



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*

*Autores:
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 estudos dos 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 melhor formação e aperfeiçoamento dos futuros Técnicos.

AGENDA 16

**JQUERY - UMA
BIBLIOTECA
JAVASCRIPT!**





MERGULHANDO NO TEMA...

O que é JQuery?

Em 2006, diante das dificuldades encontradas pelos programadores de **JavaScript** com o código puro, **John Resig**, um desenvolvedor web que trabalhava por conta própria, ficou incomodado com a limitação de recursos dessa linguagem, principalmente porque gostaria que seu código funcionasse em qualquer navegador. Ele decidiu então, escrever sua própria biblioteca **JavaScript** para resolver esse e outros problemas. Assim, surge, de forma resumida, o **jQuery**, uma **biblioteca** criada para oferecer diversas funcionalidades voltadas à solução dos problemas mais difíceis de serem contornados com o uso do **JavaScript puro**.



Pode-se dizer que a principal vantagem em utilizar uma **biblioteca JavaScript** é uma maior compatibilidade do seu código com diversos navegadores.

Além dessa vantagem, o **jQuery** é considerada a **biblioteca** padrão na programação **front-end** para **Web**, trazendo uma sintaxe simples para as tarefas mais comuns, por exemplo, selecionar um elemento do documento e alterar suas características.

Obs: Existem diversas outras bibliotecas **JavaScript** no mercado e a **jQuery** é uma delas.



Como utilizar

A **jQuery** funciona do mesmo jeito que o **JavaScript**, porém, na tag **head** precisamos fazer o **link** com a biblioteca por meio do seguinte código:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

Claro que também podemos realizar o **download** dessa **biblioteca** e salvar na mesma pasta do projeto, utilizando o seguinte código:

```
<script src="jquery-3.3.1.min.js"></script>
```

Obs: Link para download da biblioteca: <http://jquery.com/download/>. Ambas funcionam igualmente!



Na página 18, você encontra o download completo de todos os exemplos.

Plugins JQUERY?

Basicamente **plugins** são recursos disponibilizados pelo **jQuery** para que suas funcionalidades se tornem ainda maiores e melhores. O lema “**writeless, do more**” significa que coisas mais difíceis podem ser codificadas de forma mais fácil e de modo que possam ser utilizadas outra vez em outras **páginas, sites** ou projetos.

Atualmente existem inúmeros **plugins** e **bibliotecas** que oferecem quase tudo pronto para trabalharmos e obtermos melhores resultados. Vamos ao programa!

Primeiro Passo / Comando!

Crie um arquivo **HTML** e salve com o nome de “**jqInicio**”. Não esqueça de criar a estrutura básica e de **linkar** o **jQuery** em sua página, utilizando o código:

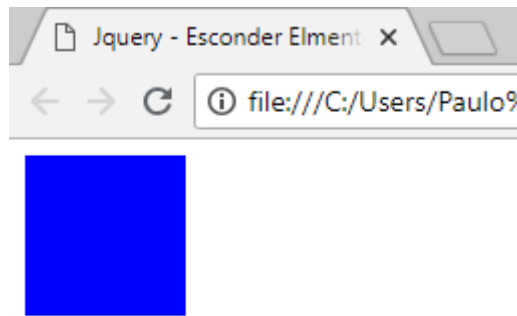
```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

Neste primeiro exemplo, vamos criar uma **div** de **80 px** de altura e **80 px** de largura e de cor azul, por meio da **CSS inline**.

Não se esqueça de definir um **id** para essa **div**. Para isso, devemos codificar da seguinte forma:

```
<body>
  <div id="div1" style="width:80px;height:80px;background-color:blue;"></div>
</body>
```

O Resultado dever ser:



A intenção deste exemplo, é fazer com que a **div** azul, em forma de quadrado (Imagem 4), desapareça da página quando o usuário clicar em cima dela, ou seja, seremos capazes de retirar um elemento da página por meio da interação do usuário. Para isso, devemos fazer a seguinte codificação dentro da tag:

```
<script>
    $(document).ready(function () {
        $("#div1").click(function () {
            $(this).hide();
        });
    });
</script>
```

O **jQuery** é feito sob medida para basicamente selecionar elementos **HTML** e/ou **CSS** e executar alguma ação. Esse exemplo deve esconder a **div** de nossa página. Vamos entender o que foi codificado:

A sintaxe é básica, por meio do caractere “\$”, indica ao navegador que vamos acessar o framework **jQuery**. Neste caso, foi definido que o código será executado no documento (**página**), apenas quando ele estiver totalmente carregado:

```
$(document).ready(
```

Logo após, criamos uma **função** que será chamada quando um evento de **click** acontecer em um elemento com o **id** “**div1**”. Veja o código utilizado para isso:

```
function () { $("#div1").click()
```

Pronto! Agora basta codificar o efeito que será aplicado após o **click**.

Por meio da função a seguir, foi definido que o **elemento** selecionado deve ser escondido, ou seja, não será exibido ao usuário pelo **navegador**.

```
function () {$(this).hide();
```

Não podemos esquecer de “fechar” as chaves e parênteses abertos! Veja outros efeitos legais disponíveis para o **jQuery**:



Na página 18, você encontra o download completo de todos os exemplos.

Eventos e Efeitos

Os eventos nos oferecem a possibilidade de executar códigos por meio das interações do usuário em uma página.

Exemplos:

- movendo um mouse sobre um elemento;
- selecionando um botão de opção;
- clicando em um elemento.

Na imagem a seguir, é possível observar alguns eventos disponíveis para a utilização em páginas **web** por meio do **jQuery**.

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

O **jQuery** permitiu na forma simples para fazer vários tipos de **efeitos** surpreendentes. Por meio de seus métodos, a biblioteca nos permite aplicar rapidamente efeitos comumente usados com uma simples e mínima codificação. Podemos listar alguns:

- | | | |
|-----------------|-----------------|----------------------|
| • .animate() | • .fadeOut() | • jQuery.fx.interval |
| • .clearQueue() | • .fadeTo() | • jQuery.fx.off |
| • .delay() | • .fadeToggle() | • jQuery.speed |
| • .dequeue() | • .finish() | • .queue() |
| • .fadeIn() | • .hide() | |

Agora mãos à obra! Vamos programar alguns exemplos.

Crie um arquivo **HTML** e salve com o nome de “**jqExemplo**”. Não esqueça de criar a estrutura básica e de **linkar** o **jQuery** em sua página, utilizando o código:


```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

Vamos criar uma animação utilizando apenas **jQuery**. Para isso, vamos precisar de um botão e de três **divs** codificadas da seguinte maneira.

```
<button>Animação</button>
  <div id="div1" style="width:80px;height:80px;background-
color:REd;position:relative;"></div>
  <br>
  <div id="div2" style="width:80px;height:80px;background-
color:blue;position:relative;"></div>
  <br>
  <div id="div3" style="width:80px;height:80px;background-
color:green;position:relative;"></div>
  <br>
```

Cada uma das **divs** está com os atributos id definidos (**div1**, **div2** e **div3**) e codificadas com **CSS Inline**. O principal atributo que você deve aplicar é **position**, deixando todas em **relative**, como demonstrado no código anterior.

Veja o resultado obtido no navegador:



Agora, vamos programar utilizando o **jQuery**. Desenvolva o código a seguir entre a **tag head**:

```
<script>
  $(document).ready(function () {
    $("button").click(function () {
      $("div").animate({
        left: '250px',
        opacity: '0.5',
        height: '150px',
        width: '150px'
      });
    });
  });
</script>
```

Note que o efeito **animate**, quando executado, levará o elemento (**div**) para esquerda em 250px, deixando-o com a opacidade 0.5 e com altura e largura de 150px.

Ao clicar no botão, observe que ocorrerá uma animação e o resultado da página ficará como demonstrado na imagem a seguir.



