Prof. MSc. Marcos Alexandruk

E-mail: alexandruk@uni9.pro.br

https://github.com/alexandruk/desenvolvimentoparainternet

Introdução para Programação à Internet

aula 1

O que é e para que serve a HTML 5

HTML (Hyper Text Markup Language) é uma Linguagem de Marcação de Hipertexto utilizada para criação de páginas da **www** na Internet. Essa linguagem é o que o seu navegador (browser, em inglês) precisa para exibir as páginas.

Desde a invenção da web por Tim Berners-Lee, a **HTML** evoluiu e, atualmente, está na **versão 5**.

Com HTML você define a estrutura de uma página, estabelecendo o que é título, texto, lista, subtítulo, local das imagens, link etc.

Para definir a estrutura da página, a HTML faz uso de **tags**, que são marcações de formatação. Aquilo que você vê quando abre uma página na internet é a interpretação que seu navegador faz do HTML, ou seja, o navegador interpreta todas as marcações, que são sinalizadas por meio de tags, que envolvem o texto, transformando-as em documento capaz de ser lido e entendido pelo usuário. A essa interpretação das marcações chamamos de **renderização**.

Uma das principais características da HTML é a facilidade de manipulação de textos e objetos como figuras, sons, fotos, animações e que tem por objetivo não apenas criar textos, mas hipertextos. Esses textos caracterizam-se por serem rápidos e pequenos, facilitando o acesso dos usuários da web.

HTML

- HTML é uma linguagem para descrever páginas Web.
- HTML é a sigla de HyperText Markup Language.
- HTML não é uma linguagem de programação e sim uma linguagem de marcação.
- A linguagem de marcação apresenta um conjunto de tags.
- HTML utiliza as tags de marcação para descrever as páginas da web.

Elementos HTML

Elemento HTML é a menor unidade de marcação. No W3C (World Wide Web Consortium) encontram-se todas as recomendações sobre a utilização dos elementos HTML, bem como os que estão em desuso.

Cada elemento tem uma especificação e o seu uso deve ser de acordo com a finalidade do elemento.

Temos, por exemplo, elementos destinados à marcação de títulos, elementos que têm a finalidade de marcar parágrafos e, assim por diante.

Tags HTML

Os comandos HTML são chamados de tags HTML.

Os elementos de marcação dos diferentes conteúdos de um documento são representados por tags.

Assim como em outras linguagens, os comandos têm uma sintaxe própria, e seguem algumas regras:

As tags HTML são palavras que estão entre o sinal de "menor que" (<) e o sinal de "maior que" (>), por exemplo: ** **.

As tags HTML normalmente são utilizadas em pares, por exemplo: ** ** e ** **.

A primeira tag do par é o início e a segunda é a finalização.

O início e o final das tags também são chamados, respectivamente, de **tags de abertura** e **tags de fechamento**.

O fechamento de uma tag é indicado por uma "/": ** **.

Algumas tags não necessitam de fechamento, como é o caso da tag **
**.

Página HTML (aula01a.html)

```
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body>
    <h1>Cabeçalho da seção</h1>
    <p>Parágrafo</p>
    <p>Quedra <br> de linha</p>
    <ul>
      <li>Primeiro elemento da lista</li>
      <li>Segundo elemento da lista</li>
    </ul>
  </body>
</html>
```

aula01a.html

Página HTML (aula01b.html)

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8" />
    <title>Título da página</title>
  </head>
  <body>
    <h1>Cabeçalho da seção</h1>
    <p>Páragrafo</p>
    <p>Quedra <br /> de linha</p>
    <ul>
      <li>Primeiro elemento da lista</li>
      <li>Segundo elemento da lista</li>
    </ul>
  </body>
</html>
```

aula01b.html

Linhas de comentários

Linhas de comentários, têm a finalidade de facilitar o entendimento e manutenção de código.

Os comentários são pequenos trechos de textos que você pode utilizar para explicar o seu código.

Para inserir um comentário em HTML, utilizamos o marcador **<!--** para iniciar o comentário e **-->** para fechar o comentário.

Todo texto que estiver entre as marcações de comentário são ignorados pelo navegador, ou seja, esse texto não será renderizado.

Veja exemplo a seguir:

```
<!-- Isto é um comentário -->
<p>Texto de um parágrafo.</p>
```

aula01c.html

HTML e XHTML

HTML não faz diferença entre maiúsculas e minúsculas (não é "case sensitive"). Portanto, a notação <title> é equivalente a <TITLE> ou <TiTlE>, mas XHTML é case sensitive, isto é, faz diferença entre maiúsculas e minúsculas, portanto, **habitue-se desde já a escrever todas as tags em letras minúsculas**, assim você poderá utilizar tanto HTML quanto XHTML.

XHTML é a sigla em inglês para **eXtensible Hyper Text Markup Language**, que em português significa Linguagem Extensível para Marcação de Hipertexto. Trata-se de uma linguagem que utiliza regras de sintaxe muito mais rígidas que as regras para a HTML.

Atributos

Os atributos são utilizados para fornecer informações adicionais para um elemento HTML e são declarados dentro da tag de abertura do elemento.

Observe a declaração:

