Curso Técnico em Desenvolvimento de Sistemas Online

DESENVOLVIMENTO WEB

GEEaD - Grupo de Estudo de Educação a Distância

Centro de Educação Tecnológica Paula Souza

Expediente

GEEaD – CETEC
GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS
FUNDAMENTOS DE INFORMÁTICA

Autor:
Paulo Eduardo Cardoso Andrade

Revisão Técnica:
Eliana Cristina Nogueira Barion
Lilian Aparecida Bertini

Revisão Gramatical:
Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:
Flávio Biazim

São Paulo – SP, 2019

APRESENTAÇÃO

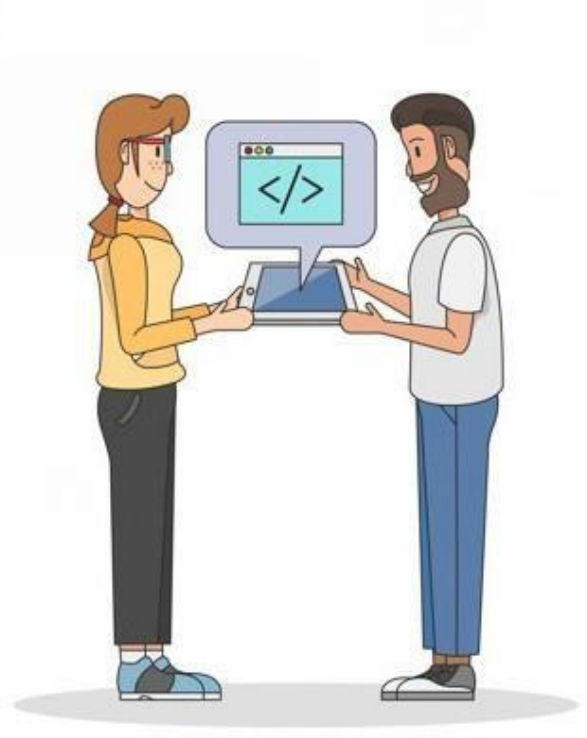
Este material didático do Curso Técnico em Desenvolvimento de Sistemas modalidade EaD foi elaborado especialmente por professores do Centro Paula Souza para as Escolas Técnicas Estaduais – ETECs.

O material foi elaborado para servir de apoio aos estudosdos discentes para que estes atinjam as competências e as habilidades profissionais necessárias para a sua plena formação como Técnicos em Desenvolvimento de Sistemas.

Esperamos que este livro possa contribuir para uma melhorformaçãoe apefeiçoamento dos futuros Técnicos.

AGENDA 16

JAVASCRIPT - UMA LINGUAGEM DE PROGRAMAÇÃO!



MERGULHANDO NO TEMA...

O que é JavaScript?

Como já vimos, no início das páginas de internet, tínhamos pouco ou nenhuma interatividade, basicamente eram páginas que expunham seu conteúdo. Navegar por meio de **links** e enviar informações por meio de formulários, resumidamente, era tudo que o usuário podia fazer. Com o grande potencial da Internet e a necessidade de maiores interações entre usuários e páginas, a empresa **Netscape**, criadora do navegador mais popular da época, criou o **LiveScript**, uma linguagem simples, que permitia a execução de scripts contidos nas páginas dentro do próprio navegador.

Mas porque tornou-se **JavaScript**? Foi apenas uma jogada de marketing! Aproveitando o sucesso da Linguagem de Programação Java, que vinha conquistando cada vez mais espaço no mercado de desenvolvimento de aplicações corporativas, a Netscape logo rebatizou o **LiveScript** como **JavaScript**. A Microsoft, então vice-líder de navegadores, adicionou ao Internet Explorer o suporte a scripts escritos em **VBScript** e criou sua própria versão de **JavaScript**, o **JScript**.

Podemos definir que o **JavaScript** é uma linguagem de programação muito popular no desenvolvimento de páginas web e suportada por praticamente todos navegadores. Sendo assim, é importante que o programador **Web** conheça essas três linguagens:

1. **HTML** para definir o conteúdo de páginas da web;
2. **CSS** para especificar o layout das páginas da web;
3. **JavaScript** para programar o comportamento de páginas da web.

Como utilizar

Da mesma forma que o **CSS**, existem três maneiras de incluir códigos **JavaScript** em um site:

- **Inline** (em linha);
- **Interno**;
- **Externo**.

Vamos programar? Assim ficará mais fácil de entender como utilizá-las!