Veja como o código completo deverá ficar:

```
<!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">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <title>Jquery - Animar</title>

  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("div").animate({
          left: '250px',
          opacity: '0.5',
          height: '150px',
          width: '150px'
        });
      });
    });
  </script>

</head>

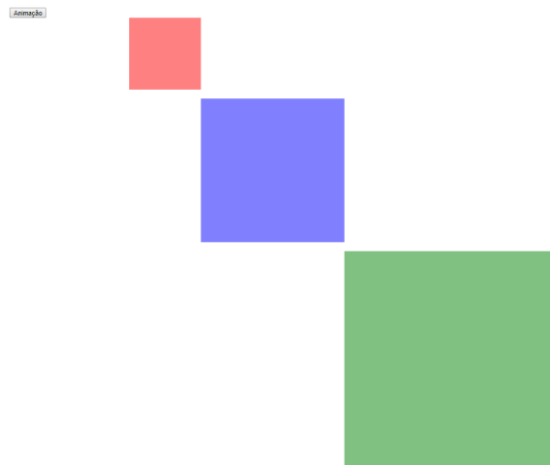
<body>
  <button>Animação</button>
```

```

<div id="div1" style="width:80px;height:80px;background-
color:REd;position:relative;"></div>
<br>
<div id="div2" style="width:80px;height:80px;background-
color:blue;position:relative;"></div>
<br>
<div id="div3" style="width:80px;height:80px;background-
color:green;position:relative;"></div>
<br>
</body>
</html>

```

Agora, utilizando o mesmo arquivo, vamos fazer uma animação exclusiva para cada uma das **divs** (quadrados), utilizando o mesmo **botão**. O objetivo é obter o seguinte resultado:



Para isso, precisamos alterar apenas o código em **JQuery**, que deve ficar da seguinte forma:

```

<script>
$(document).ready(function () {
    $("button").click(function () {
        $("#div1").animate({
            left: '250px',
            opacity: '0.5',
            height: '150px',
            width: '150px'
        });
        $("#div2").animate({
            left: '400px',
            opacity: '0.5',
            height: '300px',
            width: '300px'
        });
        $("#div3").animate({
            left: '700px',
            opacity: '0.5',
            height: '450px',
            width: '450px'
        });
    });
});

```

```

    });
  });
});
</script>

```

Note que eu posso usar a mesma **função** para criar diversas animações e como atribuímos **id's** para cada uma das **div's**, ficou fácil observar em qual **div** cada efeito acontecerá.

Vamos a mais um exemplo!

Crie outro arquivo **HTML** e salve com o nome de “**jqExemplo2**”. Não se esqueça de criar a estrutura básica e de **linkar** o **Jquery** em sua página utilizando o código:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

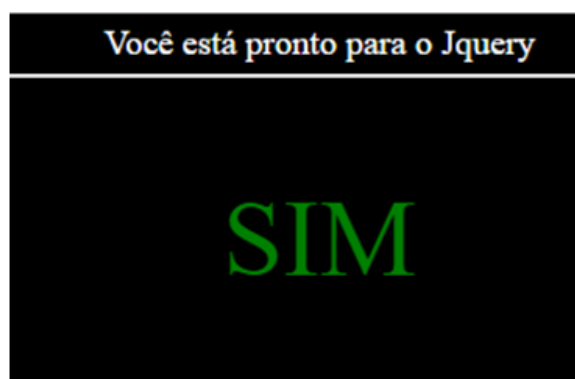
Agora, vamos criar duas **divs**:

```
<div id="titulo">Você está pronto para o Jquery</div>
<div id="mensagem">SIM</div>
```

E para ficar um pouco mais bonito, vamos utilizar **CSS interno** na tag **head**:

```
<style>
  #mensagem,
  #titulo {
    padding: 5px;
    font-size: 18px;
    text-align: center;
    background-color: black;
    color: white;
    border: solid 1px white;
    border-radius: 3px;
  }
  #mensagem {
    padding: 50px;
    font-size: 3.5em;
    color: green;
  }
</style>
```

O resultado deve ser com demonstrado pela figura a seguir.



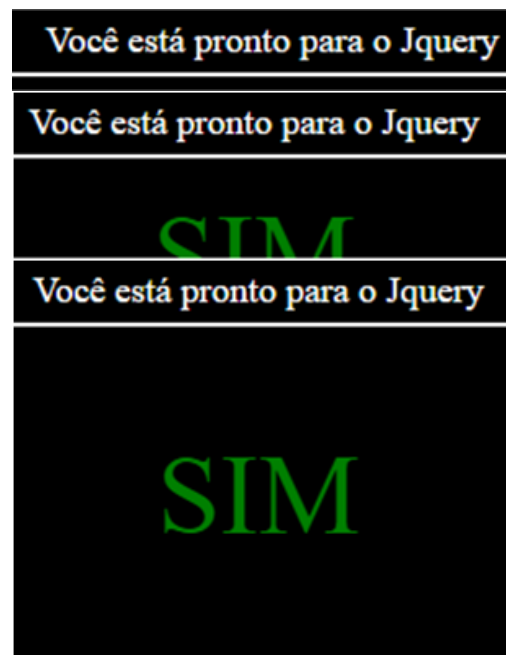
Agora, aplicaremos mais uma propriedade `display:none`; no `id mensagem` no código `CSS`. Como resultado, o retângulo que contém a palavra “**sim**” não será exibido ficando como o demonstrado na figura a seguir.