```
<h1 style="color:blue">Cabeçalho azul</h1>
```

A sintaxe para se escrever um atributo é: nome do atributo seguido por um valor, que deve ser colocado entre aspas (simples ou duplas) e separado por um sinal de igual.

O texto em negrito acima informa ao navegador que as letras do cabeçalho deverão ser azul (blue).

O que é e para que serve o CSS

Quando o World Wide Web Consortium (W3C) lançou a versão 4.0 da HTML, incorporou a ela as CSS (Cascading Style Sheets — Folhas de Estilos em Cascata), que têm a finalidade de dar uma forma de apresentação ao conteúdo do documento HTML. Passou a ser possível, portanto, **separar o conteúdo do documento de sua formatação**.

Antes de existirem as folhas de estilo, recorria-se a elementos e atributos da própria HTML para isso. Por exemplo, o elemento <font> definia a fonte, o tamanho e a cor do texto.

Os elementos e atributos específicos de formatação foram acrescentados gradativamente à especificação original da HTML pelas empresas criadoras de browsers.

As CSS oferecem uma maneira de descrever como o conteúdo deve ser apresentado, isto é, formatam o conteúdo, e controlam a apresentação do HTML.

Benefícios do uso de CSS

CSS é uma revolução no mundo do web design. Os benefícios do uso de CSS incluem:

- Controle do layout de vários documentos a partir de uma simples folha de estilos.
- Maior precisão no controle do layout.
- Aplicação de diferentes layouts para servir diferentes mídias (tela, impressora etc.).
- Emprego de variadas, sofisticadas e avançadas técnicas de desenvolvimento.

Como o CSS funciona

Suponha que desejemos uma cor de fundo verde para a página web:

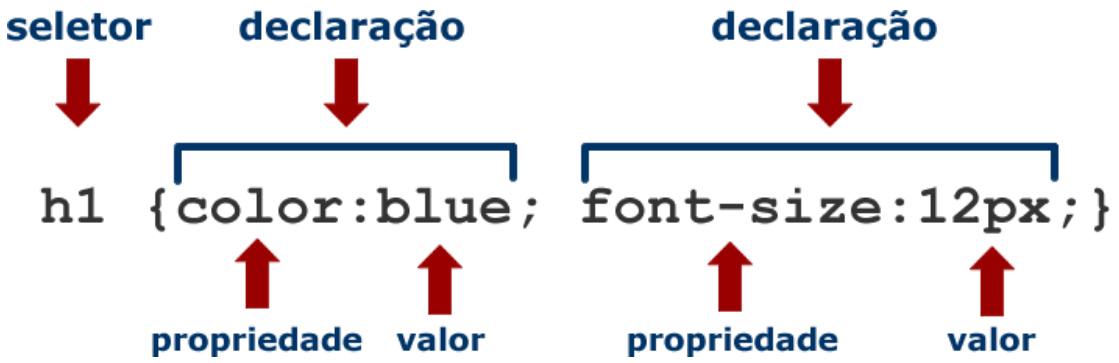
Usando HTML, podemos fazer assim:

```
<body bgcolor="#00ff00">
```

Com CSS, o mesmo resultado será obtido assim:

```
body {background-color: #00ff00;}
```

A sintaxe para CSS tem duas partes principais: um seletor e uma ou mais declarações.



aula01e.html e aula01e.css

O que é para que serve o JavaScript

JavaScript é uma linguagem de programação criada pela Netscape em 1995, que a princípio se chamava LiveScript, para atender, principalmente, as seguintes necessidades:

- Validação de formulários no lado cliente (no navegador).
- Interação do usuário com a página.

Com JavaScript é possível incluir funções e aplicações online básicas em páginas da Web. O código JavaScript, pode ser incluído em uma página da Web juntamente ao código HTML.

Com JavaScript podemos criar efeitos especiais nas páginas e definir interatividade com o usuário. O navegador do cliente é o encarregado de interpretar as instruções JavaScript e executá-las.

JavaScript é uma linguagem de programação bastante simples e pensada para fazer as coisas com rapidez e leveza.

Pessoas que não tenham experiência prévia em programação poderão aprender esta linguagem com facilidade e utilizá-la com somente um pouco de prática.

aula01f.html

O que é o W3C

O W3C (World Wide Web Consortium), criado em outubro de 1994, é um consórcio internacional formado por empresas, instituições, pesquisadores, desenvolvedores e público em geral, com a finalidade de desenvolver a web a seu potencial máximo, criando normas e especificações que se aplicam aos mais diversos segmentos e setores da web, desde tecnologia e softwares até fabricantes e fornecedores.

O W3C tem como missão conduzir a World Wide Web para que atinja todo seu potencial, desenvolvendo protocolos e diretrizes que garantam seu crescimento a longo prazo.

No site **https://www.w3c.br**, você encontra um vasto material sobre especificações, padrões e normas, bem como guias de referências e tutoriais para o desenvolvimento e criação de documentos web.

Prof. MSc. Marcos Alexandruk

E-mail: alexandruk@uni9.pro.br

https://github.com/alexandruk/desenvolvimentoparainternet

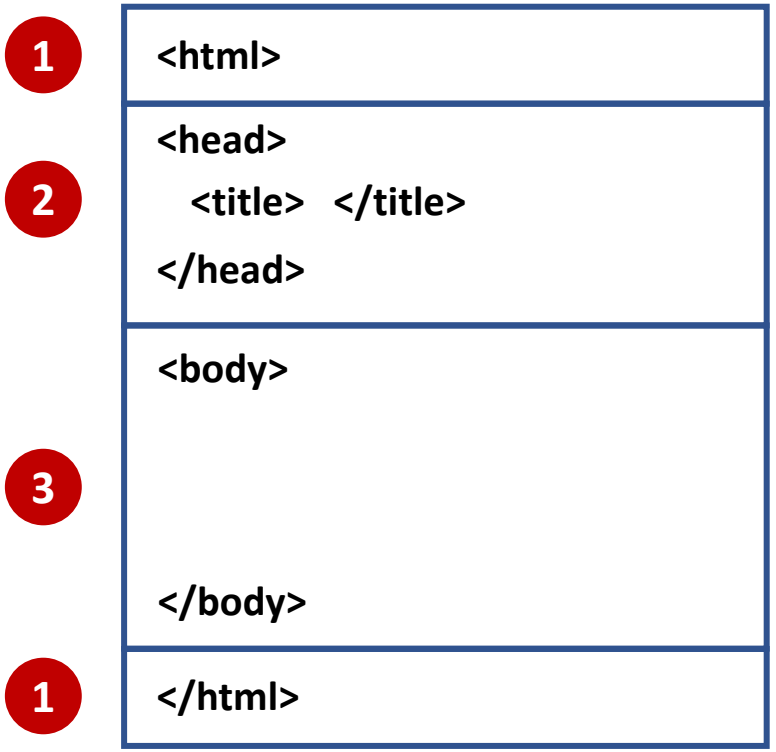
Introdução para Programação à Internet

aula 2

Estrutura de uma página HTML

Todo documento HTML, de acordo com os padrões do W3C, tem sua marcação mínima estruturada em três partes distintas:

1. Seção no documento para declarar a identificação como sendo um documento HTML.
2. Seção head definida como cabeça do documento HTML.
3. Seção body definida como corpo do documento HTML.



Estrutura de uma página HTML

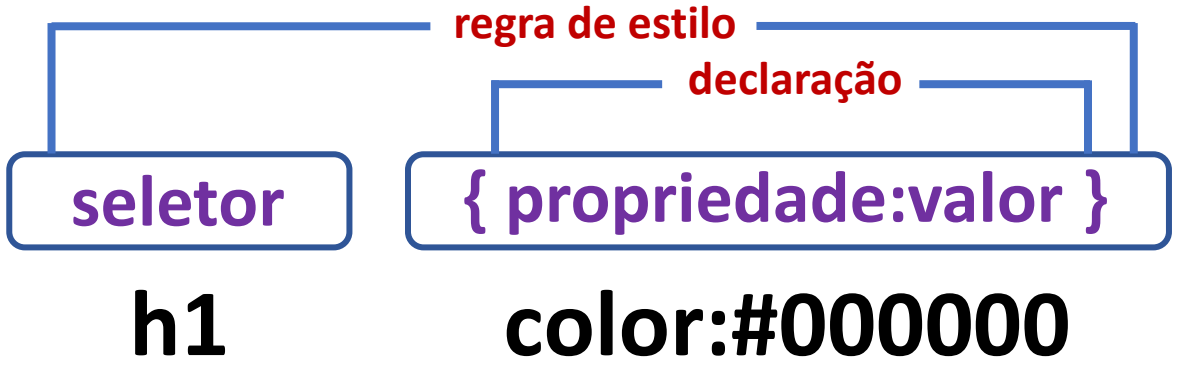
```
1.  <!--Informa ao navegador que é o início do html. É a tag de abertura-->
2.  <html>
3.    <!--Inicia a página. Aqui é determinada a cabeça da página-->
4.    <head>
5.      <!--Título da página que aparece na barra de título da janela do navegador.-->
6.      <!--A tag do elemento título está dentro da tag do elemento head-->
7.      <title>Programação para Internet</title>
8.    <!--Aqui finaliza a cabeça da página-->
9.    </head>
10.   <!--Tag de abertura do elemento body, ou seja o corpo da página-->
11.   <body>
12.   <!--Tudo que estiver entre as tags <body> e </body> é visível no navegador-->
13.     <!--Informa ao navegador onde começa e termina o cabeçalho de nível 1-->
14.     <h1>Bem-vindo(a) ao Curso de Programação para Internet</h1>
15.     <!--Informa ao navegador onde começa e termina o parágrafo-->
16.     <p>Esta é a minha primeira página Web.</p>
17.   <!--Tag de fechamento do elemento body-->
18.   </body>
19. <!--Informa o término da página. É a tag de fechamento-->
20. </html>
```

CSS – Folha de Estilo em Cascata

Lembre-se:

- HTML é utilizado para estruturar conteúdos
- CSS é utilizado para formatar conteúdos

A regra de estilo ou marcação CSS é composta por três partes distintas: um seletor, uma propriedade e um valor para a propriedade.



Formas de declarar folhas de estilo

Existem três formas de declarar as folhas de estilo:

- **inline:** as regras de estilo são escritas diretamente dentro da tag de abertura do elemento.
- **incorporado:** as regras de estilo são escritas dentro da seção head do documento e utilizando o elemento style. A folha de estilo fica dentro das tags <style> </style>.
- **externo:** as regras de estilo são escritas em um arquivo externo de texto com a extensão .css. Esse arquivo externo é "linkado" ao arquivo HTML por meio de uma declaração contida no HTML.

Folhas de estilo inline

As regras de estilo são escritas diretamente dentro da tag de abertura do elemento:

```
<html>
  <head></head>
  <body>
    <p style="font-size: large; color: green;">Parágrafo estilizado – CSS inline</p>
  </body>
</html>
```

Folhas de estilo incorporado

As regras de estilo são escritas dentro da seção head do documento e utilizando o elemento style. A folha de estilo fica dentro das tags <style> </style>.:

```
<html>
  <head>
    <style type="text/css">
      p {color:blue; font-family:Georgia; font-size:25px;}
    </style>
</head>
  <body>
    <p>Parágrafo estilizado - CSS incorporado</p>
  </body>
</html>
```

Folhas de estilo externo

As regras de estilo são escritas em um arquivo externo de texto com a extensão .css. Esse arquivo externo é "linkado" ao arquivo HTML por meio de uma declaração contida no HTML :

