

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BA – IFBA

PROGRAMAÇÃO WEB - APOSTILA CSS
PROFESSOR: ADDSON ARAÚJO DA COSTA

VALENÇA - BA

Sumário

1. INTRODUÇÃO.....	4
2. HTML.....	4
2.1 DIV.....	5
2.2 ID.....	5
2.3 Classe.....	5
2.4 Comentários.....	5
3. CSS.....	6
3.1 Seletores.....	6
→ Seletor Universal.....	6
→ Seletor por Tag.....	6
→ Seletor por Classe.....	6
→ Seletor por ID.....	7
3.2 Cores e fundo.....	7
→ Color.....	7
→ Background-color.....	8
→ Background-image.....	8
→ Background-repeat.....	8
→ Background-attachment.....	9
3.3 Fontes.....	9
→ Font-family.....	9
→ Font-style.....	9
→ Font-variant.....	9
→ Font-weight.....	10
→ Font-size.....	10
3.4 Tamanhos.....	10
3.5 Modelo de caixa.....	10
→ Margin.....	10
→ Border.....	11
→ Padding.....	11
→ Detalhes sobre Margin e padding.....	11
4. Google Fonts.....	12
5. Caixas flutuantes.....	12
→ Float.....	13
→ Clear.....	13
Exercício Proposto – Site Tableless.....	13

Lista de Figuras

Figura 1: Modelo de Caixa.....	10
Figura 2: Estilos de borda.....	11
Figura 3: Site Tableless.....	13

Lista de Tabelas

Tabela 1: Primeiros Seletores.....	6
Tabela 2: Cores e fundo.....	7
Tabela 3: Fontes.....	9
Tabela 4: Tamanhos.....	10
Tabela 5: Margens.....	10
Tabela 6: Bordas.....	11
Tabela 7: Margem, Espaçamentos e Bordas.....	11
Tabela 8: Elementos flutuantes.....	12

1. INTRODUÇÃO

Antes de iniciarmos o assunto CSS propriamente dito, vamos voltar a alguns dos elementos do HTML como o negrito, por exemplo, cuja tag é B de *Bold* (do inglês, negrito). Usar a tag faz com que o texto fique com um aspecto de letra mais espessa que caracteriza o negrito. Agora imagine que seu site depois de pronto, o cliente não quer mais essa aparência. O que fazer?

Claro, um dos caminhos é alterar todo o código fonte a procura das tags e removê-las, mas além de não ser muito prático, uma tag de abertura ou fechamento pode passar batida e comprometer todo o site. Uma outra solução nesse caso é usar o CSS.

Para utilizar o CSS, várias formas podem ser válidas, mas começaremos com um método exemplificado abaixo. Nesse método, é incluso uma tag chamada STYLE dentro da tag HEAD. É importante dizer que dentro da tag STYLE não pode ser incluso nenhum código HTML. Ao invés disso, deve conter apenas código CSS.

```
<html>
  <head>
    <style>
      b {
        font-weight: normal;
      }
    </style>
  </head>
  <body>
    <b> Teste </b>
  </body>
</html>
```

Exemplo 1: Utilização do CSS

Agora que sabemos como utilizar o CSS, vamos entender o que foi declarado dentro da tag STYLE. Primeiro, temos a letra B seguida de chaves. Chamaremos todo o código que estiver imediatamente antes das chaves por “seletor”, isto é, o critério utilizado para selecionar quais elementos devem ser modificados por este código CSS incluso dentro das chaves. No exemplo acima foi utilizado um seletor por tag, então não importa quantos negritos existam no código fonte, todos os negritos serão afetados, uma vez que foi colocado uma seleção pela tag B.

Dentro das chaves temos um “conjunto de regras” que devem ser seguidas separadas por ponto e vírgula. No exemplo acima foi utilizado apenas uma regra dentro das chaves, mas poderia ter várias outras. A regra do exemplo acima informa que o font-weight deve receber o valor normal, isto é, o peso da fonte deve ser o normal. Valores possíveis para essa regra são, por exemplo, normal, bold, bolder, lighter que significam, respectivamente, normal, negrito, negrito escuro, mais claro que o normal. Dependendo da fonte utilizada, o resultado é diferente uma vez que cada fonte tem um conjunto de espessuras que foram projetadas.