Você está pronto para o JQuery

A intenção é que quando o usuário interagir com a página, clicando em cima da mensagem, aquela outra `div` apareça. Para isso, codificaremos em **JQuery** e como prometido pelo lema, escrevendo pouco. Veja:

```
<script>
  $(document).ready(function () {
    $("#titulo").click(function () {
      $("#mensagem").slideDown(5000);
    });
  });
</script>
```

Com isso, obteremos o resultado como representado pela sequência de imagens a seguir:



O seu código completo deve ficar:

```
<!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">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <title>Jquery - slideDown</title>
  <script>
    $(document).ready(function () {
```

```

        $("#titulo").click(function () {
            $("#mensagem").slideDown(5000);
        });
    });
</script>
<style>
    #mensagem,
    #titulo {
        padding: 5px;
        font-size: 18px;
        text-align: center;
        background-color: black;
        color: white;
        border: solid 1px white;
        border-radius: 3px;
    }
    #mensagem {
        padding: 50px;
        display: none;
        font-size: 3.5em;
        color: green;
    }
</style>
</head>
<body>
    <div id="titulo">Você está pronto para o JQuery</div>
    <div id="mensagem">SIM</div>
</body>
</html>
</html>

```



Na página 18, você encontra o download completo de todos os exemplos.

Uso de Pluggin

O uso de **plugins** em **jQuery** é quase inevitável, para exemplificar sua aplicação utilizaremos um. Neste exemplo, criaremos um **slideshow** automático, utilizando um **plugin** bem simples.

O primeiro passo é baixar o arquivo **JavaScript**, que no caso encontra-se no final da página desse link: <https://medienfreunde.de/lab/innerfade/>

Download

[jquery.innerfade.zip \(124kb\)](#)

Webdesign Leipzig – Medien

Copie apenas o arquivo **jquery.innerfade**. Este arquivo é o **plugin**. Lembre-se de colocá-lo na mesma pasta do arquivo **Html**, na qual será utilizado. Crie um arquivo **HTML** e salve com o nome de “**jqPluggin**”. Não esqueça de criar a estrutura básica e de **linkar** o **Jquery** e o **plugin**, utilizando o seguinte código:

```
<script type="text/javascript" src="jquery.innerfade.js"></script>
```

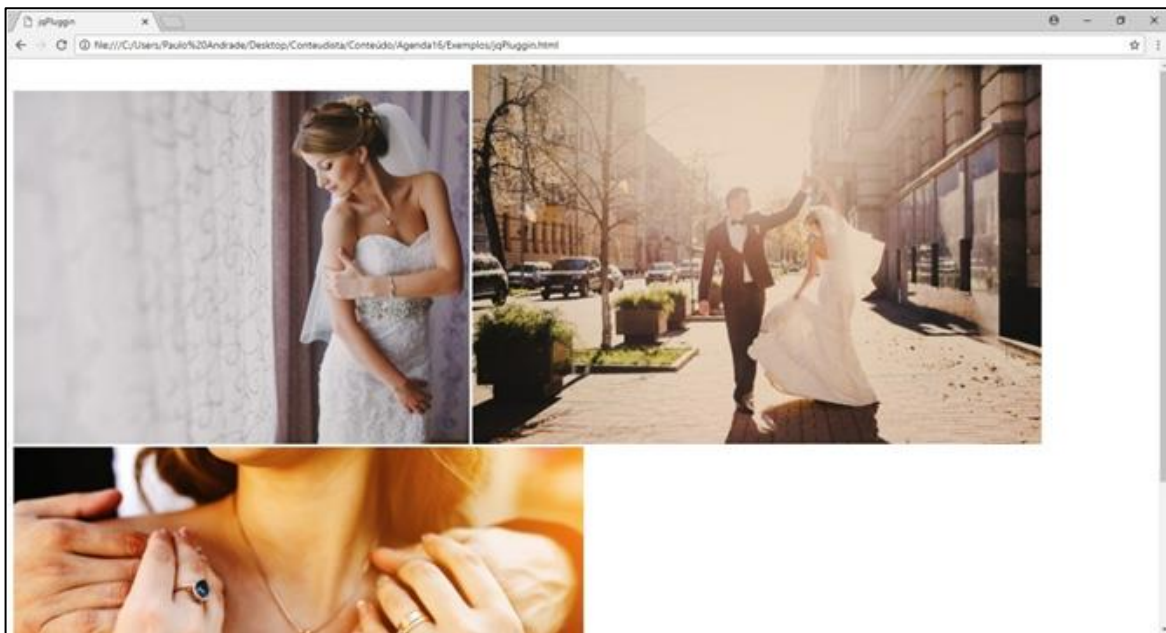
Para criar o **slide show** precisamos de no mínimo três imagens. Salve-as em uma pasta chamada “**imagens**”. Lembre-se de que essa pasta também deve estar no mesmo local que o arquivo **HTML jqPluggin**. Primeiro vamos colocar as imagens na página e dentro da tag **body** implementar o seguinte código:

```
<div id="pre1">

</div>
```