```
/*style.css*/
p {
  color: red;
  font-family: Georgia;
  font-size: 20px;
}
```

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style.css" />
  </head>
  <body>
    <p>Parágrafo estilizado - CSS externo</p>
  </body>
</html>
```

Folhas de estilo importadas

Nesse método, declara-se a folha de estilo externa utilizando a diretiva **@import** dentro do elemento *style* na seção **head** do documento HTML.
A diretiva **@import**, ela deverá vir antes de todas as declarações de folha estilo para o documento.

```
/*fontes.css*/
p {
  color: green;
  font-family: Georgia;
  font-size: 30px;
}
```

```
/*main.css*/
@charset utf-8;
@import url("fontes.css");
  body {
    margin: 50;
    background: #00ffff;
   }
```

```
<html>
  <head>
    <style rel="stylesheet" type="text/css">
      @import url("main.css");
    </style>
  </head>
  <body>
    <p>Parágrafo estilizado - CSS importado</p>
  </body>
</html>
```

Referências

MARCONDES, C. A. HTML 4.0 Fundamental: A Base da Programação para Web. São Paulo: Editora Érica, 2005.

OLIVIERO, C. A. Faça um site JavaScript - Orientado por Projeto: scripts baseados em objetos. São Paulo: Editora Érica, 2001.

SILVA, M. S. Criando Sites com HTML - Sites de Alta Qualidade com HTML e CSS. São Paulo: Novatec Editora, 2008.

Prof. MSc. Marcos Alexandruk

E-mail: alexandruk@uni9.pro.br

https://github.com/alexandruk/desenvolvimentoparainternet

JavaScript e tags meta

aula 3

Iniciando com JavaScript

JavaScript é uma linguagem de programação utilizada com o HTML, com a finalidade de tornar as páginas Web dinâmicas e que possibilitem uma interatividade com o usuário.

O HTML fornece a estrutura da página, o CSS adiciona o estilo, e o JavaScript torna a página dinâmica, podendo exibir mensagens para o usuário, validar campos em formulários, manipular datas, detectar o navegador, entre outras coisas.

Com JavaScript podemos criar pequenos programas denominados scripts, que são inseridos em um documento HTML.

Para inserir um script em um documento HTML precisamos do par da tag **<script> </script>**.

Um script, escrito na linguagem JavaScript, pode ser inserido em duas áreas de um documento HTML, entre as tags **<head> </head>** , ou entre as tags **<body> </body>**.

Seja qual for o local escolhido para inserir o script, ele deve sempre iniciar com **<script language="JavaScript">** e terminar com **</script>**.

Iniciando com JavaScript

```
<html>
  <head>
    <title>Primeira pagina com JavaScript</title>
  </head>
  <body>
    <script language="JavaScript">
      document.write("Texto impresso com uma função JavaScript");
    </script>
  </body>
</html>
```

Iniciando com JavaScript

O código JavaScript, inicia na tag **<script language="JavaScript">** e informa ao navegador, que neste ponto começa o script utilizando a linguagem JavaScript. A linha seguinte contém o objeto **document** seguido pelo método **write( )**, cuja função é escrever a frase “Exemplo da utilização de JavaScript". A tag **</script>** finaliza o código.

No JavaScript cada instrução é finalizada com ponto-e-vírgula (;).

JavaScript é case-sensitive, ou seja, é sensível a letras maiúsculas e minúsculas: "A" é diferente de "a".

A sintaxe a seguir demonstra a inclusão de linhas ou blocos de comentários em JavaScript.

```
// Este é um comentário de linha

/*Aqui estamos utilizando um bloco de comentário
  em mais de uma linha */
```

Como inserir arquivo JavaScript externo em um documento HTML

JavaScript também pode ser escrito em um arquivo separado do HTML com a extensão **.js**, por exemplo, **meujavascript.js**, que contém o código que poderá ser utilizado em diferentes páginas Web.

O arquivo externo é chamado no HTML, utilizando o atributo "**src**" na tag script.

O atributo "**src**" indica qual o arquivo JavaScript deverá ser utilizado.

```
<head>
  <script type="text/javascript" src="meujavascript.js">
  </script>
</head>
```

O arquivo **.js** deve conter somente instruções JavaScript, não podendo constar dele nenhum código HTML, nem mesmo as tags <script> e </script>.

Como inserir arquivo JavaScript externo em um documento HTML

```
/*meujavascript.js*/
document.write("Exemplo da utilizacao de JavaScript ");


<!--aula03b.html-->
<html>
  <head>
    <title>Segunda pagina com JavaScript</title>
  </head>
  <body>
    <script type="text/javascript" src="meujavascript.js"></script>
  </body>
</html>
```

Elemento meta

O elemento meta tem como principal objetivo fornecer informações adicionais sobre o documento HTML.

Para cada tipo de informação utilizamos um atributo que pode ser inserido na seção **head** da página.

Alguns são de uso generalizado e servem para qualquer tipo de página, outros são bastante específicos.

O elemento **meta** destina-se a fornecer informações sobre o documento. As informações contidas nele são chamadas de **metadados**.

Metadados incorporados ao código HTML são, na verdade, estruturas de dados sobre os próprios dados, uma breve descrição do conteúdo da página, seu autor, data de criação, linguagem e outras informações relevantes.

Alguns sistemas de busca dão aos conteúdos das tags meta uma forte ênfase no ranking dos sites, a maioria deles indexa os dados das meta tags **description** e **keywords** como sumários da página.

Se essas tags forem usadas correta e racionalmente, elas podem aumentar a relevância nos resultados de busca, o que é vantajoso tanto para o proprietário do site quanto para os usuários.

