Programação para Internet Rica

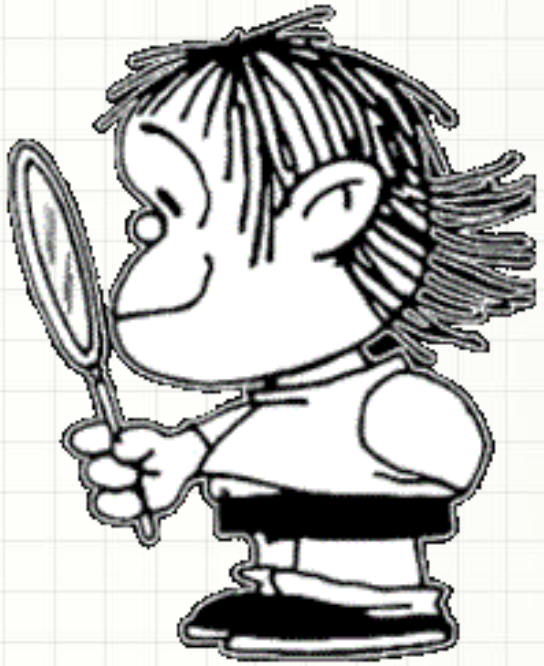
Manipulando XHTML e CSS com JavaScript

Prof. Dr. Daniel Caetano

2013 - 1

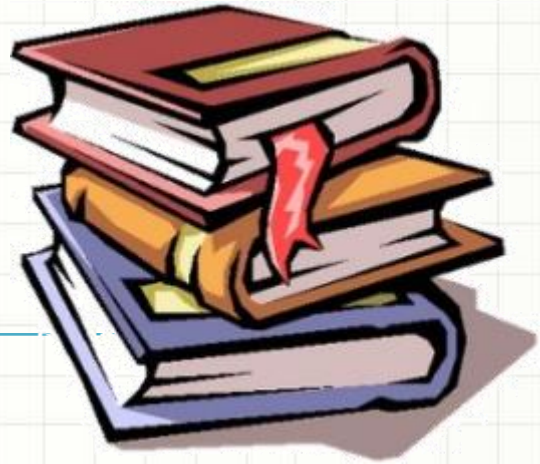
Objetivos

- Apresentar a manipulação direta de elementos do XHTML e CSS com o uso do JavaScript
- Apresentar e eliminar elementos (PopUp, por exemplo)
- Mudar funcionalidade de elementos



- **TRABALHO A ONLINE!**

Material de Estudo



Material	Acesso ao Material
Notas de Aula	http://www.caetano.eng.br/ (Aula 7)
Apresentação	http://www.caetano.eng.br/ (Aula 7)
Material Didático	Aprenda a Criar Páginas Web c/ HTML, páginas 609 a 648
Google	+“JavaScript” +tutorial +DHTML
Web Sites	http://www.w3.org/

Codificação "Client-Side"

Codificação Client-Side

- Por que codificar no lado do cliente?
- Respostas rápidas
- Sensação de interação em tempo real
- Diminuir consumo de banda
- Diminuir processamento no servidor

- Manipular dados do XHTML
  - document.getElementById(“id”)

Como "Sumir" com Elementos da Página

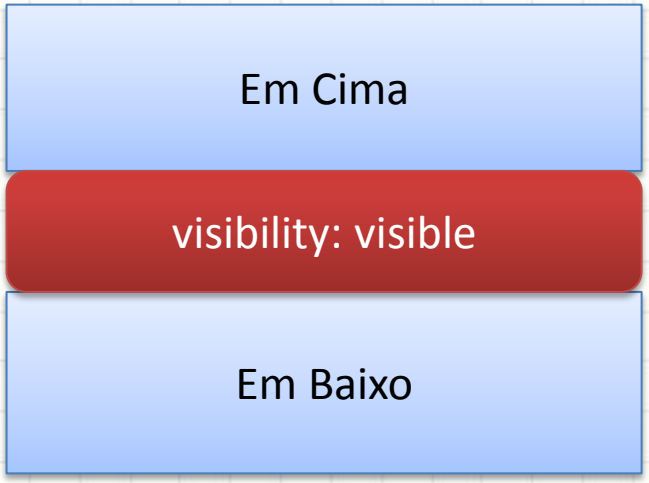
Sumindo com Elementos

- Sumir = não desenhar
- Prático:
  - Deixar tudo pronto
  - Mostrar só quando necessário
- Duas formas de fazer o “sumiço”
  A. Elemento continua ocupando seu espaço
  B. Elemento some completamente