2. HTML

Alguns assuntos sobre HTML devem ser incluídos antes de começarmos nossos estudos sobre CSS. São eles a definição de ID e Classe de elementos. Vamos aproveitar e introduzir um novo elemento HTML, uma nova tag chamada DIV.

2.1 DIV

A DIV é uma caixa transparente que em nada modifica a estética da página, apenas serve para poder colocar outros elementos dentro dela como forma de agrupar e facilitar posicionamento destes elementos na página, uma vez que ao modificar a posição da DIV, todos os elementos internos a acompanham por estarem dentro dessa caixa invisível. Ela é muito útil principalmente por não afetar a aparência do site, assim, podemos deixar esta tarefa para o CSS.

2.2 ID

Todos os elementos no HTML podem receber IDs, isto é, um atributo do elemento que armazena um identificador único para este elemento. Este ID nunca deve se repetir no código HTML. Um exemplo de como definir o ID é o código do Exemplo 2, abaixo.

2.3 Classe

As classes, ao contrário dos IDs, são atributos podem e devem se repetir. Elas permitem agrupar elementos que devem ter mesma aparência e seu uso está diretamente ligado aos códigos CSS que aprenderemos a seguir. Um detalhe é que um elemento HTML pode pertencer a várias classes ao mesmo tempo. É uma forma de dizer que ele contém a aparência combinada de várias classes de elementos. Veja o código a seguir no Exemplo 2.

2.4 Comentários

Para fins didáticos, note que no Exemplo 2 foram introduzidos códigos em azul (comentários em códigos CSS) e em vermelho (comentários em códigos HTML). A sintaxe é diferente de acordo com a linguagem e vale a pena prestar atenção na sintaxe. O exemplo 2 está comentado abaixo e sua explicação pode ser acompanhada através dos comentários.

```
<html>
  <head>
    <style>
      .azul {           /* Aqui foi definido que todos os elementos que tiverem a classe "azul" */
        color: blue;    /* Terão a cor de fonte azul */
      }
      .grande {        /* Aqui foi definido que todos os elementos que tiverem a classe "grande" */
        font-size: 30px; /* O tamanho da fonte será 30 pixels */
      }
      .centralizado {  /* Aqui foi definido que todos os elementos que tiverem a classe "centralizado" */
        width: 100%;   /* Ocupará 100% da largura de onde ele está */
        text-align: center; /* E textos e figuras dentro dele, estarão centralizados */
      }
      #elemento1 {     /* Aqui foi definido que todos os elementos que tiverem o ID "elemento1" */
        font-style: italic; /* O estilo da fonte deverá ser itálico */
      }
    </style>
  </head>
  <body>
    <!-- Abaixo foi definido uma DIV cujo ID é elemento1 e tem três classes: azul, grande, centralizado -->
    <!-- O significado da aparência desse ID e classes, é definido no CSS acima -->
    <div id="elemento1" class="azul grande centralizado"> Teste </div>
  </body>
</html>
```

Exemplo 2: Ids, Classes, DIVs e comentários

3. CSS

A Sigla CSS refere-se a Cascading Style Sheets (do inglês, folhas de estilo em cascata) e é responsável por definir o estilo dos elementos HTML usados no site, isto é, definir a estética do site. Mais para frente entenderemos o porquê de se chamar folhas de estilo em cascata, mas note que ele está ligado unicamente a estilo, isto é, aparência.

3.1 Seletores

Seletores são usados como critério para selecionar elementos HTML. Existem várias opções e eles podem ser combinados de diversas formas para atender qualquer situação. Vamos esquecer por enquanto as formas de combiná-los e mostrar algumas opções de seletores.