Inline

Crie um arquivo **HTML** e salve com o nome de "**jsInline**". Não se esqueça de criar a estrutura básica. Como pode notar, pelo nome dado ao arquivo, iremos utilizar códigos **Inline** para o primeiro exemplo. O objetivo é inserir um texto e um botão na página para que quando o usuário clicar no botão, o texto seja alterado. Vamos ao código:

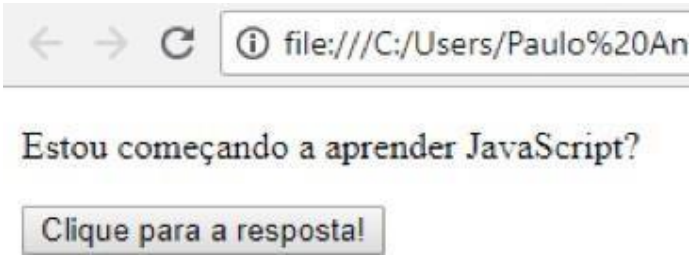
```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title> Aula Java Script </title>
</head>
<body>

    <p id="teste">Estou começando a aprender JavaScript?</p>
    <button type="button" onclick="document.getElementById('teste').innerHTML =
'SIM!'">Clique para a resposta!</button>
</body>
</html>
```

Quase tudo nesse código você já conhece, mesmo assim vale relembrar:

- Na tag **p** atribuímos um id único **(teste)** para que, por meio do **JavaScript**, seja possível identificar esse elemento e alterá-lo. Já na tag **Button** teremos o **onclick** (um Evento que executa o script quando o botão for clicado) onde escrevemos o nosso código **JavaScript**.

O primeiro código **JavaScript** consegue acesso a tag **p**, identificada de forma única no documento por meio do atributo **id** e, utilizando a propriedade **innerHtml**, substitui o texto "**Estou começando a aprender JavaScript?**" por "**SIM**".

O resultado no navegador será:



Após o usuário clicar no botão será apresentado o seguinte resultado:

Perceba que o valor atribuído à propriedade **innerHTML**, foi inserido no elemento marcado pelo **id** teste.

Na página 20, ao final desta agenda, você encontra o download completo de todos os exemplos.

Interno

Para o segundo exemplo, vamos utilizar **JavaScript** interno, para isso temos a tag **script** que pode ser usada com **JavaScript** dentro das tags **head**, **body** ou em ambas. Não há um limite para a quantidade de **scripts** utilizados dentro de uma mesma página **HTML**.

Então, vamos lá!!!

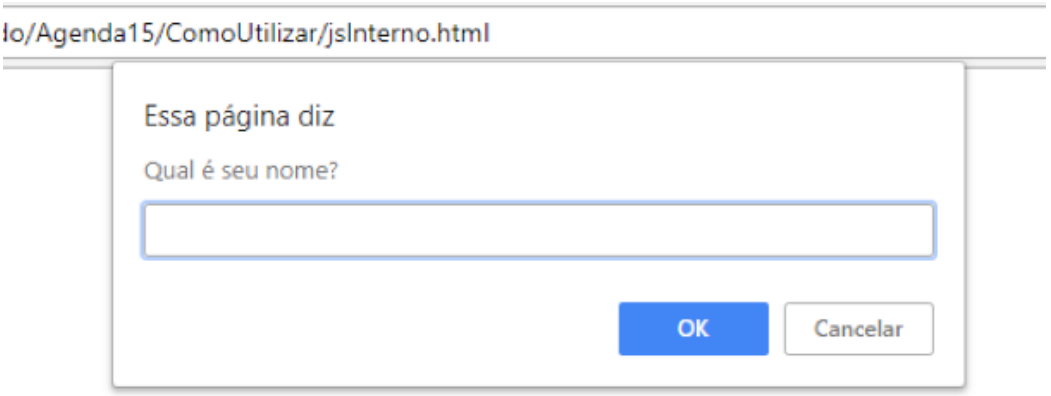
Crie um arquivo **HTML** e o salve com o nome de "**jsInterno**". Não se esqueça de criar a estrutura básica! Nesse exemplo, vamos pedir ao usuário para inserir seu nome em uma caixa de diálogo e a página responderá com uma saudação de boas-vindas. Vamos codificar!

```
<body>
    <script>
        var nome;
        nome = prompt("Qual é seu nome?");
        alert("Olá, " + nome+"! Seja Bem Vindo!");
    </script>
</body>
```