Sintaxe e exemplos do elemento meta

A sintaxe para escrever um elemento meta consiste em duas partes: **name** ou **http-equiv** define um nome para o metadado e **content** define o conteúdo.

```
meta name="author" content="Fulano de Tal" /
```

Define o autor do documento.

```
meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /
```

Informa que o tipo de conteúdo do documento é texto HTML e a codificação de caracteres é a UTF-8.

```
meta name="language" content="pt-br" /
```

Informa que o idioma principal do documento é o português do Brasil.

```
meta name="description" content="Aulas de HTML, CSS e JavaScript" /
```

Descreve o conteúdo principal do documento. É utilizada pelos mecanismos de busca para descrever o link para a página no resultado de uma busca.

```
name="description" content="none" /
```

Usada para deixar por conta do mecanismo de busca a escolha da frase que irá aparecer no resultado da busca.

```
meta name="keywords" content="css, html, xhtml, javascript" /
```

Keywords usadas por alguns motores de busca para indexar os documentos com informações encontradas em title e body. As frases ou palavras devem ser separadas por vírgulas.

```
meta http-equiv="pragma" content="no-cache" /
```

Faz com que o navegador não armazene a página em cache.

```
meta http-equiv="Content-Script-Type" content="text/javascript" /
```

Informa que a linguagem para scripts inline é a JavaScript.

```
meta http-equiv="Content-Style-Type" content="text/css" /
```

Informa que a linguagem utilizada para estilos inline é CSS.

Robots

Especifica informações de indexação para os robôs de busca, suporta os seguintes valores:

all: valor default, sem restrições para a indexação, o robô de busca não recebe nenhuma informação.

index: os robôs de busca podem incluir a página normalmente.

follow: robôs podem indexar a página e ainda seguir os links para outras páginas que ela contém.

noindex: os links podem ser seguidos, mas a página não é indexada.

nofollow: a página é indexada, mas os links não são seguidos.

none: os robôs podem ignorar a página.

noarchive: (apenas Google): a página não é arquivada.

Alguns exemplos utilizando robots

Os valores "**index**" e "**noindex**" se referem ao tratamento da página inicial: se o buscador deve ou não incluí-la nos resultados, respectivamente. Já os valores "**follow**" e "**nofollow**" se referem aos links da página inicial, se eles devem ser visitados e indexados ou não.

```
<meta name="robots" content="all" />
```

Não há restrições para a indexação. É o valor padrão e não terá efeito se for listada explicitamente.

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

Indexa a página inicial e todas as páginas nela referenciadas.

```
<meta name="robots" content="noindex,follow">
```

Não indexa a página inicial, mas indexa as páginas nela referenciadas.

```
<meta name="robots" content="index,nofollow">
```

Indexa a página inicial, mas nenhum link que ela contenha.

```
<meta name="robots" content="noindex,nofollow">
```

Não indexa nem a página inicial e nem seus links.

```
<meta name="robots" content="noarchive">
```

Evita que os sites de busca indexem o site.

Prof. MSc. Marcos Alexandruk

E-mail: alexandruk@uni9.pro.br

https://github.com/alexandruk/desenvolvimentoparainternet

Utilização de cores, dos elementos div e span

aula 4

Cores

As cores possuem um papel muito importante na composição de páginas web.

A linguagem de marcação (X)HTML tem uma sintaxe própria para defini-las.

Cores podem ser declaradas com uso de atributos ou com folhas de estilo.

As cores são indicadas em valores RGB, ou seja, para conseguir uma cor qualquer misturaremos quantidades de vermelho (Red), verde (Green) e azul (Blue).

Para declarar as cores podemos utilizar a sintaxe hexadecimal, sintaxe RGB ou sintaxe por palavra-chave.

Sintaxe hexadecimal

A utilização da sintaxe hexadecimal é muito utilizada pelos desenvolvedores.

Os valores RBG são indicados em numeração hexadecimal, em base 16, ou seja, os dígitos vão de 0 até 16. Como não existem tantos dígitos numéricos, utilizam-se as letras de A a F. Veja no quadro, a seguir, a numeração no sistema hexadecimal correspondente ao sistema decimal.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Sintaxe hexadecimal

Na sintaxe hexadecimal a cor é definida por uma sequência de seis caracteres hexadecimais precedidos por #.

A sintaxe não é case sensitive, ou seja, não faz diferença entre letras maiúsculas ou minúsculas.

Para conseguir uma cor misturamos os valores na forma: #RRGGBB, em que cada valor pode variar de 00 até FF.

Quando temos todas as cores (R,G,B) obtemos a cor branca, que é representada por **#FFFFFF**.

A ausência de cores, o preto, é representado por **#000000**.

Para a cor vermelha usamos **#FF0000**.

Para o verde usamos **#00FF00**.

Para o azul usamos **#0000FF**.

Podemos também, quando possível, utilizar a sintaxe abreviada para as cores declarando somente três valores:

#FFF equivale a **#FFFFFF**.

#0F0 equivale a **#00FF00**.

#F6C equivale a **#FF66CC**.

Sintaxe RGB

Essa sintaxe, menos utilizada, define a cor por uma lista de três números colocados entre parênteses e separados por vírgula, precedida da sigla rgb. Existem duas formas de escrever a lista dos três números.

A primeira utiliza números entre **0** e **255**:

rgb(125, 220, 90)

rgb(0, 75, 250)

A segunda utiliza valores em porcentagem de **0%** a **100%**:

rgb(10%, 90%, 45%)

rgb(0, 35%, 82%)

Não é permitido misturar números com porcentagem para compor a cor.