Sumindo com Elementos

- Elemento ocupa espaço mas não é desenhado
- Propriedade: **visibility**

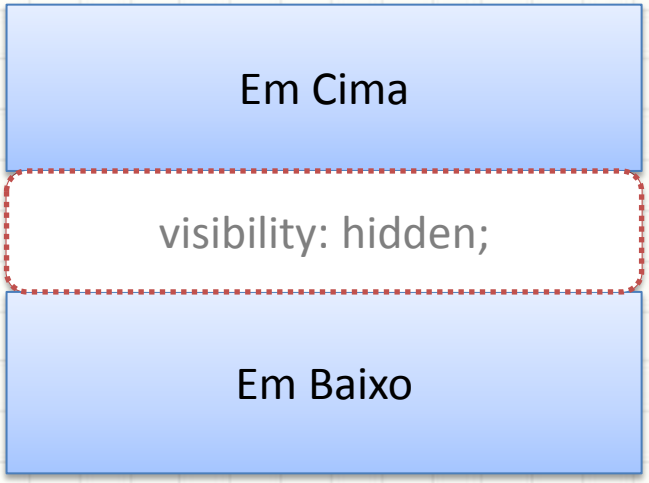
- **visibility: visible;**
  - Mostra elemento



Sumindo com Elementos

- Elemento ocupa espaço mas não é desenhado
- Propriedade: **visibility**

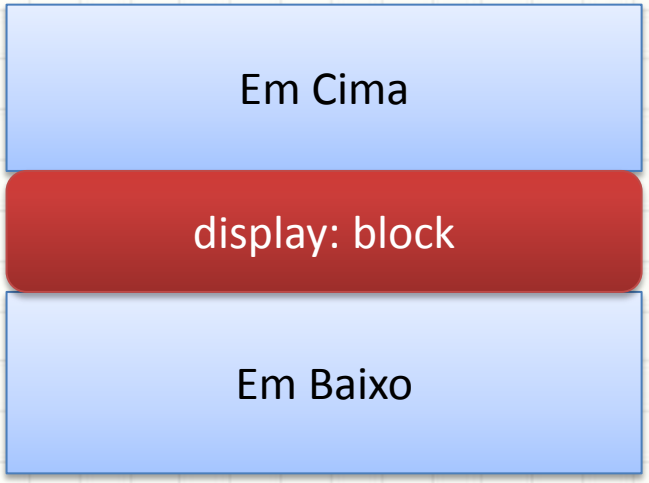
- **visibility: visible;**
  - Mostra elemento
- **visibility: hidden;**
  - Esconde elemento



Sumindo com Elementos

- Elemento nem ocupa espaço nem é desenhado
- Propriedade: **display**

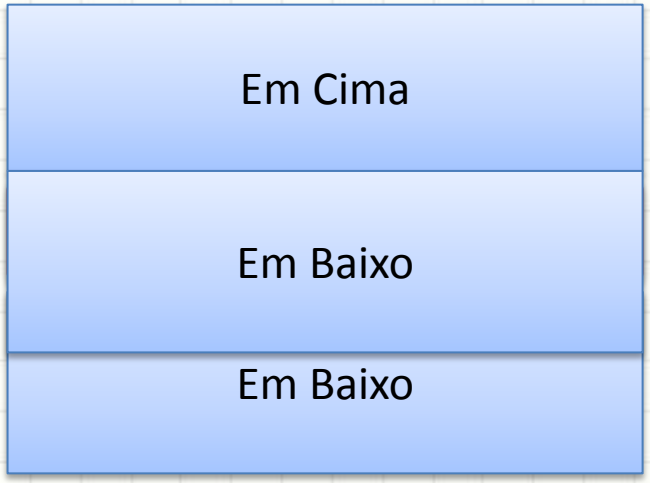
- **display: block;**
  - Mostra elemento



