# Interatividade na Web

Front End Web

Msc. Lucas G. F. Alves

e-mail: lucas.g.f.alves@gmail.com





# Planejamento de Aula

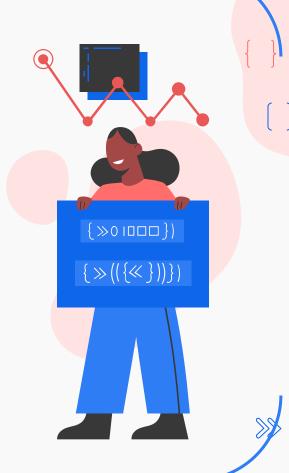
Revisão

Rodapé

Substituição por Imagem

Javascript

Exercícios





# Revisão







Algumas propriedades de elementos pais, quando alteradas, são aplicadas automaticamente para seus elementos filhos em cascata.

```
Exemplo:
```

No exemplo, todos os elementos filhos herdaram o valor da propriedade color do elemento pai a qual eles pertencem.









Existem propriedades que não são aplicadas em cascata, geralmente são as que afetam a caixa (box) do elemento, como width, height, margin e padding.

```
h1 { padding-left: 40px; }
#pai {
        color: blue;
        padding-left: 0;
}
<div id="pai">
        <h1>Sou um título</h1>
        <h2>Sou um subtítulo</h2>
</div>
```

Perceba que o padding do elemento <h1> não foi sobrescrito pelo valor do elemento pai <div>, ou seja, o valor 40px foi mantido.









### Inherit

O valor Inherit indica que o elemento filho terá o mesmo valor do elemento pai.

Toda vez que o tamanho do elemento pai for alterado, automaticamente o elemento filho herdará o novo valor, facilitando assim, a manutenção do código.

O valor inherit também afeta propriedades que não são aplicadas em cascata.







```
>>>
```

```
Inherit
Exemplo:
      <div>
            <img src="box-model.png" alt="box model">
      </div>
      div {
            border: 2px solid;
            border-color: red;
            width: 30px;
            height: 30px;
                              img {
                                                            img {
                                    width: 30px;
                                                                  width: inherit;
                                    height: 30px;
                                                                  height: inherit;
```



# Exercício - HTML





Logo

Nenhum item no carrinho



Página Principal Conteúdo Blog Contato



# **Exercício - HTML**



1) Criar um elemento destaque e, dentro dele, uma section para busca e outra para o menu, na página index.html.

```
<div class="container destaque">
     <section class="busca">
           <h2>Busca</h2> <form><input type="search"><input
           type="image" src="img/busca.png"></form
     </section> <!-- fim .busca -->
     <section class="menu-conteudo">
           <h2>Menu</h2>
           <nav>
                <a href="#">Aula1</a>
                      <a href="#">Aula2</a>
                 </nav> </section><!-- fim .menu-conteudo →
</div> <!-- fim .container .destaque →
```







## Exercício - CSS

2) Aplicar cor de fundo as caixas de busca e no menu, o texto deve estar em negrito e apresentado em maiúsculas. Aplicar também algumas regras de tamanhos e margens. .busca, .menu-conteúdo { background-color: #dcdcdc; font-weight: bold; text-transform: uppercase; margin-right: 10px;width: 230px; float: left; .busca h2, .busca form, .menu-conteudo h2 { margin: 10px; .menu-conteudo li { background-color: white; margin-bottom: 1px; padding: 5px 10px; .menu-conteudo a { color: #333333; text-decoration: none;



.menu-conteudo { clear: left; }





# Exercício - CSS



3) Na busca, use a propriedade vertical-align para alinhar o campo de texto à imagem da lupa pelo centro. Aproveite e coloque o tamanho do campo de texto para melhor encaixar no design e use seletores de atributo do CSS para isso.

```
.busca input { vertical-align: middle;}
.busca input[type=search] { width: 170px;}
```

4) Acerte as margens e posicionamentos no menu lateral e no topo.

```
.destaque { margin-top: 10px; }
.menu-conteudo { margin-top: 10px; padding-bottom: 10px; }
```







# Display Inline-Block

**>>** 

Vamos criar um painel com uma lista de pastas-aulas, onde cada aula será representado por uma .

Por padrão uma possui a propriedade display:block. Porém as pastas de cada aula serão lado a lado. Para isso defini-se a propriedade display como inline.