Neste exemplo, usamos três imagens, com os respectivos nomes: **pre1**, **pre2** e **pre3**. As imagens eram muito grandes e para serem melhor visualizadas por meio de **CSS inline**, elas foram reduzidas proporcionalmente. Todas as imagens estão contidas dentro de uma **div** com um **id** definido como “**slide**”.



Obs.1: Caso você tenha separado mais imagens, basta adicionar mais tags **imgs** com os seus respectivos nomes.

Obs.2: Caso não esteja aparecendo alguma imagem, confira o nome e tipo da mesma.

O próximo passo é programar utilizando o **plugin**. Para isso, faça a seguinte codificação dentro da tag **head**:

```
<script>
```

```

$(document).ready(function () {
    $('#slide').innerfade({
        speed: 1000,
        timeout: 3000,
        type: 'sequence'

    });
});
</script>

```

Você deve estar se perguntando: “**Mas é a mesma coisa que jQuery?**”. A resposta é sim! O **plugin** serve apenas para dar funcionalidade ao efeito **innerfade** que não está presente no **jQuery** “puro”. Observe que o efeito é aplicado por meio de passagem de parâmetros: a imagem tem 1 segundo para aparecer (**speed:1000**) e fica por três segundos na tela (**timeout:3000**) e todas são passadas sequencialmente (**type:'sequence'**).

O código completo deve ser semelhante ao da imagem a seguir:

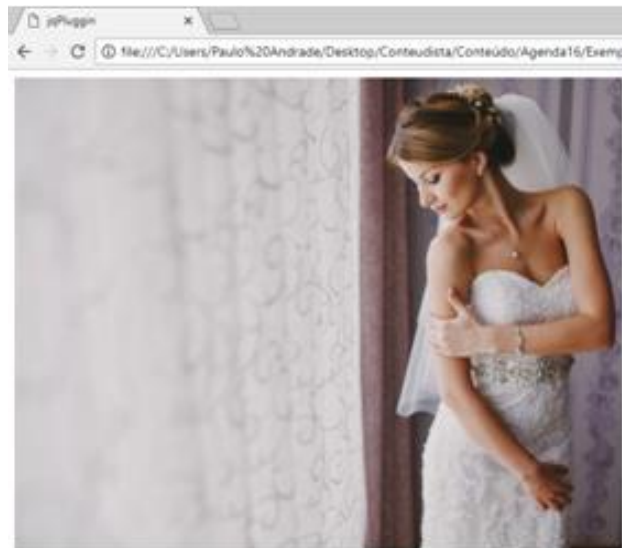
```

<!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">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script type="text/javascript" src="jquery.innerfade.js"></script>
    <title>jqPluggin</title>
    <script>
        $(document).ready(function () {
            $('#slide').innerfade({
                speed: 1000,
                timeout: 3000,
                type: 'sequence'

            });
        });
    </script>
</head>
<body>
    <div id="slide">
        
        
        
    </div>
</body>
</html>

```

E resultado deve ser parecido com a ilustração a seguir:



Obs: Caso tenha curiosidade sobre a quantidade de códigos que foram necessários para programar o efeito **innerfade**, basta abrir esse arquivo, utilizando o **Visual Code**. De qualquer forma já é possível dizer que, excluindo os comentários, são mais de **80** linhas de código!

