

Prof. Esp. Thalles Canela

- **Graduado:** Sistemas de Informação - Wyden Facimp
- **Pós-graduado:** Segurança em redes de computadores - Wyden Facimp
- **Consultor de Tecnologia - [aXR6] Cyber Security e NtecSoftware**
- **Professor no Senac (contratado)**
- **Professor na Wyden Facimp (contratado)**
 - **Pós-graduação:** Segurança em redes de computadores - Wyden Facimp
- **Professor na Wyden Facimp (Efetivado)**
 - **Graduação:** Análise e desenvolvimento de sistemas - Wyden Facimp

Redes sociais:

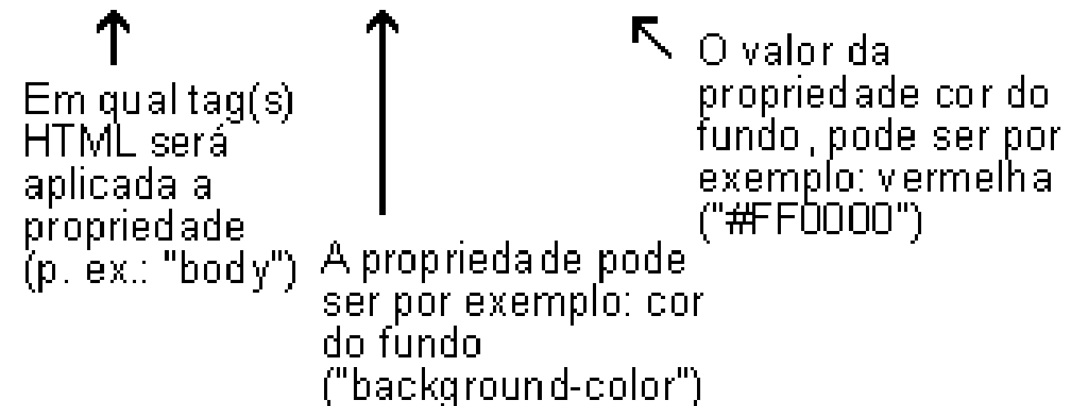
- **Linkedin:** <https://www.linkedin.com/in/thalles-canela/>
- **YouTube:** <https://www.youtube.com/aXR6CyberSecurity>
- **Facebook:** <https://www.facebook.com/axr6PenTest>
- **Instagram:** https://www.instagram.com/thalles_canela
- **Github:** <https://github.com/ThallesCanela>
- **Github:** <https://github.com/aXR6>
- **Twitter:** <https://twitter.com/Axr6S>

LINGUAGEM DE MARCAÇÃO E ESTILOS CSS

Sintaxe

- Seletor: define o elemento que sofrerá a formatação. Pode ser seletor de elemento, classe, id etc.
- Propriedade: é o nome da propriedade a ser alterada (Ex: cor, fonte, tamanho, alinhamento, etc).
- Valor: é o conteúdo atribuído à propriedade. Se o valor tiver mais de uma palavra separada por espaço ou aspas duplas. Não deixe espaço entre o valor e sua unidade de medida.

seletor { **propriedade**: **valor** }



Seletor tipo com uma propriedade:

- Define apenas uma propriedade de formatação para um elemento ou mais elementos.

Exemplo: seletor { propriedade: valor }

Estilizando: h1 {color: #ff0000;}

Aplicando: <h1> Titulo cor vermelha</h1>

Seletor tipo com diversas propriedades:

- Definem diversas propriedade de formatação para um elemento. As propriedades devem ser separadas por ponto-e-vírgula.

Exemplo: `seletor{ propriedade: valor; propriedade: valor }`

Estilizando: `p { text-align: center; color: #FFFFFF; font-size: 18px; }`

Estilizando: `body { color: #000000; font-family: Tohoma; }`

Aplicando: `<body> <p> Corpo do Texto </p> </body>`

Diferentes seletores tipo agrupados com propriedades em comum:

- Usado por diferentes elementos do documento que usarão uma formatação em comum.