Também é necessário alterar as propriedades width, margin e padding das , mas agora os elementos são inline e este modo ignora alterações de propriedades da box. Como resolver este problema?







# Display Inline-Block

**>>>** 

Vamos criar um painel com uma lista de pastas-aulas, onde cada aula será representado por uma .

Por padrão uma possui a propriedade display:block. Porém as pastas de cada aula serão lado a lado. Para isso defini-se a propriedade display como inline.

Também é necessário alterar as propriedades width, margin e padding das , mas agora os elementos são inline e este modo ignora alterações de propriedades da box.

Como resolver este problema?

Uma mistura dos dois, o inline-block.

Os elementos que recebem o valor inline-block aceitam as propriedades height (altura) e width (largura) e ainda seguem o inline.







# Display Inline-Block

```
>>
```

```
.painel li {
      display: inline-block;
      vertical-align: top;
      width: 140px;
      margin: 2px;
      padding-bottom: 10px;
}
```







## Exercício



1) Criar um elemento para conter os dois painéis. Ele deve receber a classe container, para se alinhar ao meio da tela, e a classe paineis que usaremos depois no CSS.

```
<div class="container paineis"> <!-- os paineis de pastas com aulas --> </div>
```

2) Dentro da div criada acima, crie uma nova <section> para cada painel. A primeira, receberá as classes **painel** e **aula1**. Cada exercicio deve ser representado como um item na lista () com um link para o exercício e sua imagem (representado por figure, figcaption e img).





# Exercício



3) Criar um segundo painel, para representar as segunda aula. Esse painel deve ficar após o fechamento do painel anterior, mas ainda dentro da div paineis.

O novo painel deve receber as classes painel e aula2. Sua estrutura é idêntica ao do exercício anterior (dica: copie o código para evitar refazer tudo de novo).







# Exercício



4) Posicionar os painéis no css.

```
.painel { margin: 10px 0; padding: 10px; width: 445px; }
.aula1 { float: left; background-color: #f5dcdc; }
.aula2 { float: right; background-color: #dcdcf5; }
.painel li { display: inline-block; vertical-align: top; width: 140px; margin: 2px; padding-bottom: 10px; }
.painel h2 { font-size: 24px; font-weight: bold; text-transform: uppercase; margin-bottom: 10px; }
.painel a { color: #333; font-size: 14px; text-align: center; text-decoration: none; }
```







# Seletores de atributo do CSS3



Além dos seletores de tag, classe e id que vimos anteriormente, existe mais uma série de seletores avançados do CSS.

Um dos seletores avançados do CSS é o seletor de atributo, com ele é verificado a presença ou valor de um atributo para selecioná-lo. Por exemplo:

```
input[value] { color: #cc0000; }
```

O seletor acima age em todos os elementos da tag <input> que têm o atributo "value".

Também é possível verificar se o atributo tem um valor específico: input[type="text"] { border-radius: 4px; }







# Seletores de atributo do CSS3



Além de verificar um valor, é possível utilizar alguns operadores para selecionar valores em determinadas condições, como por exemplo o seletor de atributo com prefixo:

```
div[class|="menu"] { border-radius: 4px; }
```

O seletor acima vai agir em todas as tags <div> cujo atributo "class" comece com a palavra menu seguida de um hífen e qualquer outro valor na sequência, como por exemplo menu-opções, menu-conteudo etc.

Também é possível buscar por uma palavra específica no valor, não importando o valor completo do atributo. Por exemplo:

```
input[value~="problema"] { color: #cc0000; }
```







# Seletores de atributo do CSS3



Com o CSS3 é possível utilizar novos operadores com sinais que se assemelham aos das expressões regulares:

```
/* busca por inputs com valor de "name" iniciando em "usuario" */
input[name^="usuario"] { color: 99ffcc;}

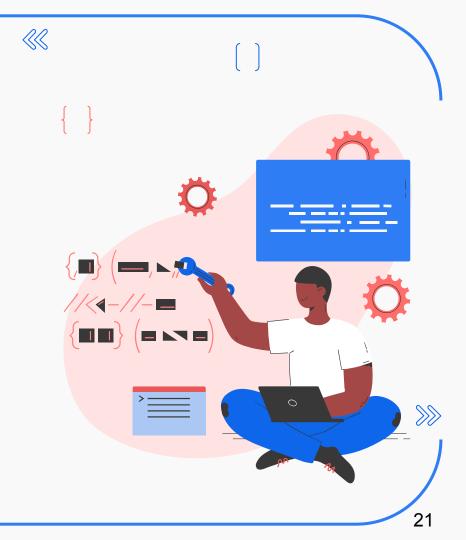
/* busca por inputs com valor de "name" terminando em "teste" */
input[name$="teste"] { background-color: #ccff00; }

/* busca por inputs com valor do atributo "name" contendo "tela" em qualquer posição */
input[name*="tela"] { color: #666666; }
```





# Rodapé





# Rodapé



No HTML5, a tag apropriada para rodapés é a <footer>.