Se você quiser encontrar mais **plugins** gratuitos, acesse o link <https://www.jqueryscript.net/>



Na página 18, você encontra o download completo de todos os exemplos.

Outros Recursos

O **JQuery** oferece muitos outros recursos. Poderíamos ter um curso inteiro apenas para esse incrível **framework**.

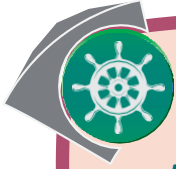
Os **links** a seguir, lhe darão uma ideia da quantidade de recursos encontrados nessa ferramenta. Veja: Neste primeiro link <https://api.jquery.com/>, é possível observar um site com toda a documentação do **jQuery**, com todos os padrões, juntamente com seus métodos, eventos e efeitos.

Obs.: Vale ressaltar que aqui também tem a documentação das versões anteriores do **Jquery**. Abaixo temos alguns links para documentação do **JQuery**.

- Global Ajax Event Handlers
 - Helper Functions
 - Low-Level Interface
 - Shorthand Methods
- Attributes
- Callbacks Object
- Core
- CSS
- Data
- Deferred Object
- Deprecated
 - Deprecated 1.3
 - Deprecated 1.7
 - Deprecated 1.8
 - Deprecated 1.9
 - Deprecated 1.10
 - Deprecated 3.0
 - Deprecated 3.3
- Dimensions
- Effects
 - Basics
 - Custom
 - Fading
 - Sliding
- Events
 - Browser Events
 - Document Loading
 - Event Handler Attachment
 - Event Object
 - Form Events
 - Keyboard Events
 - Mouse Events
- Forms
- Internals
- Manipulation
 - Class Attribute
 - Copying
 - DOM Insertion, Around
 - DOM Insertion, Inside
 - DOM Insertion, Outside
 - DOM Removal
 - DOM Replacement
 - General Attributes
 - Style Properties
- Miscellaneous
 - Collection Manipulation
 - Data Storage
- Setup Methods
- Offset
- Properties
 - Properties of jQuery Object Instances
 - Properties of the Global jQuery Object
- Removed
- Selectors
 - Attribute
 - Basic
 - Basic Filter
 - Child Filter
 - Content Filter
 - Form
 - Hierarchy
 - jQuery Extensions
 - Visibility Filter
- Traversing
 - Filtering
 - Miscellaneous Traversing
 - Tree Traversal
- Utilities
- Version
 - Version 1.0
 - Version 1.0.4
 - Version 1.1
 - Version 1.1.2
 - Version 1.1.3
 - Version 1.1.4
 - Version 1.2
 - Version 1.2.3
 - Version 1.2.6
 - Version 1.3
 - Version 1.4
 - Version 1.4.1
 - Version 1.4.2
 - Version 1.4.3
 - Version 1.4.4
 - Version 1.5
 - Version 1.5.1
 - Version 1.6
 - Version 1.7
 - Version 1.8
 - Version 1.9
 - Version 1.12 & 2.2
 - Version 3.0
 - Version 3.1



[Clique aqui para fazer o download completo de todos os exemplos.](#)



VOCÊ NO COMANDO

Depois que o Seu Ubaldo já havia definido o layout do site do casamento da sua filha, a dona Matilde, sogra da filha do Seu Ubaldo entra em cena, dizendo que ficaria melhor se as fotos pudessem ser disponibilizadas por seções, permitindo aos usuários acessar as seções de sua preferência, clicando para exibir ou esconder cada seção.

Assim, Seu Ubaldo solicitou que a nossa equipe realizasse essa atualização.

Com os estudos dessa aula, você saberia fazer as alterações solicitadas no site do casamento de Zuleika & Ubiratam?

Dicas:

- Utilize `fadeToggle` para esconder as divisões das fotos;
- Não esqueça de linkar o **Jquery** na tag `head`;
- Utilize `id` para identificar as `divs` que serão alteradas pelo **Jquery**.

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

A seguir, confira se você conseguiu resolver os desafios propostos!

Antes de exibir a resposta, lembre-se de que até agora estudamos muitos recursos em **HTML**, **CSS**, **W3CSS**, **JavaScript** e **JQuery**. A resolução mostrada a seguir é apenas uma das muitas possibilidades.