Exemplo: seletor1, seletor2, seletorN { propriedade: valor;
propriedade: valor }

Estilizando: p, h1, h2, h3 { color: Green; }

Aplicando: <h1>Título nível 1 cor verde</h1>

Aplicando: <h2>Título nível 2 cor verde</h2>

Aplicando: <h3>Título nível 3 cor verde</h3>

Aplicando: <p>Parágrafo cor verde</p>

Seletores de classe para o mesmo tipo de elemento:

- Diferentes classes de formatação podem ser criadas para um mesmo tipo de elemento. Para definir classe para um elemento usamos o nome do elemento + o ponto + um nome(sem espaço/acentos/caracteres_especiais).

Exemplo: nome_da_tag . nome_da_classe { propriedade: valor; propriedade: valor }

Estilizando: p.centralizado { text-align: center; color: green; }

Estilizando: p.justificado { text-justify: center; color: blue; }

Aplicando: <p class="centralizado">Centralizado</p>

Aplicando: <p class="justificado ">Justificado</p>

Sub-top.: Um elemento também pode usar mais de uma classe ao mesmo tempo.

Estilizando: `p.vermelho { color: red; }`

Estilizando: `p.esquerda { text-align: left; }`

Estilizando: `p.direita { text-align: right; }`

Aplicando: `<p class="vermelho direita">Vermelho à direita</p>`

Aplicando: `<p class="vermelho esquerda">Vermelho à esquerda</p>`

Seletor de classe para qualquer elemento:

- Uma classe pode ser também usada por diversos elementos HTML que terão uma aparência comum. Para definir uma classe para qualquer elemento usamos o ponto + um nome (sem espaço/acentos/caracteres especiais).

Exemplo: nome_da_classe { propriedade: valor; propriedade: valor }

Aplicando: .fundo { background-color: rgb(250, 0, 255) }

Aplicando:

```
<h1 class="fundo">Título com cor de fundo</h1>
<p class="fundo">Parágrafo com cor de fundo</p>
<table class="fundo">
  <tr>
    <td>Conteúdo</td><td>Conteúdo</td>
  </tr>
</table>
<!-- tabela com cor de fundo -->
```

Seletor de ID para um único elemento:

- Ao criar uma regra com ID, apenas um único elemento da página poderá usar a formatação. Para definir uma regra um elemento único usamos o sinal # + um nome (sem espaço/acentos/caracteres especiais e nem início numérico).

Exemplo: `#nome_do_id { propriedade: valor; propriedade: valor }`

Estilizando: `#tituloPrincipal { color: white; text-align: center; } .destaque { background-color: yellow; }`

Aplicando: `<h1 id="tituloPrincipal">Título no padrão</h1>`

Sub-top.: O elemento pode também usar ID e CLASSE ao mesmo tempo:

- `<h1 id="tituloPrincipal" class="destaque">Título no padrão</h1>`

Seletores aninhados:

- Usado para elementos que estão aninhados no documento. Assim elementos com aninhamento diferente do apontado na regra não usarão a formatação.

Exemplo: nome_da_tag_externa nome_da_tag_externa { propriedade: valor; propriedade: valor }

Estilizando: p b { font-size: 18pt; }

Aplicando: <p>Texto normal Texto negrito tamanho 18 cor azul.</p>

Sub-top.: Já o conteúdo da tag que está dentro da tag NÃO apresenta o estilo acima, pois possui outro aninhamento:

```
<ol>
```

```
  <li><b>texto negrito</b></li>
```

```
  <li><b>texto negrito</b></li>
```

```
</ol>
```

Seletor universal:

- Usado para todos os elementos do documento. É uma configuração em comum.

Exemplo: `* { propriedade: valor; propriedade: valor }`

Estilizando: `* { font-size: 12pt; }`

Aplicando: `<p>Parágrafo com fonte tamanho 12pt</p>`

Aplicando: `<p>Título com fonte tamanho 12pt</p>`

Aplicando: `<p>Título com fonte tamanho 12pt</p>`

Comentários em CSS

- São usados para documentar o código e facilitar a manutenção.

```
<style type="text/css">
    /* comentário de linha */
    p {
        font-family: arial;
        color: red;
    }
    /* comentário
    de bloco ou
    várias linhas */
</style>
```

Modos/Formas de Uso

- Estilo inline;
- Estilo incorporado;
- Estilo vinculado/externo;

Estilo inline

- Os estilos in-line (na linha) são utilizados na própria tag, sendo criados elemento a elemento, com seu próprio estilo. Essa forma de trabalho se assemelha muito aos atributos de formatação que os elementos oferecem. Trabalhando com estilos in-line, não somos capazes de desfrutar de sua capacidade de reaproveitamento de código e padronização de estilos. Contudo, caso o desenvolvedor esteja usando também CSS em arquivo vinculado ou incorporado, o estilo In-Line é que prevalece para a formatação do elemento.
- Para utilizarmos o estilo in-line, devemos usar o atributo style na tag desejada.

