Dando continuidade no artigo anterior que foi explicado como é dividido a estrutura básica de um site com HTML5 e como organizar sua semântica. Vimos que podemos remover aquele monte de DIVS que eram criadas antigamente e usando a semântica, logo abaixo veja a estrutura das respectivas páginas. Lembrando que a semântica não ira excluir as DIVS, porém elas iram aparecer em casos específicos, veremos mais adiantes quais serão esses casos.



Observado o esquema a cima podemos visualizar que o HTML5 otimizou e muito os nossos códigos inserindo a semântica.

A partir desse momento iremos utilizar os conceitos do artigo anterior e a aplicar os novos conceitos nos próximos tópicos.

Vamos criar um arquivo index.html, dentro dele iremos inserir uma caixa principal uma DIV, que vai ficar responsável por armazenar todo o conteúdo, assim vamos conseguir posicionar todo o nosso conteúdo da forma que desejarmos, coloque o nome dela de principal, para isso inicialize uma DIV e digite class = nome da classe.

O nome que será atribuído a classe é principal, ele será chamado no arquivo .css para modificar a estrutura dessa div, fica assim:

Arquivo index.html

<!DOCTYPE html>

<html lang="pt-br">

<!-- Cabeçalho -->

<head>

<meta charset="utf-8">

<title> Curso de HTMl5 </title>

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

<meta name="description" content="Curso de HTMl5">

<link rel="stylesheet" href="css/estilo.css" >

</head>

<!-- Corpo do Documento -->

<body>

<!—Minha caixa principal -->

<div class="principal">

<!— Todo conteúdo do site aqui. Header, Section, Aside e etc. -->

</div>

</body>

</html>

Já com nossa DIV principal, vamos montar nossa estrutura semântica e realizar as alterações necessárias no arquivo .css.

Copie o código abaixo e modifique o corpo do documento index.html, salve e abra ele no navegador de internet.

<!-- Corpo do Documento -->

<body>

<!—Inicio da DIV principal-->

<div class="principal">

<!-- Cabeçalho da página-->

<header>Cabeçalho</header>

<!-- Menu do Site-->

<nav>Menu</nav>

<!-- Conteúdo Principal-->

<section>Conteúdo Principal</section>

<!--Conteúdo Relacionado, colocar ou não-->

<aside>Conteúdo Relacionado</aside>

<!-- Rodapé do site-->

<footer>Rodapé</footer>

<!—Fim da DIV principal-->

</div>

</body>

Ao abrir no navegador vera que não apareceu nada, isso é por que nós estamos somente com a estrutura da nossa página montada, agora vamos personalizar nossa página criando um arquivo .css dentro da pasta css.

Introdução ao CSS3

Para começarmos a editar o nossa pagina feita em HTML5 crie um arquivo chamado estilo com extensão .css dentro da pasta css.

Vou colocar uma lista das propriedades do CSS que iremos usar neste artigo e também o código css comentado para que entendam suas funcionalidades

Propriedades CSS

|  |  |  |
| --- | --- | --- |
| Propriedade | Descrição | Exemplo em CSS |
| margin | Definir de uma só vez o tamanho da margem para todos os lados do elemento. |  |
| margin-top  margin-right  margin-bottom  margin-left | Definindo o tamanho das margens externas do elemento, serve para informar qual distancia dos blocos e da extremidade do navegador, superior, inferior, direito e esquerdo | .principal{  margin-top: 10px;  } |
| \* | É um seletor universal. Usando o ‘\*’ você informa que está modificando os elementos de toda sua página, isso ira resetar toda página conforme o parâmetros que você definir. | \*{  margin: 1200px;  } |
| padding | definir de uma só vez a distância de espaçamento para todos os lados do elemento | div.pricipal{  padding:5px;  } |
| padding-top  padding-right  padding-bottom  padding-left | Define a distância ou espaçamento de cada parte interna do elemento, superior, inferior, direito e esquerdo | div.principal{  padding-top: 10px;  } |
| Font-family: | define a família de fontes iniciais que seram utilizadas, com ela você pode declarar uma fonte específica e uma genérica. | \*{  font-family: Arial, serif;  } |
| background-color: | Define uma cor para o fundo do elemento | div.principal{  background-color: #000;  } |
| width: | Define a largura do elemento | .principal{  Width: 1200px;  } |
| height | Define a altura do elemento | .principal{  heigth: 100px; |
| Float | Faz com que o nosso elemento flutue, left: esquerda e rigth: direita. Dessa forma podemos definir o nosso objeto do lado esquerdo ou direito. Propriedade nome o elemento não flutua. | .principal{  float: right;  } |
| Clear | Controla o posicionamento do elemento float, sendo 4 tipos:  none: não ira limpar nenhuma flutuação.  left: limpa a flutuação à esquerda: o elemento será posicionado abaixo da borda inferior do elemento flutuante à esquerda  right: limpa a flutuação à direita: o elemento será posicionado abaixo da borda inferior do elemento flutuante à direita  both: o elemento será posicionado logo abaixo da borda inferior de qualquer elemento flutuante, seja ele à direita ou à esquerda | Footer {  clear: both;  } |
|  |  |  |