Padrão	Nome	Exemplo CSS	Significado
*	Seletor universal	* { }	Seleciona todos os elementos
T	Seletor por tag	b { }	Seleciona todos que têm a tag T. Qualquer tag é válida.
.CLASS	Seletor por classe	.azul{ }	Seleciona todos os elementos que têm a classe CLASS. Não pode começar com número, ter acentos e/ou pontuação
#ID	Seletor por ID	#elemento1 { }	Seleciona todos os elementos que têm o ID

Tabela 1: Primeiros Seletores

O seletor por TAG pode ser visto no Exemplo 1, já os seletores por classe e ID, podem ser vistos no Exemplo 2 que, ao contrário do Exemplo 1, o seletor usado não inicia com uma letra, ao invés disso, inicia com um ponto (“.”) ou hashtag (“#”). Além destes, foi introduzido mais um tipo de seletor, o asterisco.

→ Seletor Universal

Ele é responsável por selecionar todos os elementos do site de uma só vez. Isto é muito útil, por exemplo, para quando se quer definir a fonte de todo o site. Dessa forma, de uma só vez, se define algo para todo o projeto. Para isso, basta colocar um asterisco no lugar do seletor.

→ Seletor por Tag

Com ele pode-se selecionar qualquer elemento através de sua Tag, isto é muito útil por permitir ajustes como, por exemplo, para que todos os parágrafos tenham o recuo na primeira linha, para definir o nível de negrito em todos os negritos, para alterar a aparência de todas as âncoras (Tag A usada para indicar links) para que não fiquem com suas aparências toscas e padrão, estilizar todos os botões, etc. Para usar o seletor por tag, basta digitar o nome da tag desejada no lugar do seletor, por exemplo, body, b, i, table, etc.

→ Seletor por Classe

Ao iniciar o seletor com o caracter de ponto final (“.”) estamos informando que desejamos buscar elementos que contenham aquela classe. Note que um elemento pode conter várias classes como observado no Exemplo 2. Assim, a idéia é definir classes mais genéricas e combiná-las ao invés de definir uma classe específica para um elemento.

→ Seletor por ID

Por último, quando iniciamos o seletor com hashtag, estamos informando que desejamos buscar um elemento com base em seu ID. Isto é útil no caso contrário ao seletor por classes. Aqui é usado quando desejamos definir algo específico que só existe para aquele elemento. Nesse caso, se atribui um ID ao elemento e definimos as regras CSS específicas.

3.2 Cores e fundo

Antes de entrarmos nas regras CSS que permitem definir cores e fundos, vamos entender como o computador faz para definir as cores na tela. É preciso saber que dentro do olho humano existem conjuntos de células capazes de reconhecer três cores, são elas, vermelho, verde e azul que em inglês são, respectivamente, *red*, *green* e *blue*, cuja sigla é *RGB*.

A partir de combinações destas 3 cores do RGB em diferentes intensidades, são formadas todas as cores visíveis ao ser humano, assim, em informática é usado a combinação destas cores como forma de definição de cor, ao contrário de outras possibilidades como, por exemplo, nos meios artísticos muitas vezes se usam como cores primárias a união de vermelho, azul e amarelo. Em impressoras usa-se em alguns modelos a união entre Ciano, Magenta, Amarelo e Preto (CMYK).

Regra CSS	Nome	Valores possíveis
color	Cor do texto	Nome da cor em inglês rgb (255,0,0) rgba (255,0,0,0.5) #FF0000
background-color	Cor de background	Nome da cor em inglês rgb (255,0,0) rgba (255,0,0,0.5) #FF0000 transparent
background-image	Imagem de background	url("caminho/img.jpg")
background-repeat	Repetição do background	repeat repeat-x repeat-y no-repeat
background-attachment	Anexação de background	scroll fixed

Tabela 2: Cores e fundo

→ Color

A primeira regra da Tabela 2 se chama color, ela é responsável pela cor da fonte e, entre os valores possíveis, note que a primeira opção é colocar o nome da cor em inglês, para muitas cores isto funcionará como, por exemplo, black, blue, red, purple, white, etc.

A segunda opção é colocar rgb seguido de parênteses com três números entre vírgulas, o primeiro valor refere-se a quantidade de vermelho, o segundo refere-se a quantidade de verde e o último, a quantidade de azul pois esta é a ordem na sigla RGB. É importante ressaltar que o valor para cada cor a ser usado deve ser um número inteiro entre 0 e 255, não sendo aceito outros valores.