O rodapé é utilizado para colocar informações do site, copyrights, redes sociais e imagens.

Vamos colocar 2 conteúdos: o logo em negativo do lado esquerdo e ícones de acesso a redes sociais do lado direito.

O logo no lado esquerdo é uma simples imagem:

```
<img src="img/logo-rodape.png" alt="Logo site">
```

Para as redes sociais utilize lista:

```
  \\ < a \ href="http://facebook.com/">Facebook</a> < a \ href="http://twitter.com/">Twitter</a>
```



Substituição por imagem





# Substituição por imagem



Um truque usado em CSS é chamado **Image Replacement** (substituição por imagem).

### A ideia básica é:

- Acertar o tamanho do elemento para ficar igual ao da imagem;
- Colocar a imagem como background do elemento;
- Esconder o texto.

Para esconder o texto, é utilizado um text-indent negativo bastante alto. Isso, na prática, faz o texto ser renderizado "fora da tela"

.facebook { /\* tamanho do elemento = imagem \*/ height: 55px; width: 85px; /\* imagem como fundo \*/ background-image: url(../img/facebook.png); /\* retirando o texto da frente \*/ text-indent: -9999px; }





# Estilização e posicionamento do Rodapé





# Estilização e posicionamento Footer



### Container interno.

O rodapé ocupa 100% da largura da página e não é restrito ao tamanho limitado do miolo do site.

Mas o conteúdo dele é limitado e centralizado junto com o resto da página onde estávamos usando a class container.

O que precisamos fazer então é ter o <footer> com 100% além de uma tag interna, essa tag será o container:







# Estilização e posicionamento Footer



### **Posicionamento**

Ao adicionar o rodapé, ele subirá na página ao invés de ficar em baixo.

Isso porque os painéis de destaque, estão flutuando na página e, portanto, saíram do fluxo de renderização.

Para corrigir isso, basta usar a propriedade clear: both no CSS do footer.

Dentro do rodapé em si teremos a lista de ícones à direita, para isso será definido o posicionamento absoluto, desde que o container do rodapé esteja posicionado (basta dar um position:relative a ele).

Já os itens dentro da lista (os 3 links), devem ser flutuados lado a lado (e não um em cima do outro). É fácil fazer com float:left no li.







# Estilização e posicionamento Footer



### Estilização

O rodapé si terá um background-image repetido infinitamente.

Os elementos internos são todos ícones a serem substituídos por imagens via CSS com image replacement.

Para saber qual ícone atribuir a qual link da lista de mídias sociais, serã utilizado os seletores de atributo do CSS3:

```
.social a[href*="facebook.com"] {
          background-image: url(../img/facebook.png);
}
```





# Exercícios





# **Exercícios**



- Aplique um rodapé do layout. Crie uma estrutura semântica no HTML usando a tag <footer> e tags <img>, , e <a> para o conteúdo. Atenção especial para a necessidade de um elemento container dentro do rodapé para alinhar seu conteúdo com o restante da página.
- 2) Coloque o background no rodapé e faça as substituições de imagens. Use seletores de atributo do CSS3 para identificar os ícones de cada rede social.
- 3) Ajuste a flutuação do rodapé.
- 4) Posicione os elementos internos do rodapé apropriadamente.











Para rodar JavaScript em uma página Web, precisamos ter em mente que a execução do código é instantânea.

Para inserirmos um código JavaScript em uma página, é necessário utilizar a tag <script>:

```
<script>
alert("Olá, Mundo!");
</script>
```

O exemplo acima é um "hello world" em JavaScript utilizando uma função do navegador, a função alert.

É possível adicionar essa tag em qualquer local do documento que a sua renderização ficará suspensa até o término dessa execução.









Arquivo Externo.

No arquivo HTML

<script src="scripts/hello.js"></script>

Arquivo externo script/hello.js

alert("Olá, Mundo!");

Com a separação do script em arquivo externo é possível reaproveitar alguma funcionalidade em mais de uma página.