Copie esse trecho para o estilo.css, definiremos algumas propriedades para inicializar a pagina

\*{

margin: 0px;

padding: 0px;

font-family: Arial;

}

Formatando DIV principal

Para chamar uma DIV que criamos existem duas formas, coloque [div.|nome da classe|] ou [.|nome da classe|]. Ex: div.principal ou .principal! Para atribuir as propriedades dessa DIV coloque tudo dentro do { “ propriedades aqui! ” }.

.principal{

background-color: #ccc;

width: 1200px;

margin: auto;

margin-top: 20px;

padding: 10px;

}

Observação: Quando usamos o padding ele aumenta o tamanho do elemento, isso causa um problema quando colocamos o padding dentro de outros elementos que estão juntos, mais logo veremos como usar o padding sem que altere o tamanho do elemento.

header{

background-color: #fff;

width: 1200px;

height: 200px;

}

nav{

background-color: #fff;

margin-top: 10px;

width: 1200px;

height: 50px;

}

Se olhar o desenho do nosso Layout vera que o section está do lado esquerdo e o aside do lado direto, para isso termos que fazer o section flutuar a esquerda da página e o aside flutuar a esquerda do section. O aside também flutua a esquerda, porque queremos que ele fique ao lado do elemento section.

Fique atento aos tamanhos que definimos para nossa DIV principal, se ela tem o tamanho de 1200px, vamos ter que definir o tamanho da section e aside de forma que uma caiba um do lado da outro, coloquei a section com 900px e o 290px, que chega bem perto do tamanho da DIV principal que é 1200px.

O nosso section ira flutuar para esquerda:

section{

background-color: #fff;

margin-top: 10px;

width: 900px;

height: 500px;

float: left;

}

O nosso aside ira flutuar a esquerda do elemento section, ficando ao lado:

aside{

background-color: #fff;

width: 290px;

margin-top: 10px;

height: 500px;

float: left;

margin-left: 10px;

margin-bottom: 10px;

}

Para começar a estilizar o footer primeiramente teremos que colocar o comando Clear, ele limpa o comando de flutuar anterior. Como o footer vem depois do section e do aside e eles dois são elementos fluentes, precisamos informar que após esses elementos terá outro elemento, que no nosso caso é o footer, senão colocarmos a propriedade ‘clear:both’ o navegador não ira entender onde queremos que esse elemento se posicione e ficará desconfigurado.

footer{

background-color: #fff;

width: 1200px;

height: 60px;

clear: both;

}

Código fonte completo

Index.html

<!DOCTYPE html>

<html lang="pt-br">

<!-- Cabeçalho -->

<head>

<meta charset="utf-8">

<title> Curso de HTMl5 </title>

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

<meta name="description" content="Curso de HTMl5">

<link rel="stylesheet" href="css/estilo.css" >

</head>

<!-- Corpo do Documento -->

<body>

<div class="principal">

<!-- Cabeçalho da página-->

<header>Cabeçalho</header>

<!-- Menu do Site-->

<nav>Menu</nav>

<!-- Conteúdo Principal-->

<section>Conteúdo Principal</section>

<!--Conteúdo Relacionado, colocar ou não-->

<aside>Conteúdo Relacionado</aside>

<!-- Rodapé do site-->

<footer>Rodapé</footer>

</div>

</body>

</html>

Estilo.css

\*{

margin: 0px;

padding: 0px;

font-family: Arial;

}

.principal{

background-color: #ccc;

width: 1200px;

margin: auto;

margin-top: 20px;

padding: 10px;

}

header{

background-color: #fff;

width: 1200px;

height: 200px;

}

nav{

background-color: #fff;

margin-top: 10px;

width: 1200px;

height: 50px;

}

section{

background-color: #fff;

margin-top: 10px;

width: 900px;

height: 500px;

float: left;

}

aside{

background-color: #fff;

width: 290px;

margin-top: 10px;

height: 500px;

float: left;

margin-left: 10px;

margin-bottom: 10px;

}

footer{

background-color: #fff;

width: 1200px;

height: 60px;

clear: both;

}

Leia e releia esse artigo quantas vezes precisar, volte e refaça os códigos acima. O importante é entender quando podemos usar certas propriedades do HTML e CSS.