Sintaxe por palavra-chave

Podemos também utilizar o nome da cor em inglês, porém, de acordo com o W3C, na CSS2 somente 16 cores eram válidas para efeito de marcação (conforme tabela abaixo). Na CSS3, a lista de cores suportadas foi estendida para 147 palavras-chave (disponível em: https://www.w3schools.com/colors/colors_names.asp)

Black	Purple
White	Fuchsia
Red	Lime
Green	Olive
Blue	Yellow
Silver	Navy
Gray	Teal
Maroon	Aqua

Cores compatíveis em todos os sistemas

Como as páginas web têm de ser vistas por todos os usuários, e os sistemas que eles utilizam são diferentes, é recomendável utilizar cores compatíveis com a paleta de cores de todos eles.

Uma forma de conseguir isso é limitar a combinação de cores utilizando os valores da seguinte forma:

Utilizar os valores
00
33
66
99
CC
FF

Podemos utilizar, por exemplo, as seguintes combinações: **#3366FF** - **#FF9900** - **#666666**

Atributo id

O atributo **id** é um identificador único e destina-se a atribuir um nome identificador para o elemento. Em um mesmo documento, o nome usado para identificar um elemento deve ser único, ou seja, não podemos atribuir o mesmo nome de um id para outros elementos.

O valor do atributo **id** é um nome que você pode escolher. Devemos respeitar algumas regras para a escolha do nome: não é válido começar o nome de um atributo com um número ou hífen. Já o caractere underscore (_) é válido. Na prática utilizam-se nomes que começam com letras minúsculas.

A utilização do atributo **id** é útil em consequência dos programas, scripts e folhas de estilo terem a capacidade de entender essa marcação no HTML, conseguindo selecionar os elementos pelo seu identificador.

Atributo id

A declaração é feita na seção **head**, os elementos encontram-se no **body** e utilizam o **id** declarado.

```
<html>
  <head>
    <title>Atributo Id</title>
    <style type="text/css">
      /*Declaração dos atributos id*/
      p#destaque{
        font-family: sans-serif; /*indica qual a fonte utilizada*/
        font-style: italic; /*indica o estilo da fonte: itálico*/
        font-size: 30px; /*tamanho da fonte: 30 pixels*/
        text-align: left; /*alinhamento do texto: esquerda*/
      }
    </style>
  </head>
  <body>
    <p id="destaque">Paragrafo com atributo id "destaque"</p>
  </body>
</html>
```

Atributo class

O atributo **class** destina-se a atribuir uma classe identificadora para o elemento.

Também é um atributo identificador, porém a diferença dele e do atributo **id** é que com o atributo **class** podemos, em um mesmo documento, utilizar o nome da classe para identificar várias instâncias de um mesmo elemento e também elementos diferentes.

Uma folha de estilo é capaz de identificar os diferentes elementos identificados com a mesma classe e estilizar todos eles de acordo com a declaração dela.

Atributo class

Veja a sintaxe e como declarar um atributo class.

```
<html>
  <head>
    <title>Atributo class</title>
    <style type="text/css">
      /*Declaração do atributo class*/
      p.fonte1{
        font-size:20px; /*tamanho da fonte: 20 pixels*/
        color: #ff0000; /*cor do texto: verde*/
      }
      p.fonte2{
        font-size:30px; /*tamanho da fonte: 30 pixels*/
        color: #00ff00; /*cor do texto: verde*/
      }
    </style>
  </head>
  <body>
    <p class="fonte1">Texto do parágrafo marcado com class fonte1.</p>
    <p class="fonte2">Texto do parágrafo marcado com class fonte2.</p>
  </body>
</html>
```

Elementos span e div

Os elementos **div** e **span**, em conjunto com os atributos **id** e **class**, tornam-se um poderoso mecanismo para estruturar e estilizar conteúdos.

Esses elementos definem conteúdo para ser utilizado **inline** (**span**) ou em nível de **bloco** (**div**).

Devem ser utilizadas com folhas de estilo em cascata (CSS).

Elemento span

Utiliza-se o elemento **span** com o atributo **class** para marcar conteúdos **em linha** genericamente.

```
<html>
  <head>
    <title>Utilizando span</title>
    <style type="text/css">
      span.azul{
        color: #0000FF; /*cor do texto: azul*/
      }
    </style>
  </head>
  <body>
    <p>A Terra é <span class="azul">azul</span>, disse Gagarin.</p>
  </body>
</html>
```

Elemento div

Utiliza-se o elemento **div** com o atributo **class** para estruturar e estilizar blocos de conteúdos.

```
<html>
  <head>
    <style>
      .azulBox {
        background-color: #0000ff; /*cor do fundo: azul*/
        text-align: center; /*alinhamento do texto: centro*/
        font-size:30px; /*tamanho da fonte: 20 pixels*/
        color: #ffffff; /*cor da fonte: branco*/
      }
    </style>
  </head>
  <body>
    <div class="azulBox">
      <p>Parágrafo dentro da div azulBox</p>
    </div>
      <p>Parágrafo fora da div azulBox</p>
  </body>
</html>
```