Arquivo Externo.

No arquivo HTML

<script src="scripts/hello.js"></script>

Arquivo externo script/hello.js

alert("Olá, Mundo!");

Com a separação do script em arquivo externo é possível reaproveitar alguma funcionalidade em mais de uma página.









### Console no navegador

Alguns navegadores dão suporte a entrada de comandos pelo console.

Por exemplo:

No Google Chrome o console pode ser acessado por meio do atalho Control + Shift + C; No Firefox, pelo atalho Control + Shift + K.

Experimente executar um alert no console e veja o resultado:

alert("interagindo com o console!");









### Sintaxe básica

Operações matemáticas. Teste algumas contas digitando diretamente no console:

25

> 14 \* 3

42

> 10 - 4

6

> 25 / 5

5

> 23 % 2

1









### Variáveis

Para armazenar um valor para uso posterior, podemos criar uma variável:

```
var curso = "FWeb-23";
alert(curso);
```

No exemplo acima, foi guardo o valor FWeb-23 na variável curso. A partir desse ponto, é possível utilizar a variável para obter o valor que guardamos nela.









### **Tipos**

O JavaScript dá suporte aos tipos **String** (letras e palavras), **Number** (números inteiros, decimais), **Boolean** (verdadeiro ou falso) entre outros.

```
var texto = "Uma String deve ser envolvida em aspas simples ou duplas.";
var numero = 2012;
var verdade = true;
```

Outro tipo de informação que é considerado um tipo no JavaScript é o **Array**, onde podemos armazenar uma série de informações de tipos diferentes:

```
var pessoas = ["João", "José", "Maria", "Sebastião", "Antônio"]; (\{((\{ \gg \})) \ll \})
```







**Tipos - Array** 

O JavaScript é utilizado para interagir com os elementos da página. Para fazer alguma coisa com cada elemento de uma coleção é necessário efetuar uma iteração. A mais comum é o **for**:

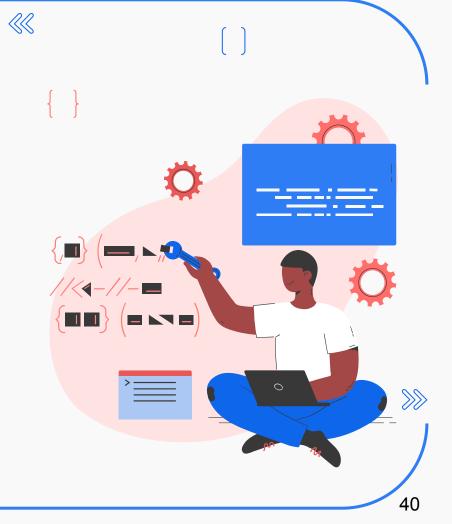
```
var pessoas = ["João", "José", "Maria", "Sebastião", "Antônio"];
for (var i = 0; i < pessoas.length; i++) {
        alert(pessoas[i]);
}</pre>
```

alert(pessoas[0]);
alert(pessoas[1]);
alert(pessoas[4]);





# Exercícios





# **Exercícios**

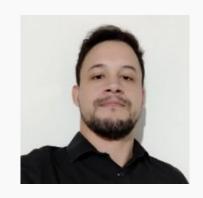


- 1) Escreva um código que mostre os números ímpares entre 1 e 10.
- 2) Escreva um código que calcule a soma de 1 até 100. (obs: a resposta é 5050)
- 3) Crie um Array igual ao abaixo e mostre apenas os nomes das pessoas que tenham 4 letras. var pessoas = ["João", "José", "Maria", "Sebastião", "Antônio"]; Dica: use o atributo length das Strings.
- 4) Escreva um código que leia o nome de uma pessoa e mostre um alerta de boas-vindas.





# **Professor**



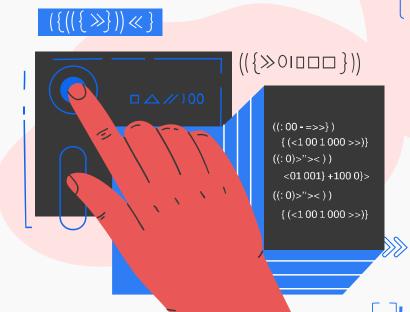
Lucas G. F. Alves





# Obrigado!

E-mail :lucas.g.f.alves@gmail.com



**>>>>** 