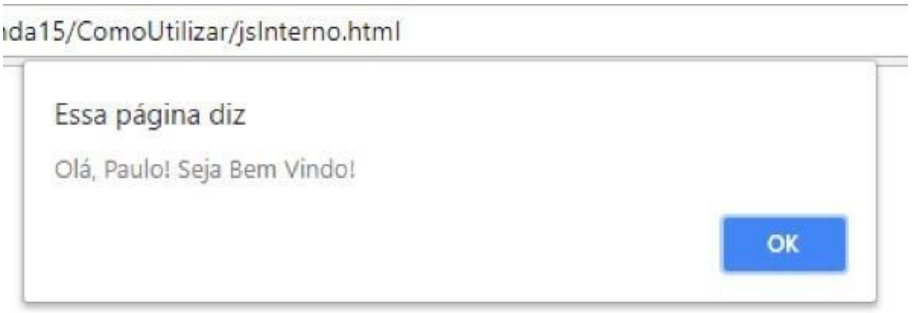
Note que nessa codificação declaramos uma **variável** denominada "**nome**", e que a palavra reservada **var** define que a próxima palavra será uma **variável**.

Os tipos de dados das variáveis não precisam ser declarados (tipagem dinâmica), ou seja, o tipo é definido assim que o **valor** é atribuído. A função **promp** envia uma mensagem para o usuário e oferece um **campo** para ser inserido, como um texto ou valor numérico, por exemplo. Neste exemplo, o **valor** está sendo atribuído na variável denominada "**nome**". A função **alert**, por sua vez, envia uma caixa de diálogo com uma mensagem de boas-vindas concatenada com o nome inserido pelo usuário.

Assim que a página é carregada pelo navegador, temos o seguinte resultado:



Agora basta inserir o nome e clicar em **OK** e obtem-se o resultado como demonstrado na imagem a seguir.



Na página 20, ao final desta agenda, você encontra o download completo de todos os exemplos.

Externo

Por fim, o terceiro exemplo: o **JavaScript Externo**. Observe que a tag **script** pode ser usada em diversos lugares dentro do código, por exemplo, na tag **head**, body ou em ambas, a diferença para os outros exemplos é que temos um atributo **src** (o mesmo usado para indicar a localização de uma imagem na tag **img**) indicando a localização do **script**

```
<script src="arquivo.js"></script>
```

Então, crie mais um arquivo **HTML** e salve com o nome de "**jsExterno**". Não se esqueça de criar a estrutura básica e um arquivo **JavaScript** com o nome **javas** que deverá ter a extensão **js**. Nesse exemplo, vamos escrever todos os números inteiros de **0** a **100** na página. Vamos codificar!

O Primeiro passo é adicionar, no arquivo **jsExterno.html**, o script que está no arquivo **javas.js**, como no código a seguir:

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title> Aula Java Script </title>
    <script src="javas.js"></script>
</head>
```

Agora no arquivo **javas.js**, codifique:

```
var i;
for (i = 0; i <= 100; i++) {
        document.write(i + " <br>");
}
```

Perceba que criamos uma **variável** denominada **i** e, utilizando a estrutura de repetição **for**, criamos um laço de repetição que será executado **101** vezes.

Neste laço de repetição, o código: **document.write(i+"
")**; será executado todas as **101** vezes, sua função é escrever no arquivo **html**, não somente textos para usuário; mas, também, podem ser inseridas ou "escritas" tags para que o navegador interprete. Nesse exemplo, inserimos uma tag **br** para que pule uma linha após cada número exibido.

O resultado será como o apresentado a seguir.

Obs.: O **JavaScript** é uma linguagem de programação que oferece diversos recursos como estruturas de repetição (**for**, **while**, **foreach** etc), estruturas condicionais (**if**..**else**) e etc.

Comentário

Como em **HTML** e **CSS**, o **JavaScript** oferece a opção de colocar comentários, apenas relembrando, como o próprio nome define, que são notas, informações ou observações que podem ser incluídas no código fonte para descrever o que se quiser. Você pode adicionar comentários em JavaScript, usando a seguinte sintaxe:

• Comentário em linha - basta adicionar "**//**" antes de escrever o comentário.

- // Comentário

• Comentário com múltiplas linhas - tudo que está entre **/*** e ***/** será um comentário.

- /*
- Comentário.
- */

Na página 20, ao final desta agenda, você encontra o download completo de todos os exemplos.

SlideShow

Existem várias maneiras para a criação de slideshow e este recurso é praticamente utilizado em todos os sites. Vamos entender como criar um slideshow automático e manual. Veja:

Automático

Na página 20, ao final desta agenda, você encontra o download completo de todos os exemplos.

Crie mais um arquivo **HTML** e salve com o nome de "**automaticoSlideShow**". Não se esqueça de criar a estrutura básica. Escolha 4 imagens de sua preferência, crie uma pasta denominada "**imagens**" e salve as 4 imagens previamente selecionadas nesta pasta. Neste exemplo será utilizado o framework **W3.CSS**. Não se esqueça de codificar o link na tag<**head**>.

Após as configurações iniciais, codificaremos as quatro imagens para que sejam exibidas de acordo com o padrão a seguir:

```
<h2 class="w3-center">Slideshow Automatico</h2>
  <div class="w3-content w3-section w3-center" style="max-width:500px">
    <img class="slide" src="imagens/1.jpg" style="width:100%">
    <img class="slide" src="imagens/2.jpg" style="width:100%">
    <img class="slide" src="imagens/3.jpg" style="width:100%">
    <img class="slide" src="imagens/4.jpg" style="width:100%">
    <a href="http://www.freepik.com">Designed by fanjianhua / Freepik - Imagem 1</a>
    <br>
    <a href="http://www.freepik.com">Designed by evening_tao / Freepik - Imagem 2</a>
```

```
    <br>
    <a href="http://www.freepik.com">Designed by Molostock / Freepik - Imagem 3</a>
    <br>
    <a href="http://www.freepik.com">Designed by evening_tao / Freepik - Imagem 4</a>
    <br>
  </div>
```

Em um primeiro momento, por meio da tag **h2**, colocamos um subtítulo indicando o **slideshow** e, logo após, criamos uma divisão com três classes (**w3-content**, **w3-section**, **w3-center**) já conhecidas do **framework** utilizado nas agendas anteriores.

Note que a cada imagem, é configurada uma classe denominada "**slide**", essa classe será importante para o código em **javascript**.

O resultado até o momento é como demonstrado na figura a seguir.

Obs: Como as imagens não são autorais é sempre importante colocar suas referências.

Agora devemos programar o código em **JavaScript** logo abaixo da **div**. Lembre-se de criar a tag **script** para ser possível programar em **Javascript** e então codifique.

```
<script>
  var index = 0;
  carousel();

  function carousel() {
    var i;
    var x = document.getElementsByClassName("slide");
    for (i = 0; i < x.length; i++) {
      x[i].style.display = "none";
    }
    index++;
    if (index > x.length) { index = 1 }
    x[index - 1].style.display = "block";
    setTimeout(carousel, 2000); //Imagem é trocada a cada 2 segundos
  }
</script>
```

Para entender o código **javascript**, é criada uma variável chamada "**index**", e abaixo estamos chamando a função "**carousel**" que será implementada logo em seguida. Note que em **JavaScript**, toda a programação de uma função começa com a palavra chave "**function**", seguido por seu nome. Dentro da função temos o seguinte código:

```
var x = document.getElementsByClassName("slide");
```

Basicamente, ele busca todos os elementos dentro do documento que tenha a **classe "slide"** configurada e armazena sua referência na variável **x.** Em nosso caso, como há **4 elementos** configurados com essa classe, **x** se tornará um vetor, cujo **índice** será um elemento diferente a cada execução do **laço de repetição**, ou seja, primeiro índice = imagem 1 e assim por diante.

Depois de criar o vetor com as imagens que serão exibidas pelo **slideshow**, é necessário melhorar a codificação para que elas não apareçam todas juntas.

Para isso, cria-se uma estrutura de repetição **for**, cuja função será percorrer todo o vetor e definir que cada elemento (imagem) referenciado nesse vetor, não seja exibido pelo navegador, o que é implementado no código a seguir:

```
      for (i = 0; i < x.length; i++) {
        x[i].style.display = "none";
      }
```

Você se lembra da variável **index**? Cada vez que a função **carousel** for chamada, é incrementado **1** em seu valor. Para que seja possível que o **slideshow** recomece, logo após a exibição de todas as imagens, é necessário criar um desvio condicional simples que atribua (após a última imagem ser exibida), o valor **1** na variável **index**. Para isso basta verificar neste momento, se a variável **index**, está com um valor maior que o tamanho do vetor **x**.

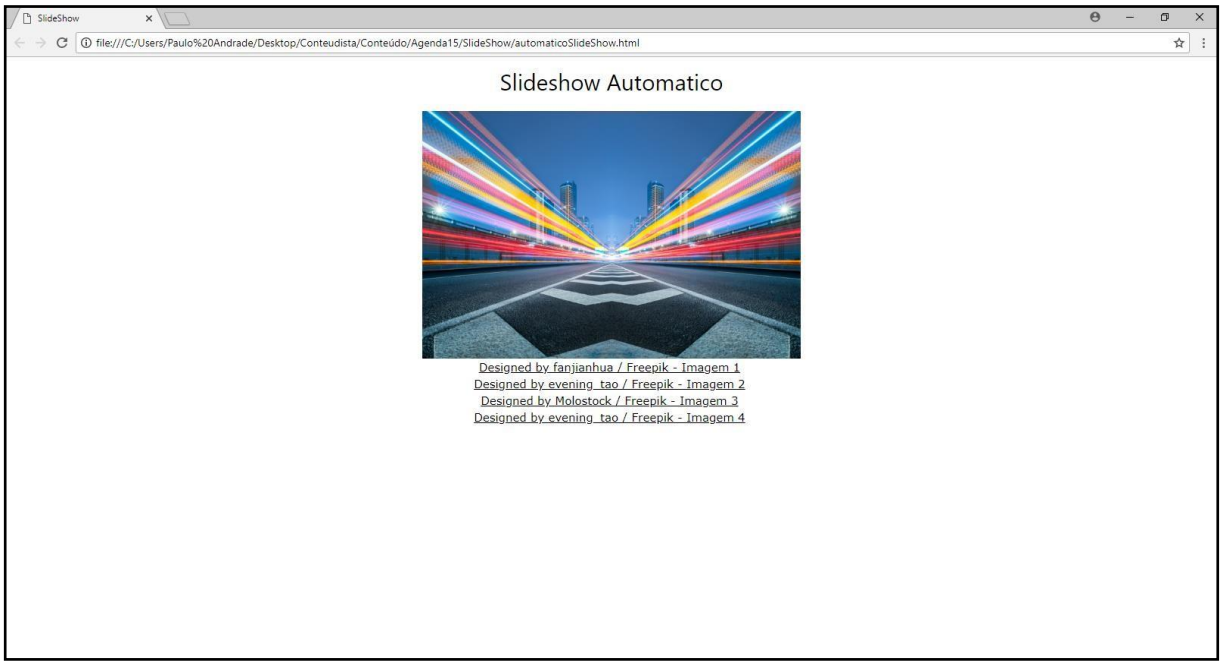
```
if (index > x.length) { index = 1 }
```

O Código a seguir faz com que a imagem referenciada por meio do índice da variável **index** seja exibida, desta forma, utilizando uma função pronta do **javascript**, denominada **setTimeout**, podemos determinar de quanto em quanto tempo a função é "**chamada**". Neste exemplo, a função é invocada a cada **2** segundos.

```
x[index - 1].style.display = "block";
setTimeout(carousel, 2000);
```

Obs: com a utilização da variável **index** para definição do índice, estamos subtraindo **1**, isso é realizado porque o primeiro índice é zero em qualquer vetor **JavaScript**.

Pronto! Após tudo isso teremos o seguinte resultado.



Obs.: A cada 2 segundos aparecerá uma das 4 imagens inseridas na página web.

Manual

Crie mais um arquivo **HTML** e salve com o nome de "**manualSlideShow**". Não se esqueça de criar a estrutura básica. Vamos utilizar as mesmas 4 imagens do exemplo anterior, para isso, certifiquem-se de que elas estão dentro de uma pasta denominada "**imagens**", no mesmo local do arquivo html. Neste exemplo, será utilizado o framework **W3.CSS**. Não se esqueça de codificar o link na tag<**head**>.

Após as configurações iniciais, vamos codificar as quatro imagens, de acordo com o seguinte padrão:

```
<h2 class="w3-center">Slideshow Manual</h2>

<div class="w3-content w3-display-container w3-center">

  <img class="slides" src="imagens/1.jpg" style="width:100%">
  <img class="slides" src="imagens/2.jpg" style="width:100%">
  <img class="slides" src="imagens/3.jpg" style="width:100%">
  <img class="slides" src="imagens/4.jpg" style="width:100%">