Prof. MSc. Marcos Alexandruk

E-mail: alexandruk@uni9.pro.br

https://github.com/alexandruk/desenvolvimentoparainternet

Padrões de medidas e formatação de conteúdos com o atributo style

aula 4

Box model

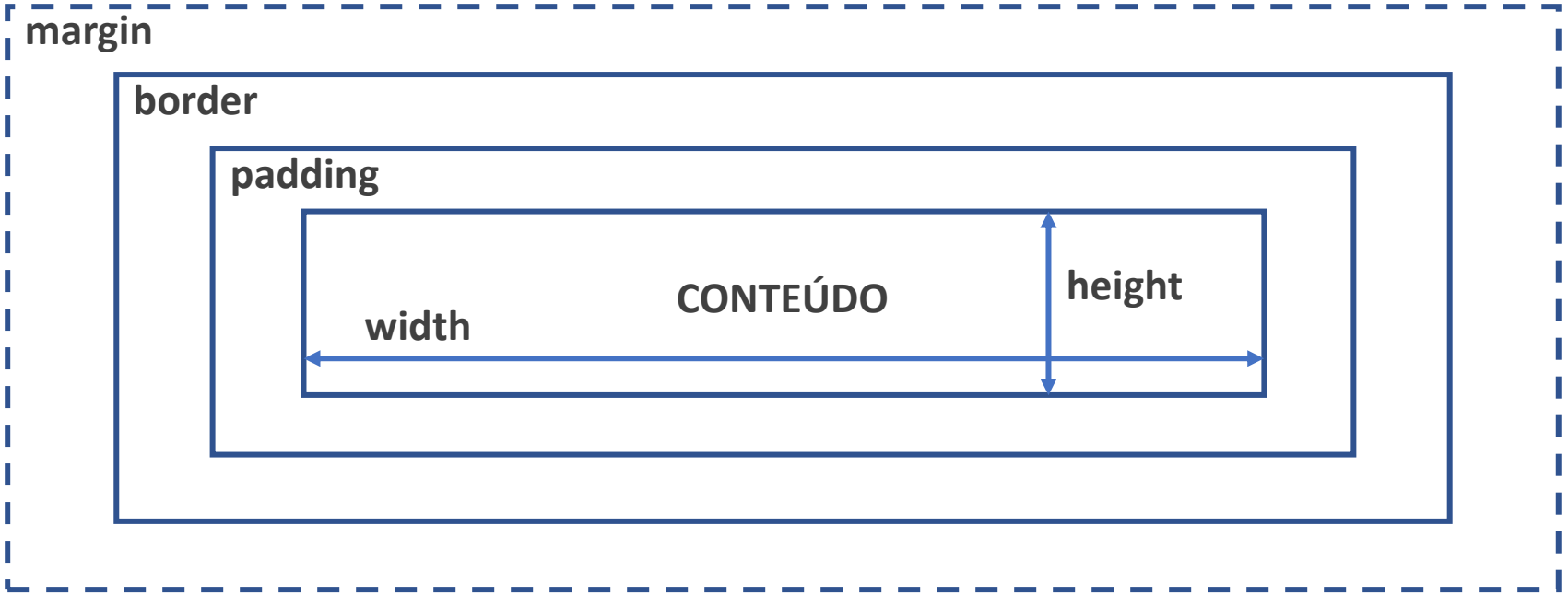
Podemos entender como boxes os blocos formados por um dos elementos HTML quando renderizados em uma tela.

Por exemplo, um parágrafo contendo um texto ocupa na tela um retângulo com uma largura e uma altura bem definida, que é fácil de visualizar.

Box Model é um modelo-padrão de renderização ou apresentação visual de um box de acordo com a formatação CSS.

Box model

Observe na figura abaixo que existe uma área denominada de conteúdo e suas dimensões são definidas pelas propriedades CSS **width** e **height**. Depois temos a área de enchimento, em que a espessura é definida pela propriedade **padding**. Ainda, em volta do enchimento, temos uma borda em que sua espessura e cor são definidas na propriedade **border**. Por último, existe um espaço chamado de margem e sua espessura é definida na propriedade **margin**. A área da margem é sempre transparente.
A propriedade **background** define o fundo a ser aplicado nas áreas de conteúdos, de enchimento e da borda.



Box model

Para definir corretamente a altura e largura de um elemento precisamos saber como o box model trabalha.

Note que quando você especificar as propriedades de altura e largura de um elemento com CSS, você está apenas definindo a altura e largura da área do conteúdo. Para saber o tamanho total do elemento, você deverá adicionar o enchimento, a borda e a margem.

O total do width de um elemento sempre deverá ser calculado da seguinte forma:

width total do elemento = width + left padding + right padding + left border + right border + left margin + right margin

O total do height de um elemento sempre deverá ser calculado da seguinte forma:

height total do elemento = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

Veja as seguintes definições de um determinado elemento:

width:250px; padding:10px; border:5px solid gray; margin:10px;

250px do width;
+ 20px do padding (10px para a direita e 10px para a esquerda);
+ 10px de border (5px para a direita e 5px para a esquerda);
+ 20 px de margin (10px para a direita e 10px para a esquerda).

300px comprimento total do elemento

Unidades de medida em CSS

As unidades de medida são utilizadas em propriedade CSS que controlam dimensões em geral. Normalmente se utiliza em CSS a unidade de medida em pixel (px) ou o centímetro (cm), que raramente é utilizado. Há unidades de medida absolutas e relativas.

sigla	unidade absolutas
in	Polegadas
cm	Centímetros
mm	Milímetros
pt	Ponto. Unidade de medida tipográfica (1 pt é igual a 1/72 polegadas)
pc	Paica. Unidade de medida tipográfica (1 pc é igual a 12 pontos)
sigla	**unidade relativas**
em	1 em é igual ao tamanho da fonte definido para o elemento. O em é uma unidade muito útil em CSS, uma vez que pode se adaptar ao tipo de fonte utilizada.
ex	1 ex é igual à altura da letra x minúscula da fonte definida para o elemento. Não é comum a utilização dessa medida.
px	Pixel. Um ponto na tela do computador.
%	Porcentagem. Calculada em relação a um valor preexistente, normalmente, uma unidade de medida.