Código HTML / JavaScript / Jquery Completo:

```
<!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">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <title>Casamento de Minha Princesa</title>
  <script>
    $(document).ready(function () {
      $("#pre").click(function () {
        $("#pre1").fadeToggle(1000);
      });
    });
  </script>
</head>
```

```

        $("#dia").click(function () {
            $("#dia1").fadeToggle(1000);
        });
    });
</script>
</head>

<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">
            
        </div>
        <div class="w3-third">

            <br>
            <div class="w3-content w3-section w3-center" style="">
                
                
                
            </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">
            
        </div>
    </div>
    <br>
    <div id="pre" class="w3-container" style=" background-color: rgb(195, 165, 116); color:white;">
        <h2>Pré Casamento</h2>
    </div>

```



```

<div id="pre1" class="w3-row-padding w3-margin-top ">
  <div class="w3-third">
    <div class="w3-card">
      
      <div class="w3-container">
        <h5>Linda</h5>
      </div>
    </div>
  </div>

  <div class="w3-third">
    <div class="w3-card">
      
      <div class="w3-container">
        <h5>Casal</h5>
      </div>
    </div>
  </div>

  <div class="w3-third">
    <div class="w3-card">
      
      <div class="w3-container">
        <h5>Amor</h5>
      </div>
    </div>
  </div>
</div>

<div id="dia" class="w3-container" style=" background-color: rgb(195, 165, 116);
color:white;">
  <h2>Dia da Noiva</h2>
</div>
<div id="dia1" class="w3-row-padding w3-margin-top">
  <div class="w3-third">
    <div class="w3-card">
      
      <div class="w3-container">
        <h5>Ela!</h5>
      </div>
    </div>
  </div>

  <div class="w3-third">
    <div class="w3-card">
      
      <div class="w3-container">
        <h5>O vestido!</h5>
      </div>
    </div>
  </div>
</div>

```

```

        </div>
    </div>
</div>

<div class="w3-third">
    <div class="w3-card">
        
        <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>
</html>

```

Observe o código! É importante ressaltar que definimos **id** para as **divs** e que foram aplicados os efeitos por meio do evento **click**.

Nas **divs pré** e **dia** serão configurados os eventos de **clicks** para que o efeito de aparecer ou desaparecer seja aplicado nas **divs pré1** e **dia 1**, respectivamente. Para que isso seja possível, é necessário o código a seguir:

```

$(document).ready(function () {
    $("#pre").click(function () {
        $("#pre1").fadeToggle(1000);
    });
    $("#dia").click(function () {
        $("#dia1").fadeToggle(1000);
    });
});

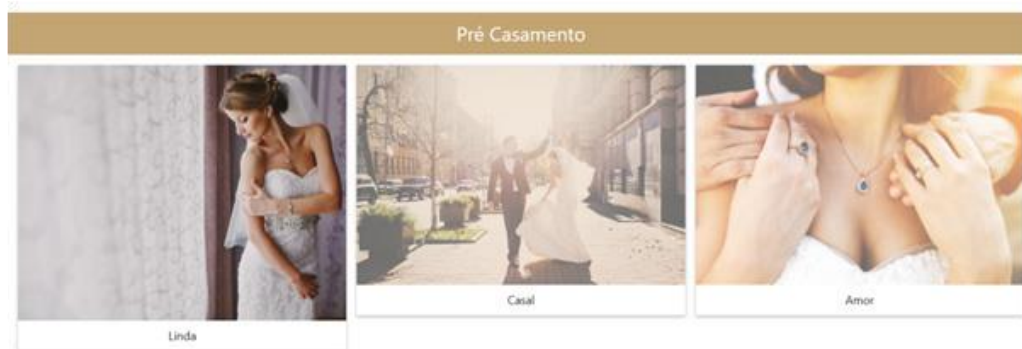
```

Obs: Note que é possível definir vários elementos para terem **eventos** e **efeitos aplicados** simultaneamente.

Resultado Navegador:



Ao clicar em qualquer uma das seções, será aplicado um efeito **fade** que trará apenas as fotos dessa seção.



[Clique aqui para fazer o download completo dos arquivos do “Você no Comando”](#)

Imagens: Crédito Freepik.com e Pixabay.com