  <a href="http://www.freepik.com">Designed by fanjianhua / Freepik - Imagem 1</a><br>
  <a href="http://www.freepik.com">Designed by evening_tao / Freepik - Imagem 2</a><br>
  <a href="http://www.freepik.com">Designed by Molostock / Freepik - Imagem 3</a><br>
  <a href="http://www.freepik.com">Designed by evening_tao / Freepik - Imagem 4</a><br>
```

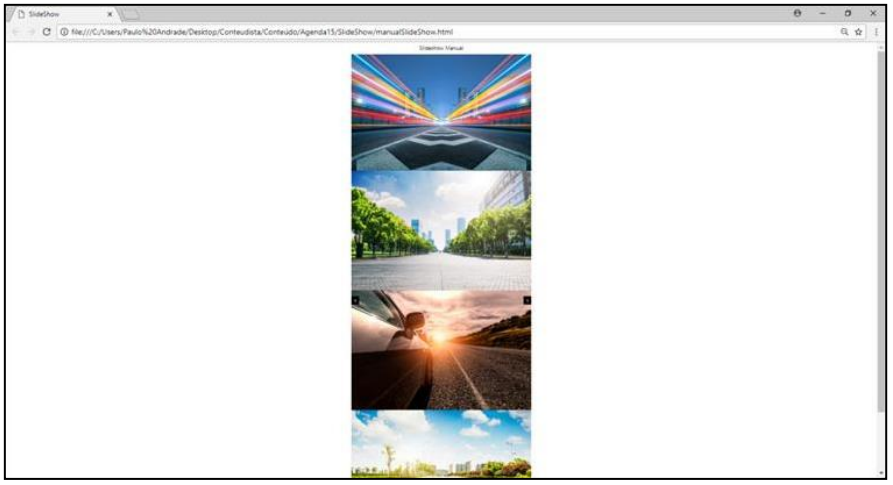
Por meio da tag **h2**, do mesmo modo que o exemplo anterior, foi inserido um subtítulo indicando o **slide show** e, em seguida, foi criada uma divisão com três **classes** (**w3-content**, **w3-display-container**, **w3-center**). Perceba que cada uma das imagens está configurada com a classe denominada **"slide"**, e lembre-se, com a utilização dessa classe será possível que o código em **JavaScript** realize as alterações necessárias na página para que o objetivo seja alcançado.

Note que foram codificados dois **botões** que terão a função de passar os **slides** para frente e para trás.

```
  <button class="w3-button w3-black w3-display-left" onclick="plusDivs(-
1)">&#10094;</button>
  <button class="w3-button w3-black w3-display-right"
onclick="plusDivs(1)">&#10095;</button>
</div>
```

Neles há algumas classes do framework e, no atributo do botão "**onclick**", é chamada uma função (**plusDivs**) que será programada em **JavaScript**.

O resultado até o momento pode ser visualizado por meio da imagem a seguir.



Agora é hora de programar o código em **JavaScript**, logo abaixo da **div**, lembre-se de criar a tag **script** e codifique.

```
<script>
var slideIndex = 1;
showDivs(slideIndex);

function plusDivs(n) {
  showDivs(slideIndex += n);
}

function showDivs(n) {
  var i;
```

```
  var x = document.getElementsByClassName("slides");
  if (n > x.length) {slideIndex = 1}
  if (n < 1) {slideIndex = x.length}
  for (i = 0; i < x.length; i++) {
     x[i].style.display = "none";
  }
  x[slideIndex-1].style.display = "block";
}
</script>
```

Entenda o código **JavaScript** para saber diferenciá-lo do exemplo anterior.

Foi criada a variável com o nome de **"slideIndex",** em seguida foi chamada a função **"showDivs"**, passando como parâmetro a variável **slideIndex**. Logo após, foi criada mais uma função com o nome de "**plusDivs**" e solicitado um valor como parâmetro.

Dentro da função temos o seguinte código:

```
showDivs(slideIndex += n);
```

Recebendo como parâmetro a variável **slideIndex** e somando ao valor de n, esse código realiza a chamada da função **showDivs**. Lembre-se, no código em html dos botões, quando a função **plusdiv** é chamada, é passado como parâmetro o valor **-1** para voltar o slide e o valor **1** para avançar ao próximo **slide**.

Veja agora a codificação da função **showDivs**. Da mesma forma que o exemplo anterior, é criado um vetor **x** para referenciar cada elemento que contenha a classe slide programada.

A diferença está nos desvios condicionais que ao atribuir o valor **1** para a variável **slideIndex**, fazem com que os slides recomecem após o último slide ter sido exibido. Também é necessário realizar um desvio condicional para que seja referenciada a última imagem quando o usuário estiver no primeiro slide e clicar no botão de voltar. Deve-se então, colocar o **valor** na variável **slideIndex** (que representa a última posição do vetor). Isso é possível por meio da seguinte codificação:

```
if (n > x.length) {slideIndex = 1}
if (n < 1) {slideIndex = x.length}
```

Obs: perceba que quando **n** se tornar menor do que ***1***, deve-se atribuir em slideIndex o tamanho do vetor, isso porque foi utilizado como índice **slideIndex-1**.

Após essa codificação, você irá obter o seguinte resultado:

Obs.: basta clicar nos botões para navegar entre os slides (imagens)

Outros Recursos

Claro que o **JavaScript** oferece muitos outros recursos, é uma linguagem incrível.

Por meio dos links apresentados a seguir, você poderá notar a grandiosidade dessa linguagem.

Neste link (https://www.w3schools.com/jsref/default.asp) é possível ver algumas referências do **JavaScript**.

Na página 20, ao final desta agenda, você encontra o download completo de todos os exemplos.

VOCÊ NO COMANDO

Com o conhecimento adquirido nessa agenda, incremente a página do casamento de Zuleicka e Ubiratam, fazendo com que as fotos sejam automaticamente alteradas, conforme o seu tempo de permanência na página.

Coloque no mínimo três imagens que devem ser direcionadas para este **slideshow**.

Dicas: utilize **slideshow** automático;

Obs: Não se esqueça de colocar a referência das fotos!

Confira abaixo se você conseguiu resolver os desafios propostos!

Antes de exibir a resposta, lembre-se, vimos muitos recursos até aqui e o **Javascript** nos oferece muito mais, por esta razão, a resolução mostrada a seguir é apenas uma de muitas possibilidades.

Clique aqui para fazer o download dos arquivos do "Você no Comando".

Código HTML / JavaScript Completo:

```
<body class="w3-center">
    <div class="w3-container" style="color: rgb(195, 165, 116); ">
        <h1>Zuleika & Ubiratam</h1>
    </div>
    <div class="w3-row-padding w3-margin-top " style=" background-color: rgb(195, 165,
116);">
        <div class="w3-third w3-margin-top">
            <img src="imagens/n.jpg" style="width:50%" class="w3-opacity w3-hover-
opacity-off">
        </div>
        <div class="w3-third">

            <br>
            <div class="w3-content w3-section w3-center" style="">
                <img class="mySlides" src="Imagens/noiva3.jpg" style="width:100%">
                <img class="mySlides" src="Imagens/pre3.jpg" style="width:100%">
                <img class="mySlides" src="Imagens/pre2.jpg" style="width:100%">
            </div>

            <script>
                var myIndex = 0;
                carousel();
```

```
                function carousel() {
                    var i;
                    var x = document.getElementsByClassName("mySlides");
                    for (i = 0; i < x.length; i++) {
                        x[i].style.display = "none";
                    }
                    myIndex++;
                    if (myIndex > x.length) { myIndex = 1 }
                    x[myIndex - 1].style.display = "block";
                    setTimeout(carousel, 2000); // Change image every 2 seconds
                }
            </script>
        </div>
        <div class="w3-third w3-margin-top w3-margin-bottom">
            <img src="imagens/na.jpg" style="width:50%" class="w3-opacity w3-hover-
opacity-off">
        </div>
    </div>
    <br>
    <div class="w3-container" style=" background-color: rgb(195, 165, 116);
color:white;">
        <h2>Pré Casamento</h2>
    </div>

    <div class="w3-row-padding w3-margin-top ">
        <div class="w3-third">
            <div class="w3-card">
                <img src="imagens/pre1.jpg" style="width:100%" class="w3-opacity w3-
hover-opacity-off">
                <div class="w3-container">
                    <h5>Linda</h5>
                </div>
            </div>
        </div>

        <div class="w3-third">
            <div class="w3-card">
                <img src="imagens/pre2.jpg" style="width:100%" class="w3-opacity w3-
hover-opacity-off">
                <div class="w3-container">
                    <h5>Casal</h5>
                </div>
            </div>
        </div>

