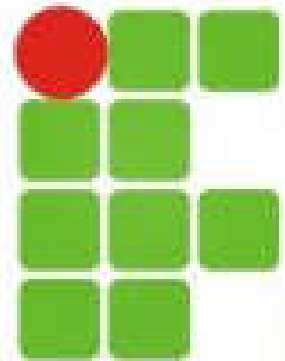


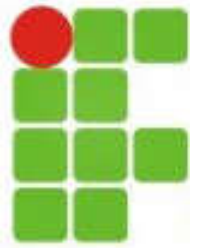
Fundamentos de Web Design 2

Professor Eng. Dr. Will Roger Pereira



Objetivos desta aula

- Conhecer os eventos jQuery.





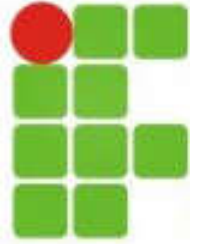
Introdução

- Todas as ações do visitante que uma página Web pode responder/reagir, são chamadas eventos.
- Um evento representa o momento preciso quando algo acontece:
 - Colocar um mouse encima de um elemento;
 - Clicar em um elemento;
 - Mudar o estado de um checkbox.
- No momento que um evento acontece, abre-se a possibilidade de executar ações.



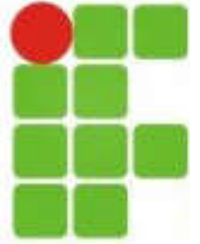
Introdução

- O termo "dispara" é utilizado muito em eventos. Vem do inglês *trigger*.
- Referências sobre todos os eventos jQuery:
 - www.w3schools.com/jquery/jquery_ref_events.asp
- Eventos também seguem a sintaxe jQuery:
 - SELETOR.ação(argumentos)
 - Ex: `$("#will").css("color","red");`
- Eventos → SELETOR.evento(argumentos);



Sintaxe Eventos jQuery

- Um evento dispara uma ação. Uma ação é representada por qual estrutura? R: Função.
- **Esta função é denominada função de callback.**
- Assim sendo, quando quiseres anexar um evento a um seletor, utilize a seguinte sintaxe:
 - `SELETOR.evento(função);`
- Neste caso, sempre o que estará dentro dos parênteses do evento será uma função, como feito na primeira aula.



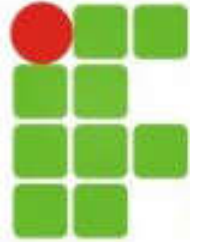
Sintaxe Eventos jQuery

- Nela será codificada a ação, ou o conjunto de instruções que serão executados no momento que o evento disparar.
- Quando o evento disparar, a função dentro dele será executada.
- Dentro destas funções, os seletores aprendidos poderão ser utilizados para manipular os mais diversos elementos.
- Os exemplos mostrados aqui podem ser utilizados com qualquer seletor. Porém, alguns eventos são restritos a alguns elementos.



Principais Eventos

- **click** → Clique sobre o elemento;
- **dblclick** → Duplo clique sobre o elemento;
- **mouseenter** → Mouse fica sobre o elemento;
- **mouseleave** → Mouse deixa de ficar sobre o elemento;
- **hover** → Combinação de **mouseenter** e **mouseleave**;
- **change** → O **value** de um elemento **input** é modificado;
- **focus** → Um **input** ganha foco;
- **blur** → Um **input** perde foco;
- **mousedown** → O botão do mouse é pressionado;
- **mouseup** → O botão do mouse é solto;
- **submit** → Um **form** é submetido.



Sintaxe Completa

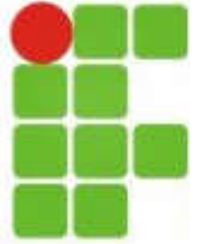
```
SELETOR.EVENTO(function(e) {  
    /*Código a ser executado quando  
    EVENTO for disparado em um elemento  
    atingido por SELETOR*/  
});
```

- **SELETOR**: Indica qual(is) elemento(s) terão o EVENTO anexado;
- **EVENTO**: Indica qual evento será anexado;
- **e**: Estrutura que contém as propriedades do evento disparado. Seu uso é facultativo.



Problema!!!

- Dentro da função de callback, pode-se utilizar todos os seletores que foram aprendidos;
- Quando somente um elemento é contemplado pelo seletor, fica muito fácil saber quem disparou o EVENTO;
- Porém quando um EVENTO é anexado a diversos elementos de uma só vez, como manipular o elemento que disparou o evento????
- R: O seletor **this**.



Seletor *this*

- Este seletor especial permite selecionar quem disparou o evento;
- Este é o único seletor aprendido que não necessita de aspas;

```
SELETOR.EVENTO(function(e) {  
    $(this).css("color", "red");  
});
```

- Mudará a cor da letra de quem disparou o evento para vermelho (se for possível).



Seletor *this*

- Para aproveitar este seletor e também selecionar filhos, utilize o seletor `$(this).find("filho")`:

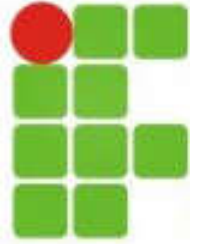
```
SELETOR.EVENTO(function() {  
    $(this).find("li").css("color", "red");  
});
```

- Caso o elemento selecionado tenha filho(s) li, mudará a cor da letra deles para vermelho.



Observações importantes

- Cada elemento pode possuir quantos eventos forem desejados;
- Caso um evento seja anexado novamente ao elemento, será priorizado o último evento anexado;
- Um elemento pode ter um evento anexado só, e somente se ele já estiver criado;
- Quando for criar um elemento e desejar que ele possua determinado evento, anexe o evento após a criação do elemento. Ou seja, **não funciona como CSS!!!**



Eventos

- Os eventos serão todos demonstrados em arquivos HTML, que se encontram anexos a esta aula;
- Cada evento possuirá um arquivo específico;
- Particularidades os eventos, bem como uma explicação mais detalhada sobre seu funcionamento estarão dentro do arquivo HTML.