A terceira opção se chama RGBA, isto é, idêntico ao modelo RGB, porém adiciona-se mais um valor que deve ser informado por último. Este último valor refere-se ao canal Alpha e é quem define a opacidade da cor caso não se deseje uma cor opaca como, por exemplo, deseja-se atribuir cor a um elemento, mas continuar mostrando um efeito de fundo que existe na página. Para o canal Alpha

deve-se informar um valor entre 0 e 1, para informar metade da opacidade, por exemplo, deve ser atribuído o valor “0.5” note que usamos o ponto, e não a vírgula, nesse caso.

Antes de entendermos a quarta opção, é preciso entender a diferença entre sistema decimal e hexadecimal. No sistema decimal, cada dígito assume os valores de 0 a 9, o que totaliza dez opções de números em cada dígito dando a esse sistema o nome de decimal. Já no sistema hexadecimal cada dígito pode assumir valores de 0 a 9 e também as letras de A a F que significam, respectivamente, 10 a 15.

A quarta opção é ainda uma definição com base em RGB, mas a forma de escrever é ligeiramente diferente. O símbolo de hashtag aqui quando usado para definir cores significa que os valores para o RGB serão informados usando números hexadecimais. Nesse caso, os primeiros dois dígitos significam a quantidade de vermelho, os dois dígitos do meio significam a quantidade de verde e os últimos significam a quantidade de azul.

Assim, o número 255 em decimal, quando convertido para hexadecimal equivale a FF, então note que a cor #ff05f4 significa que terá “ff” de vermelho (que em decimal equivale a 255), “05” de verde (em decimal equivale a 5) e “f4” de azul (que em decimal equivale a 244). Ou seja, as cores #ff05f4 e rgb(255, 5, 244), são equivalentes usando formas diferentes de escrita.

A preferência entre os programadores costuma ser pelo sistema hexadecimal uma vez que é mais fácil achar paletas de cor usando esse sistema e requer menos texto a ser escrito.

→ **Background-color**

Para cor de fundo, pode-se usar o mesmo modo de valores que o usado para definir cor de fonte, porém pode-se utilizar também a palavra “transparent” que é a cor de fundo padrão de todos os elementos.

→ **Background-image**

Esta regra permite usar uma imagem de fundo e para isto deve-se especificar um caminho onde será encontrada a imagem. Este caminho pode ser um endereço na internet e nesse caso o endereço deve ser completo especificando a partir do “http://” ou “https://” que é o protocolo usado para obter a imagem no servidor.

Caso a imagem esteja copiada na mesma pasta em que encontra-se seu código HTML, basta informar o nome da imagem seguido de sua extensão. Um erro comum aqui é que o windows oculta a extensão, mas ela faz parte do nome do arquivo, então uma imagem jpeg, por exemplo, pode estar escrito no nome do arquivo como “professor.jpg”, “professor.jpeg”, “professor.JPG”, “professor.JPEG”, mas o windows mostrará apenas o nome da imagem como sendo professor, cabendo o programador chutar qual o nome do arquivo acha ser o correto ou verificar nas propriedades do arquivo ao clicar com o botão direito do mouse e verificando os detalhes do arquivo.

→ **Background-repeat**

Esta regra é útil quando se quer colocar uma imagem de fundo e permitir que ela se repita na tela. Uma possível aplicação é ao colocar uma textura como imagem de fundo. Com esta regra, pode-se informar quatro opções de valores.

O primeiro, repeat, permite que a imagem de fundo se repita horizontal e verticalmente. O segundo, repeat-x, permite repetições apenas no eixo horizontal. O terceiro, repeat-y, permite repetições apenas no eixo vertical. O último, no-repeat, não permite repetições.

→ **Background-attachment**

Esta regra é responsável por definir como se dá a posição do background quando a página sofre o efeito de scroll, ou seja, o efeito de descer a tela pois existe mais conteúdo do que o que cabe sendo exibido.

