Interatividade na Web

Front End Web

Msc. Lucas G. F. Alves
e-mail: lucas.g.f.alves@gmail.com

Senac

Planejamento de Aula

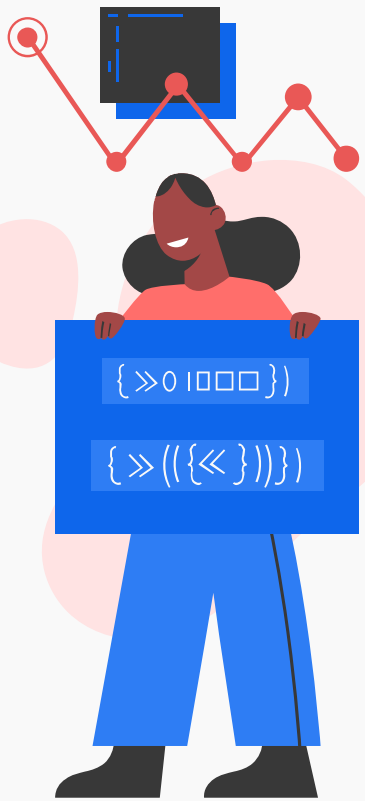
Revisão

Introdução a Javascript

Interatividade na Web

Funções Temporais

Exercícios



Revisão

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:

```
<ul class="social">
    <li><a href="http://facebook.com/">Facebook</a></li>
    <li><a href="http://twitter.com/">Twitter</a></li>
    <li><a href="http://plus.google.com/">Google+</a></li>
</ul>
```

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 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:

```
<footer>
        <div class="container">
        ....
        </div>
</footer>
```

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

1) Aplique um rodapé do layout. Crie uma estrutura semântica no HTML usando a tag <footer> e tags <img>, <ul>, <li> 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.

Introdução ao Javascript

Introdução ao Javascript

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.

Introdução ao Javascript

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.

Introdução ao Javascript

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.

Introdução ao Javascript

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!");
```

Introdução ao Javascript

Sintaxe básica

Operações matemáticas. Teste algumas contas digitando diretamente no console:

```
> 12 + 13
25
> 14 * 3
42
> 10 - 4
6
> 25 / 5
5
> 23 % 2
1
```

Introdução ao Javascript

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.

Introdução ao Javascript

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"];
```

Introdução ao Javascript

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]);
}
```

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

Interatividade na Web

Interatividade na Web

O JavaScript é a principal linguagem de programação da Web.
Com essa linguagem é possível encontrar um elemento da página e alterar características dele pelo código JavaScript:

```
var titulo = document.querySelector("#titulo");
titulo.textContent = "Agora o texto do elemento mudou!";
```

No exemplo é selecionado um elemento do documento e alterado sua propriedade textContent.Também é possível selecionar elementos de maneira similar no CSS, através de seletores CSS:

```
var painelNovidades = document.querySelector("section#main .painel#novidades");
painelNovidades.style.color = "red"
```

Interatividade na Web

querySelector vs querySelectorAll

A função querySelector sempre retorna um elemento, mesmo que o seletor potencialmente traga mais de um elemento, neste caso, apenas o primeiro elemento da seleção será retornado.
A função querySelectorAll retorna uma lista de elementos compatíveis com o seletor CSS passado como argumento. Sendo assim, para acessarmos cada elemento retornado, precisaremos passar o seu índice conforme o exemplo abaixo:

```
var paragrafos = document.querySelectorAll("div p");
paragrafos[0].textContent = "Primeiro parágrafo da seleção";
paragrafos[1].textContent = "Segundo parágrafo da seleção";
```

Interatividade na Web

As alterações serão feitas quando o usuário executa alguma ação, como por exemplo, clicar em um elemento.
Para fazer isso é necessário utilizar de dois recursos do JavaScript no navegador.
O primeiro é a criação de **funções**:

```
function mostraAlerta() {
    alert("Funciona!");
}
```

Esse código é guardado e só será executado quando chamarmos a função. Para isso será utilizado o segundo recurso, os **eventos:**

```
// obtendo um elemento através de um seletor de ID
var titulo = document.querySelector("#titulo");
titulo.onclick = mostraAlerta;
```

Interatividade na Web

As alterações serão feitas quando o usuário executa alguma ação, como por exemplo, clicar em um elemento.
Para fazer isso é necessário utilizar de dois recursos do JavaScript no navegador.
O primeiro é a criação de **funções**:

```
function mostraAlerta() {
    alert("Funciona!");
}
```

Esse código é guardado e só será executado quando chamarmos a função. Para isso será utilizado o segundo recurso, os **eventos:**

```
// obtendo um elemento através de um seletor de ID
var titulo = document.querySelector("#titulo");
titulo.onclick = mostraAlerta;
```

Interatividade na Web

Chamar a função sem evento:

```
// Chamando a função:
mostraAlerta();
```

A função vai ter algum valor variável que será definido quando executá-la:

```
function mostraAlerta(texto) {
    // Dentro da função  a variável "texto" conterá o valor passado na execução.
    alert(texto);
}
// Ao chamar a função é necessário definir o valor do "texto"
mostraAlerta("Funciona com argumento!");
```

Interatividade na Web

Eventos:
onclick: clica com o mouse; ondblclick: clica duas vezes com o mouse;
onmousemove: mexe o mouse; onmousedown: aperta o botão do mouse;
onmouseup: solta o botão do mouse (útil com os dois acima para gerenciar drag'n'drop);
onkeypress: ao pressionar e soltar uma tecla; onkeydown: ao pressionar uma tecla;
onkeyup: ao soltar uma tecla. Mesmo acima; onblur: quando um elemento perde foco;
onfocus: quando um elemento ganha foco;
onchange: quando um input, select ou textarea tem seu valor alterado;
onload: quando a página é carregada; onunload: quando a página é fechada;
onsubmit: disparado antes de submeter o formulário. Útil para realizar validações.

Exercícios

Exercícios

1) Utilizar o Google como mecanismo de busca apenas como ilustração. Para isso, basta fazer o formulário submeter a busca para a página do Google. No <form>, acrescente um atributo action= apontando para a página do Google:

```
<form action="http://www.google.com.br/search" id="form-busca">
<input type="search" name="q" id="q">
```

2) Validar quando o usuário clicar em submeter, verificar se o valor foi preenchido. Se estiver em branco, mostrar uma mensagem de erro em um alert. A validação será escrita em JavaScript. Portanto, crie o arquivo index.js na pasta js/ do projeto. Referencie esse arquivo no index.html usando a tag <script> no final da página, logo antes de fechar o </body>:

```
<body>
    ....
    <script src="js/home.js"></script>
</body>
```

Exercícios

3) Criar a função que verifica se o elemento com id=q (o campo de busca) está vazio. Se estiver, mostramos um alert e abortamos a submissão do formulário com return false:

```
function validaBusca() {
    if (document.querySelector('#q').value == '') {
        alert('Não podia ter deixado em branco a busca!');
        return false;
    }
}
```

4) Indicar que a função anterior deve ser executada quando o usuário disparar o evento de enviar o formulário (onsubmit). Coloque no final do arquivo JavaScript:

```
// fazendo a associação da função com o evento
document.querySelector('#form-busca').onsubmit = validaBusca;
```

Funções Temporais

Funções temporais

No JavaScript, se pode criar um timer para executar um trecho de código após um certo tempo, ou ainda executar algo de tempos em tempos.

A função **setTimeout** permite que agende alguma função para execução no futuro e recebe o nome da função a ser executada e o número de milissegundos a esperar:

```
// executa a minhaFuncao daqui um segundo
setTimeout(minhaFuncao, 1000);
```

Se for um código recorrente, pode-se usar o **setInterval** que recebe os mesmos argumentos mas executa a função indefinidamente de tempos em tempos:

```
// executa a minhaFuncao de um em um segundo
setInterval(minhaFuncao, 1000);
```

Funções temporais

clearInterval

É utilizado para cancelar a execução no futuro. É especialmente interessante para o caso do interval que pode ser cancelado de sua execução infinita:

```
// agenda uma execução qualquer
var timer = setInterval(minhaFuncao, 1000);

// cancela execução
clearInterval(timer);
```

Exercícios

Exercícios

1) Implementar um banner rotativo na Pagina Principal usando JavaScript. Colocar duas imagens, a destaque-home.png e a destaque-home-2.png para trocar a cada 4 segundos;

```
var banners = ["img/destaque-home.png", "img/destaque-home-2.png"];
var bannerAtual = 0;
function trocaBanner() {
    bannerAtual = (bannerAtual + 1) % 2;
    document.querySelector('.destaque img').src = banners[bannerAtual];
}
setInterval(trocaBanner, 4000);
```

```
<div class="container destaque">
    ...
    <img src="./img/logo.png" alt="Imagem de destaque">
</div
```

Exercícios

2) Faça um botão de pause que pare a troca do banner.
Dica: use o clearInterval para interromper a execução.

```
<a href=”#” class=”pause”></a>
var controle = document.querySelector('.pause');
controle.onclick = function() { if (controle.className == 'pause') { clearInterval(timer);
controle.className = 'play'; } else { timer = setInterval(trocaBanner, 4000);
controle.className = 'pause'; } return false; };
```

3) Faça um botão de play para reativar a troca dos banners.

4) Dentro da página Blog colocar as tags de estrutura e desenvolver uma calculadora.

({(({≫})) ≪ }

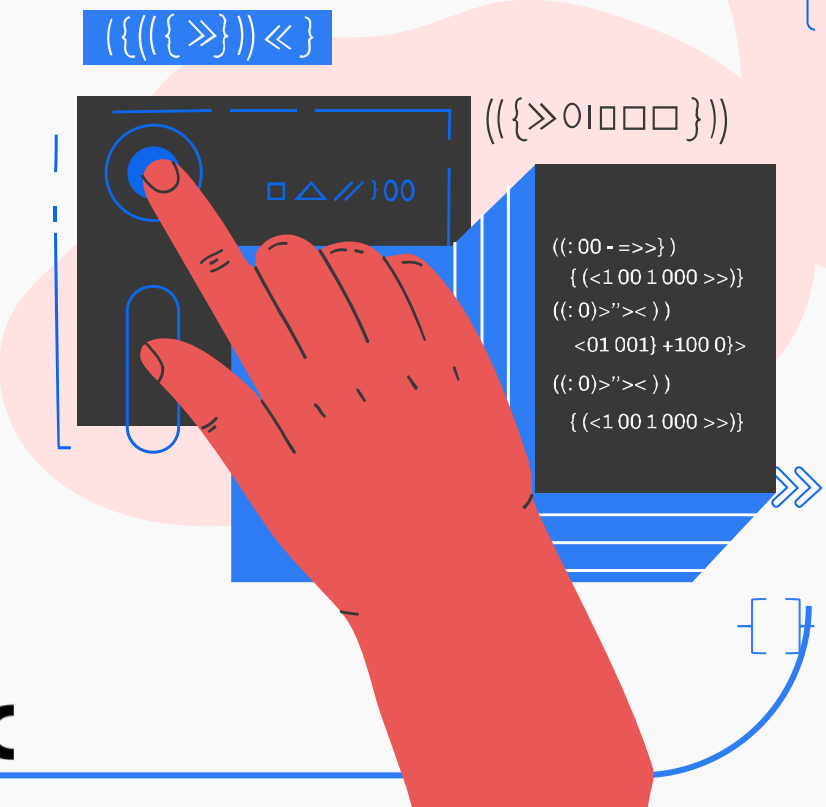
Professor

Lucas G. F. Alves

Senac

Obrigado!

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



Senac