        <div class="w3-third">
            <div class="w3-card">
                <img src="imagens/pre3.jpg" style="width:100%" class="w3-opacity w3-
hover-opacity-off">
                <div class="w3-container">
                    <h5>Amor</h5>
                </div>
            </div>
        </div>
```

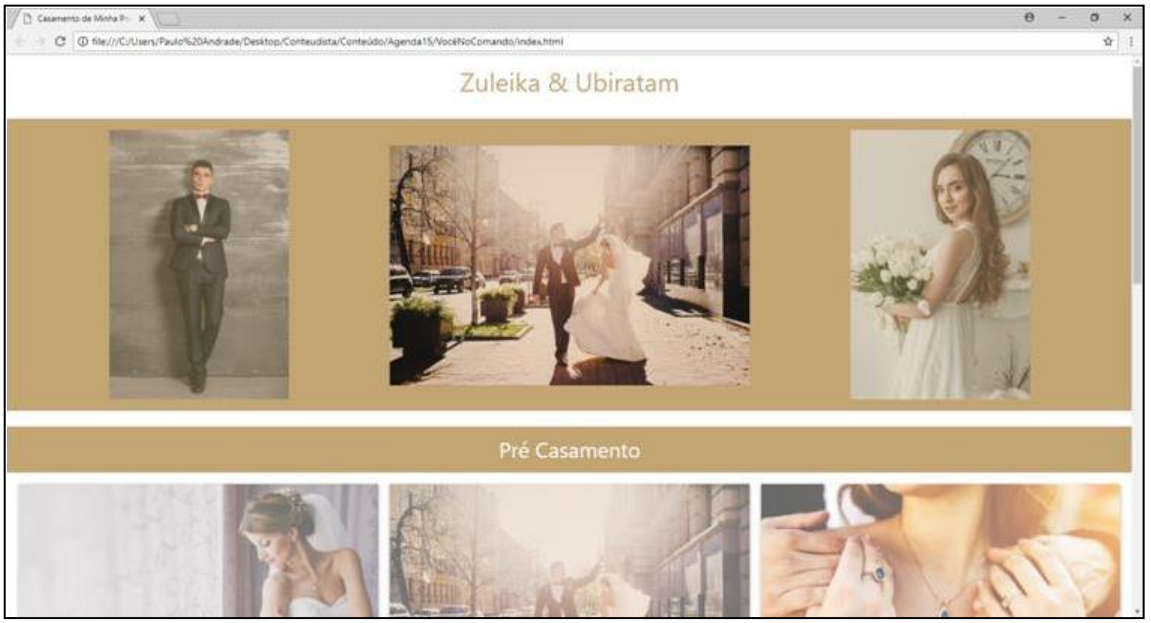
```
    </div>


    <div class=class="w3-container" style=" background-color: rgb(195, 165, 116);
color:white;">
        <h2>Dia da Noiva</h2>
    </div>
    <div class="w3-row-padding w3-margin-top">
        <div class="w3-third">
            <div class="w3-card">
                <img src="imagens/noiva1.jpg" style="width:100%" class="w3-opacity w3-
hover-opacity-off">
                <div class="w3-container">
                    <h5>Ela!</h5>
                </div>
            </div>
        </div>

        <div class="w3-third">
            <div class="w3-card">
                <img src="imagens/noiva2.jpg" style="width:100%" class="w3-opacity w3-
hover-opacity-off">
                <div class="w3-container">
                    <h5>O vestido!</h5>
                </div>
            </div>
        </div>

        <div class="w3-third">
            <div class="w3-card">
                <img src="imagens/noiva3.jpg" style="width:100%" class="w3-opacity w3-
hover-opacity-off">
                <div class="w3-container">
                    <h5>Perfeita!</h5>
                </div>
            </div>
        </div>
    </div>
    <h5>Fonte das imagens:
        <a href="https://br.freepik.com/">br.freepik.com/</a>
    </h5>
</body>
```

Código em JavaScript:

```
    <div class="w3-row-padding w3-margin-top " style=" background-color: rgb(195, 165,
116);">
        <div class="w3-third w3-margin-top">
            <img src="imagens/n.jpg" style="width:50%" class="w3-opacity w3-hover-
opacity-off">
        </div>
        <div class="w3-third">
```

```
        <br>
        <div class="w3-content w3-section w3-center" style="">
            <img class="mySlides" src="Imagens/noiva3.jpg" style="width:100%">
            <img class="mySlides" src="Imagens/pre3.jpg" style="width:100%">
            <img class="mySlides" src="Imagens/pre2.jpg" style="width:100%">
        </div>

        <script>
            var myIndex = 0;
            carousel();

            function carousel() {
                var i;
                var x = document.getElementsByClassName("mySlides");
                for (i = 0; i < x.length; i++) {
                    x[i].style.display = "none";
                }
                myIndex++;
                if (myIndex > x.length) { myIndex = 1 }
                x[myIndex - 1].style.display = "block";
                setTimeout(carousel, 2000); // Change image every 2 seconds
            }
        </script>
    </div>
```

Resultado:



Obs.: A foto do centro é trocada a cada 2 segundos.

Clique aqui para fazer o download dos arquivos do SlideShow Automático
Clique aqui para fazer o download dos arquivos do SlideShow Manual
Clique aqui para fazer o download dos arquivos de imagens do SlideShow – Parte I
Clique aqui para fazer o download dos arquivos de imagens do SlideShow – Parte II

Imagens: Crédito Freepik.com e Pixabay.com