Ao utilizar essa regra com o valor “fixed”, o background não rolará na tela mesmo que o conteúdo seja rotacionado pelo scroll. Já usando o valor “scroll” informará que o fundo deve também ser rotacionado junto ao conteúdo.

3.3 Fontes

Para fontes, temos um conjunto de regras CSS que podem ser vistas na Tabela 3, abaixo.

Regra CSS	Nome	Valores possíveis
font-family	Família(s) de fontes	Nome da fonte
font-style	Estilo utilizado na fonte	normal italic
font-variant	Variação entre letra maiúsculas e minúsculas	normal small-caps
font-weight	Peso da fonte	normal bold bolder lighter
font-size	Tamanho da fonte	Valor do tamanho

Tabela 3: Fontes

→ **Font-family**

Com a regra font-family, se define qual o nome da fonte usar. Apesar de ser semelhante ao que encontramos em editores de texto quando fazemos um relatório ou conjunto de slides, em web não podemos usar nomes de fontes disponíveis em nosso computador, uma vez que esse código será executado por qualquer computador na internet. Mais para frente, será mostrado como incluir uma fonte em nosso site de forma a ela estar disponível em qualquer computador que abrir nosso site.

→ **Font-style**

Na regra font-style podemos definir se queremos a fonte normal ou em itálico. Note que para definir negrito é outra regra CSS, isto porque a fonte pode estar em itálico e negrito ao mesmo tempo.

→ **Font-variant**

Na regra font-variant podemos definir como queremos a diferenciação entre maiúsculas e minúsculas que pode ser da forma normal, ou seja, existe uma versão de letra maiúscula e uma versão para letra minúscula. Já se escolhermos a font-variant da forma small-caps, ao invés de usar a letra minúscula, é usado a letra maiúscula em tamanho menor, criando uma escrita semelhante a letra de forma.

→ Font-weight

Com a regra font-weight podemos definir se queremos a fonte normal, bold, bolder ou lighter que significam, normal, negrito, negrito escuro e mais leve que o normal, respectivamente.

→ Font-size

Esta regra serve para definir o tamanho da fonte. Esse valor é normalmente em pixels e representa a altura da fonte do seu ponto mais baixo até a sua parte mais alta. Recebe um valor numérico.

3.4 Tamanhos

Todos os elementos podem ter seus tamanhos definidos. Os valores podem ser definidos para o tamanho padrão, o máximo e mínimo. Isso faz diferença quando a página é redimensionada. O navegador tentará manter sempre o elemento do tamanho padrão, mas caso não caiba, ele irá reduzir para caber.

Regra CSS	Nome	Valores possíveis
width	Largura padrão	Valor numérico
min-width	Largura mínima	
max-width	Largura máxima	
height	Altura	
min-height	Altura mínima	
max-height	Altura máxima	

Tabela 4: Tamanhos

3.5 Modelo de caixa

Para todos os elementos no html, podemos imaginá-los como em um modelo como se existissem várias caixas uma dentro da outra como na Figura 1, ao lado.

→ Margin

A região em amarelo, chamada de margem, é uma área invisível que separa a borda do elemento de outros elementos ao redor. Imagine, por exemplo, que você criou a borda em uma caixa (DIV), mas não gostaria que essa borda encoste em outros elementos. Para isso, você deverá usar as regras referentes a margin em CSS.

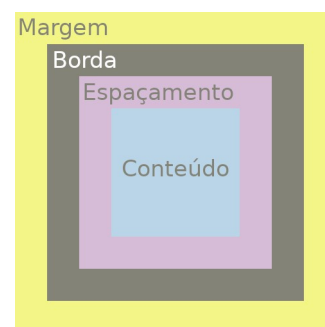


Figura 1: Modelo de Caixa

Regra CSS	Nome	Valor
margin-left	Margem esquerda	valor numérico
margin-right	Margem direita	
margin-top	Margem acima	
margin-bottom	Margem abaixo	
margin	Margem em todas as direções	

Tabela 5: Margens

→ Border

Enquanto a área de margem é uma área invisível, isto é, transparente, a área de borda pode ser visível caso você escolha uma borda para o elemento e pode assumir qualquer padrão de desenho ou cor.