Estilo inline

```
<html>
<head>
  <title>Programando em HTML e CSS 1</ title>
</head>
<body>
  <p style="font-family: arial; color: red">Parágrafo usando estilo in-line</p>
</body>
</html>
```

Estilo incorporado

- É quando o bloco de estilo CSS fica na seção de cabeçalho dentro da mesma página criando-se uma folha de estilos padrão para toda a página. Esta prática é melhor que utilizar estilos in-line, mas limita-se apenas a página corrente e não podem ser compartilhados com outras páginas. Caso o desenvolvedor esteja usando também CSS em arquivo vinculado, o estilo incorporado é que prevalece para a formatação do elemento.
- Esta seção definida no início da página vai dentro da seção de cabeçalho do documento.

Estilo incorporado

```
<html>
<head>
  <title>Programando em HTML e CSS 1</ title>
  <style type="text/css">
    p {
      font-family: arial;
      color: red
    }
  </style>
</head>
<body>
<p>Parágrafo utilizando estilo incorporado</p>
</body>
</html>
```

Estilo vinculado/externo

- É quando se cria um arquivo unicamente de estilos, também chamado de página de estilo ou folha de estilo. Este arquivo deve ter todas as regras de estilos para todos os elementos de todas as páginas do site. Então, basta vincular suas páginas ao arquivo de estilos, e todas seguirão o mesmo padrão. A página de estilos vinculada é uma excelente prática de desenvolvimento HTML, facilitando o desenvolvimento e a manutenção dos sites. A extensão do arquivo deve ser .css.

Estilo vinculado/externo

Arquivo CSS (estilo1.css):

```
p {  
    font-family: arial;  
    color:red;  
}
```

Arquivo HTML (index.html):

```
<html>  
<head>  
    <title>Programando em HTML 4.0 e CSS 1</title>  
    <link rel="stylesheet" href="estilo1.css" type="text/css">  
</head>  
<body>  
    <p>Parágrafo utilizando estilo vinculado/externo</p>  
</body>  
</html>
```

Método 1: In-line (o atributo style)

- Uma maneira de aplicar CSS é pelo uso do atributo style do HTML. Tomando como base o exemplo mostrado anteriormente a cor vermelha para o fundo da página pode ser aplicada conforme mostrado a seguir:

```
<html>
  <head>
    <title>Exemplo</title>
  </head>
  <body style="background-color: #FF0000;">

    <p>Esta é uma página com fundo vermelho</p>
  </body>
</html>
```

Método 2: Interno (a tag style)

- Uma outra maneira de aplicar CSS e pelo uso da tag <STYLE> do HTML. Como mostrado a seguir:

```
<html>
  <head>
    <title>Exemplo</title>
    <style type="text/css">

      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p>Esta é uma página com fundo vermelho</p>
  </body>
</html>
```


Método 3: Externo (link para uma folha de estilos)

- O método recomendado é o de linkar para uma folha de estilos externa. Usaremos este método nos exemplos deste tutorial.
- Uma folha de estilos externa é um simples arquivo de texto com a extensão .css. Tal como com qualquer outro tipo de arquivo você pode colocar uma folha de estilos tanto no servidor como no disco rígido.
- Vamos supor, por exemplo, que sua folha de estilos tenha sido nomeada de style.css e está localizada no diretório style. Tal situação está mostrada a seguir:



O "truque" é criar um link no documento HTML (default.htm) para a folha de estilos (style.css). O link é criado em uma simples linha de código HTML como mostrado a seguir:

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

Notar que o caminho para a folha de estilos é indicado no atributo href.

Esta linha de código deve ser inserida na seção header do documento HTML, isto é, entre as tags <head> e </head>. Conforme mostrado abaixo:

```
<html>
  <head>
    <title>Meu documento</title>
    <link rel="stylesheet" type="text/css"
href="style/style.css" />
  </head>
  <body>
    ...
```

Este link informa ao navegador para usar o arquivo CSS na renderização e apresentação do layout do documento HTML. A coisa realmente inteligente disto é que vários documentos HTML podem linkar para uma mesma folha de estilos. Em outras palavras isto significa que um simples arquivo será capaz de controlar a apresentação de muitos documentos HTML.



