possíveis e comentamos a menos utilizada.

Fora estes metadados, existem alguns que são específicos para cada rede social ou plataforma. Existem os metadados do Facebook, Twitter e até mesmo o Google. Mas, não é nosso objetivo. Vocês podem ler mais sobre os metadados, caso se interessem.

Semântica e definição do cabeçalho

Semântica é um dos conceitos mais importantes que o HTML 5 trouxe de novidade para o mundo web, que serve para padronizar as estruturas.

Podemos dizer que, semântica é um padrão de marcação que facilita os navegadores reconhecerem qual tipo de conteúdo está sendo exibido. Podemos definir: cabeçalho, rodapé, conteúdo principal, artigos, entre outros. Os elementos de marcação são:

* header
* section
* footer
* nav
* aside
* main
* article
* figure

Antigamente, as pessoas não deixavam de marcar os blocos dos sites, porém, marcavam com o nome que queriam e isso dificultava para os motores de buscas. Por este motivo o HTML 5 decidiu padronizar estas marcações. Se todos marcarem os blocos com os mesmos nomes, o mecanismo de busca consegue melhorar seu algoritmo e otimizar as buscas.

Vejam como foi a evolução dos códigos:

| **HTML 4** | **HTML 5** |
| --- | --- |
| \<div class="header"> | <header> |
| <div class="nav"> | <nav> |
| <div class="section"> | <section> |
| <div class="article"> | <article> |
| <div class="aside"> | <aside> |
| <div class="footer"> | <footer> |

Estamos exemplificando, levando em consideração que o desenvolvedor estivesse utilizando o mesmo nome que a tag atual, porém não é bem o que acontecia. Alguns colocavam nomes que não eram intuitivos e que somente ele conhecia.

Criando nosso header

<header>

<h1>Main title</h1>

<h2>Secondary title</h2>

<nav>

<ul class="menu">

<li><a href="index.html">Home</a></li>

<li><a href="about.html">About</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

</header>

Notem que utilizamos as tag **header** para englobar todo cabeçalho. Utilizamos **h1** para título principal, **h2** para título secundário e depois a tag **nav**, para criação do menu.

Façam, também, no arquivo index.html e depois acessem o navegador para conferirem o resultado.

# Organização do CSS e criação de Sections

Veremos como criar uma section no HTML 5. Esta tarefa não é complicada.

Primeiro, criem uma pasta, na raiz do projeto, chamada css. Dentro desta pasta, criem um arquivo chamado style.css e dentro deste arquivo, coloquem o conteúdo abaixo. Não esqueçam de remover as tags **style**, pois arquivos de css não precisam das mesmas.

<style>

header, section, footer, nav, main, article, figure {

display: block;

}

myTag{

display: block;

bachground-color: #ddd;

padding: 50px;

font-size: 30px;

}

</style>

Após termos adicionado o conteúdo acima, no arquivo style.css, apagaremos o mesmo conteúdo do arquivo index.html e chamaremos este arquivo externo. Para isso, vocês deverão adicionar o seguinte código:

<link rel="stylesheet" type="text/css" href="./css/style.css">

Desta forma, estamos mantendo o mesmo efeito. Estamos, somente, passando a responsabilidade de estilização para um arquivo externo. Assim, o código fica mais limpo e organizado. Outra vantagem é que, quando trabalhamos com a mesma folha de estilo para várias páginas, como centralizamos em apenas um arquivo, basta alterarmos em, apenas, um arquivo. Se fosse de outra forma, teríamos que alterar arquivo por arquivo.

Outro ponto que precisamos elucidar, antes de continuarmos com as sections, é sobre os scripts. Antigamente, precisávamos informar o tipo do script, dentro da abertura da tag. Com o HTML 5 não precisamos mais. Por padrão, o navegador já entenderá que se trata de um Javascript. Vejam o exemplo abaixo:

| **HTML 4** | **HTML 5** |
| --- | --- |
| <script type="text/javascript"></script> | <script></script> |

### Sections

A section é o menos específico, dos novos elementos do HTML 5.

Todos os outros servem para um determinado fim, porém, o section, serve para dividir os demais elementos. Em uma section podemos ter todos os outros elementos sendo utilizados, ou seja, podemos ter um header dentro de uma section, ter, também, um footer dentro de uma section.

Desta forma, podemos subdividir o projeto em sections, sem problema algum. O importante é mantermos o código, sempre, bem dividido e organizado.

A diferença da section e da div é que, a div serve para dividir qualquer elemento e a section foi criada para dividir os próprios elementos do HTML 5. Nada impede de utilizarem divs. Vocês podem utilizar melhor as sections, porque, semanticamente, estarão trabalhando da forma mais correta.

<section>

<article>

<header>

<h1>Title of a content</h1>

<p>Description of contents...</p>

</header>

... contents ...

<footer>

<small>

<a href="news.html">View more...</a>

</small>

</footer>

</article>

</section>

Notem que subdividimos nossa section. Utilizamos um article com header e footer para que comecem a entender que, cada tag tem seu papel específico, dentro do contexto em que se encontra.

Utilizamos a tag header para o cabeçalho do documento principal, mas isso não quer dizer que não possamos utilizar a tag header, novamente, para delimitar o cabeçalho de qualquer outra section.

Poderíamos, também, ter adicionado uma tag aside. Deixaremos para o próximo módulo