Regra CSS	Nome	Valor
border-width	Largura da borda	valor numérico
border-color	Cor da borda	uma cor (rgb, rgba, hexadecimal, ...)
border-style	Estilo da borda	Valores possíveis: none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset

Tabela 6: Bordas

Para borda, podemos definir sua largura, cor e estilo. Os estilos podem melhor ser entendidos observando a Figura 2, ao lado.

Para as definições de borda foram explicados apenas três regras, mas a realidade é que existem algumas muitas que encontram-se listadas abaixo. Note que são variações destas três, porém para aplicar a borda apenas de algum dos lados e seu uso é idêntico as regras acima. São elas:

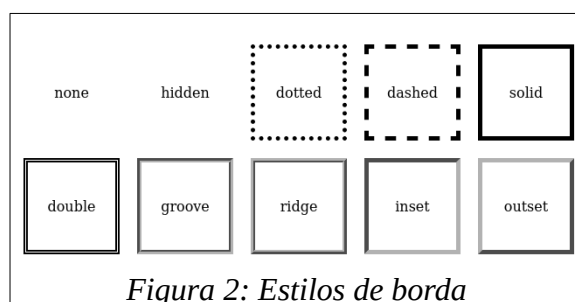


Figura 2: Estilos de borda

- border-left-width, border-right-width, border-top-width, border-bottom-width
- border-left-color, border-right-color, border-top-color, border-bottom-color
- border-left-style, border-right-style, border-top-style, border-bottom-style

→ Padding

Quando definimos uma cor de fundo ou imagem de fundo para um elemento, toda a área dentro da borda será afetada, ou seja, as áreas em roxo e azul na Figura 1. A diferença é que a área em roxo serve como um espaçador para que o conteúdo não toque na borda. Este é o padding (espaçamento).

Regra CSS	Nome	Valores possíveis
padding-left	Espaçamento esquerdo	Valor numérico
padding-right	Espaçamento direito	
padding-top	Espaçamento acima	
padding-bottom	Espaçamento abaixo	
padding	Espaçamento em todas as direções	

Tabela 7: Margem, Espaçamentos e Bordas

→ Detalhes sobre Margin e padding

Uma curiosidade é que ao usar a regra margin ou padding informando apenas um valor numérico, este valor é aplicado em todas as direções, como de se esperar.

Porém ao fornecer dois valores separados por um espaço em branco, o primeiro valor é aplicado para superior e inferior, enquanto o segundo valor é aplicado para esquerda e para direita.

Já se fornecermos quatro valores para margem ou espaçamento, o primeiro valor é aplicado para cima, o segundo para a direita, o terceiro para baixo e o quarto valor para a esquerda.

Uma dica para memorizar é que começa de cima e depois vai em sentido horário.

4. Google Fonts

Acima, no tópico sobre a regra font-family foi dito que para usar uma fonte, ela precisa estar disponível no computador de todos os usuários que abrirem nosso site. Não adiantando ter a fonte em nosso próprio computador uma vez que o código será executado em outra máquina. Bom, chamamos os computadores que forem executar nosso site de computadores cliente, mas como garantir que no computador do cliente terá a fonte que desejamos?

A resposta é simples, precisamos apenas linkar um arquivo de fonte disponível online, usando um link seguro, rápido e confiável. Assim, caso o computador do cliente não tenha a fonte que queremos, ele terá um link onde pode encontrá-la e fará todo o processo automaticamente.

Nesse sentido, surgiu um projeto da Google chamado Google Fonts <fonts.google.com>, onde pode-se escolher a fonte que se deseja vendo já sua aparência. Dentre as fontes, pode-se escolher os estilos entre os disponíveis para cada fonte. Os estilos são a espessura do traço e está diretamente relacionado ao peso da fonte que se deseja usar, isto é, intensidade para o que seria normal, bold (negrito), bolder (negrito escuro), etc. Os estilos de fonte são específicos para a espessura de traço e também para o estilo normal ou itálico.