Esta técnica pode economizar uma grande quantidade de trabalho. Se por exemplo, você quiser trocar a cor do fundo de um site com 100 páginas, a folha de estilos evita que você edite manualmente uma a uma as páginas para fazer a mudança nos 100 documentos HTML. Usando CSS a mudança se fará em uns poucos segundos trocando-se a cor em uma folha de estilos central.

Exemplo:

Abra o Notepad (ou qualquer outro editor de texto que queira usar) e crie dois arquivos — um arquivo HTML e um arquivo CSS — com os seguintes conteúdos:

1. index.html

```
<html>
  <head>
    <title>Meu documento</title>
    <link rel="stylesheet" type="text/css" href="style.css"/>
  </head>
  <body>
    <h1>Minha primeira folha de estilos</h1>
  </body>
</html>
```

2. style.css

```
body {
  background-color: #FF0000;
}
```

Salve os dois arquivos na mesma pasta. Lembre-se de salvar os arquivos com a extensão apropriada (".css" e ".html")

As cores podem ser definidas pelo seu valor hexadecimal como no exemplo acima (#ff0000), com uso do nome da cor ("red") ou ainda pelo seu valor rgb (rgb(255,0,0)). Algumas cores pelo nome: red, grey, yellow, blue, green, pink, purple, etc.

Abra **index.html** no seu navegador e veja uma página com o fundo vermelho.

O que é Design Responsivo?

Design Responsivo é uma técnica de estruturação do layout de um site adaptado para acompanhar tamanhos e formatos em que ele está sendo visualizado. Ou seja, a sua resolução em smartphones ou tablets, manterá a mesma coesão e qualidade independentemente do dispositivo ser acessado por desktop ou mobile.

Embora tenha a ver com a aparência de uma página, a técnica não visa apenas a adaptação de layout em telas, mas, sim, em proporcionar uma boa utilização, experiência, e, sobretudo, total entendimento do site em questão.

Vantagens de utilizar Design Responsivo

- diminui a taxa de rejeição — a quantidade de usuários que entram e saem de um site diminui quando adaptado a um Design Responsivo;
- aumenta vendas — embora muitos brasileiros (36% segundo o relatório Digital, Social & Mobile 2015) ainda prefiram fazer compras por desktops, as [vendas pelos dispositivos móveis vem crescendo](#) a cada dia;
- fácil compartilhamento de informações — uma pessoa que possui um smartphone em suas mãos, quando gosta de um assunto, deseja compartilhar com sua rede de amigos, podendo ser por conversas ou redes sociais, no Design Responsivo você possui essa possibilidade de compartilhamento;
- melhor ranqueamento no Google — o Google favorece, em questões de ranqueamento, [sites que possuem versões adaptadas](#) para o Design Responsivo.

O que é Mobile First?

Já no Mobile First, o projeto consiste em fazer a criação de websites primeiro em dispositivos mobiles, para depois adaptar para desktops e outras plataformas. Afinal, o seu nome já diz “móvel primeiro”. Focado na arquitetura, esse conceito surgiu devido às telas dos celulares terem se transformado no principal dispositivo para acesso dos sites pelos usuários.

Vantagens de utilizar Mobile First

- melhor ranqueamento no Google — a empresa anunciou em março de 2018 que o ranqueamento na plataforma seguiria o conceito de Mobile First, ou seja, sites que utilizam dessa técnica são melhores posicionados pelo buscador;
- melhor experiência ao usuário — o Mobile First garante uma experiência completa, informativa e mais agradável aos olhos, pois os desenvolvedores optam por colocar as principais informações a serem exibidas;
- mais credibilidade aos usuários e a outras marcas — em vista que é um conceito inovador, tanto os clientes quanto os concorrentes de mercado irão analisar que a empresa em questão está preocupada em proporcionar uma melhor visita e processo de informações aos seus consumidores;
- mais rápido e imediato — nos dispositivos móveis tudo deve ser mais rápido para otimizar o tempo dos usuários, e é exatamente isso que uma página criada por Mobile First faz.