Propriedade margin (margem)

A propriedade **margin** define uma margem em volta dos boxes. O valor declarado pode ser qualquer unidade CSS de medida.

Propriedade	Descrição	Declaração
margin	Define todas as margens na declaração	margin: 20px 30px 5px 10px;
margin-top	Define a margem superior	margin-top: 20px;
margin-right	Define a margem direita	margin-right: 30px;
margin-bottom	Define a margem inferior	margin-bottom: 5px;
margin-left	Define a margem esquerda	margin-left: 10px;

Propriedade margin (margem)

As propriedades CSS padding, font, background, margin, border, border-width, borderstyle, border-color e outline admitem a sintaxe abreviada, que consiste em declarar uma lista de valores separados por espaço, como apresentado no quadro (margin: 20px 30px 5px 10px;).

A ordem em que os valores são escritos na lista corresponde aos lados superior, direito, inferior e esquerdo, respectivamente.

Veja também outras opções para declaração abreviada:

margin: 20px; /* margem de 20px nos quatro lados */

margin: 15px 10px; /* margem superior e inferior de 15px e direita e esquerda de 10px */

margin: 20px 10px 15px; /* margem superior de 20px, margens direita e esquerda de 10px e margem inferior de 15px */

Propriedade padding (espaçamento interno)

A propriedade **padding** define um espaçamento interno entre o conteúdos do box e seus limites. O valor declarado pode ser qualquer unidade CSS de medida.

Propriedade	Descrição	Declaração
padding	Define todos os espaçamentos na declaração	padding: 10px 5px 25px 12px;
padding-top	Define o espaçamento superior	padding-top: 20px;
padding-right	Define o espaçamento direito	padding-right: 30px;
padding-bottom	Define o espaçamento inferior	padding-bottom: 5px;
padding-left	Define o espaçamento esquerdo	padding-left: 10px;

Propriedade background (fundo)

A propriedade background define as características de fundo do elemento. Com essa propriedade podemos definir a cor, imagens e seu posicionamento no fundo do elemento.

Veja no quadro as principais definições que podemos utilizar.

Propriedade	Descrição
background	Define todas as propriedades de fundo na declaração
background-attachment	Define se uma imagem ficará fixa ou se irá rolar em relação à área de renderização em que foi colocada
background-color	Define uma cor para o fundo de um elemento
background-image	Define uma imagem de fundo de um elemento
background-position	Define uma posição inicial para a imagem de fundo
background-repeat	Define como a imagem se repetirá no fundo do elemento
background-size	Define o tamanho da imagem de fundo

Exemplo

```
<p>Parágrafo sem especificação de margem</p>

<p style="background-color: ffff00; margin-top:100px; margin-bottom:100px; margin-righ px;
margin-left:50px;"> Parágrafo com especificação de margem</p>

<p style="border-style= solid; border-color: #f00; padding: 10px ; margin: 20px;">Parágrafo com
padding, border e margin</p>

<p>Parágrafo normal, sem padding</p>

<p style="padding-left: 2cm;">Parágrafo com padding-left de 2cm</p>

<p style="padding-left: 50%;">Parágrafo com padding-left de 50 %</p>
```

Exercício

Aplicar os estilos acima através de folhas de estilo (CSS) externas:

a) com seletor de classe

b) com seletor de id

Seletor de classe aplicado a qualquer elemento

Os seletores de **classe** também podem ser definidos sem colocarmos o nome de um elemento no início da definição. Quando isso acontece as definições podem ser aplicadas a qualquer elemento.

```
.a {
  background-color: #ffff00;
  margin: 50px;
  color: #ff0000;
}
```

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style05d.css" />
  </head>
  <body>
    <h1>Título sem especificação</h1>
    <h1 class="a">Título sem especificação</h1>
    <p>Parágrafo sem especificação</p>
    <p class="a">Parágrafo com especificação</p>
  </body>
</html>
```

Seletor de id aplicado a qualquer elemento

Se quisermos que a regra se aplique a qualquer **elemento** que tenha determinado, para1 basta escrevê-la na forma seguinte:

```
*#a {
  background-color: #ffff00;
  margin: 50px;
  color: #ff0000;
}
```

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style05d.css" />
  </head>
  <body>
    <h1>Título sem especificação</h1>
    <h1 id="a">Título sem especificação</h1>
    <p>Parágrafo sem especificação</p>
    <p id="a">Parágrafo com especificação</p>
  </body>
</html>
```

Exemplo

aula05g.html

Validador W3C

Existem os padrões recomendados pelo W3C para criação e desenvolvimento de páginas Web, mas como saber se nossa página está de acordo com esses padrões?

É aí que entra o órgão para validar nosso trabalho, o Markup Validation Service: **http://validator.w3.org**.

Essa ferramenta de validação é on-line e nos ajuda a corrigir erros que possam existir na criação da nossa página web.

A cada passo do desenvolvimento de um documento para web, você deve validar sua marcação. Ficará mais fácil corrigir cada etapa do que deixar para validar o site todo de uma só vez.

O validador HTML verifica se a marcação do documento está de acordo com a sintaxe prevista para a versão HTML ou XHTML, devendo constar no documento o DOCTYPE, que contém as informações sobre a versão e sintaxe que será avaliada.

Existem três maneiras de validar um documento:

- **Validate by URI:** informa-se o endereço URL da página para documentos na internet;
- **Validate by File Upload:** para validar documentos que estão no seu HD informando o local do documento;
- **Validate by Direct Input:** Copie e cole a marcação no campo texto.

Adquira o hábito de validar sua marcação, assim sempre facilitará o seu trabalho de desenvolvimento.