Após selecionar as fontes desejadas e seus respectivos estilos, aparece um carrinho onde o serviço do google fornece uma tag do tipo LINK que deve ser copiada e colada na tag HEAD de seu site. Ainda no Google Fonts ele mostra um exemplo de como usar a fonte, fornecendo a regra font family. O serviço é gratuito e pensado para ser utilizado por desenvolvedores.

5. Caixas flutuantes

A partir deste momento, não usaremos mais tabelas para organizar o nosso site. Tabelas devem ser usadas apenas em pontos onde o conteúdo exige uma tabela como, por exemplo, as tabelas presentes nesta apostila. Nunca como método para organizar o site.

O nome desse conceito usado em WEB é que as páginas devem ser Tableless, isto é, sem tabelas e o motivo é a falta de controle na forma de exibir tabelas de acordo com o navegador que o cliente estará executando. Um site feito com tabelas vai diferir a estética de acordo com o navegador usado pelo cliente.

Regra CSS	Nome	Valores possíveis
float	Flutuação do elemento	left right
clear	Limpar flutuação	left right both

Tabela 8: Elementos flutuantes

Para que isto seja possível, vamos pensar na tag DIV apresentada no início da apostila. Ela representa uma caixa vazia e por padrão, quando colocamos duas caixas elas ficam empilhadas, isto é, uma abaixo da outra, mas isto pode ser modificado no CSS.

→ Float

O comando necessário para esta tarefa de colocar os elementos lado a lado é o Float que significa flutuar. Ele faz com que as caixas tentem subir (flutuar) sempre que possível (caso tenha espaço). Caso as caixas tentem subir, elas irão competir por espaço e, conseqüentemente, ficarão uma ao lado da outra.

Para cada elemento, podemos escolher se desejamos que ele flutue para a direita ou esquerda. Assim escolhendo a posição de todos, uma vez que os elementos inseridos primeiro no código irão flutuar primeiro.

→ Clear

Enquanto a regra Float faz os elementos tentarem subir, a regra clear cria um bloqueio, obrigando todos os elementos restantes a não ultrapassarem o elemento que estabeleceu a barreira.

Essa barreira criada pelo comando clear pode ser estabelecida apenas para os elementos que tentarem flutuar pela direita, apenas para elementos que tentarem flutuar pela esquerda ou ambos os lados, criando uma verdadeira divisão onde nenhum elemento pode ultrapassar flutuando.

O mais comum é usar essa regra com o valor both que significa ambos os lados pois é mais simples entender o fluxo de elementos na página desta forma.

Exercício Proposto – Site Tableless

Com os conhecimentos adquiridos até aqui você deve ser capaz de criar um site como o proposto abaixo. Isto é, com um topo, rodapé, menu e área de conteúdo corretamente alinhados como na Figura 3, abaixo. Deve também ter cores de sua escolha, figuras, textos e também ter uma fonte usada do Google Fonts. Todo o conteúdo deve ser de acordo com seu gosto, desde que contenha material adequado para menores de idade. Aproveite para fazer algo bonito e elegante. Observe as dicas dentro do exercício.



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BA - IFBA

Exercício II - Crie um site semelhante a esta figura

Dicas para desenvolver este exercício:

Principal

Mapa

Links

1 - Crie uma classe que manda elementos flutuarem a esquerda

2 - Crie uma classe que manda criar uma barreira impedindo flutuacao

3 - Crie mais uma classe que associa uma cor de fundo ao elemento

4 - Crie 4 Divs (topo, menu, conteudo, rodapé)

5 - Associe a elas as classes que forem necessárias e atribua IDs diferentes

6 - Use os IDS criados para definir margem, padding e tamanho das divs criadas

6 - Quando as quatro divs anteriores estiverem nos locais e tamanhos corretos, Coloque novas divs dentro das que você já criou e repita o processo.

OBS: Tamanhos, conteúdos são personalizáveis desde que o layout seja mantido e tenha elementos nos locais preenchidos por esse layout, observe as distâncias.

Redes Sociais:


MAPA DO SITE
Principal
Mapa
Links

Figura 3: Site Tableless