Sumindo com Elementos

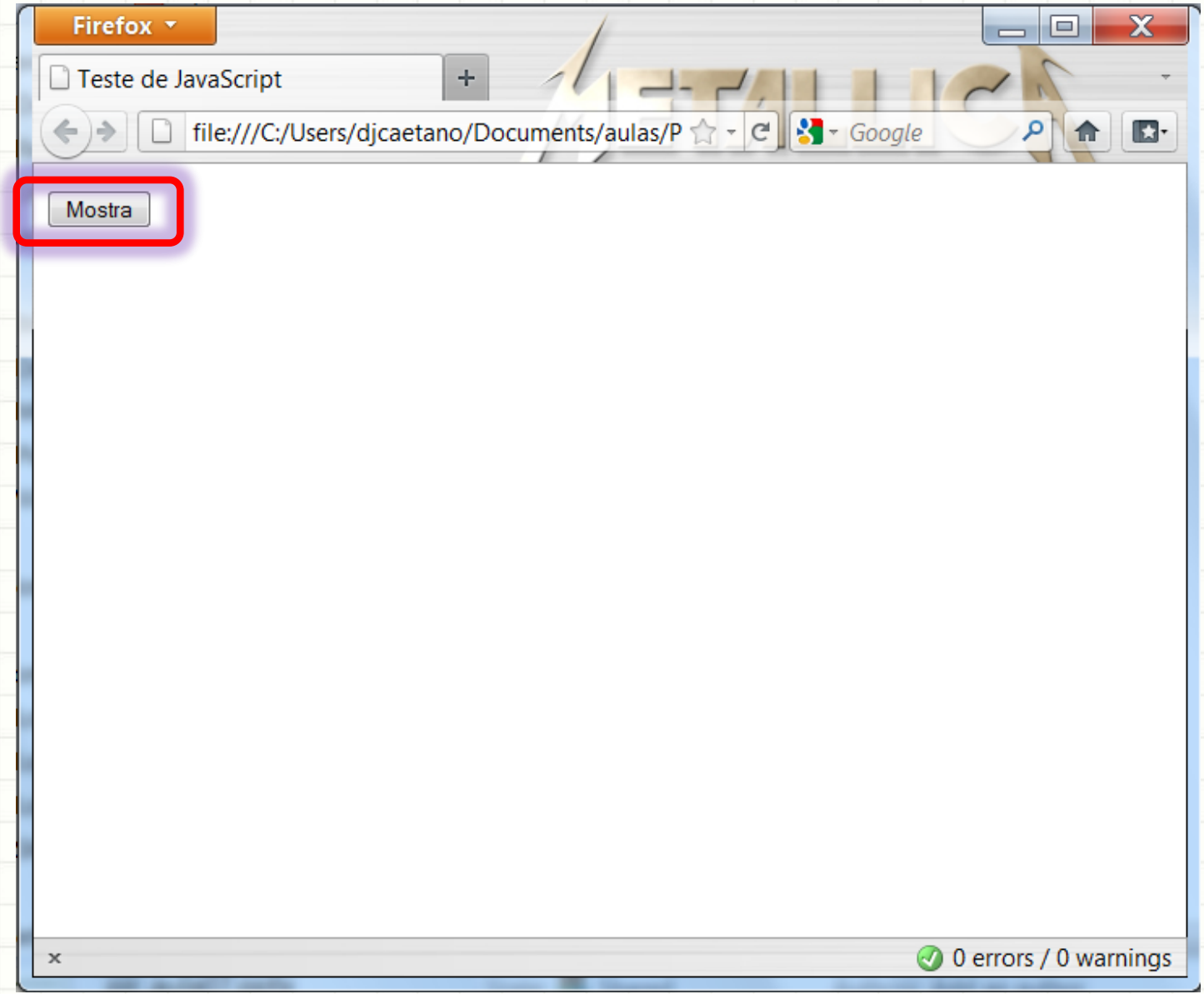
- Elemento nem ocupa espaço nem é desenhado
- Propriedade: **display**

- **display: block;**
  - Mostra elemento
- **display: none;**
  - Esconde elemento

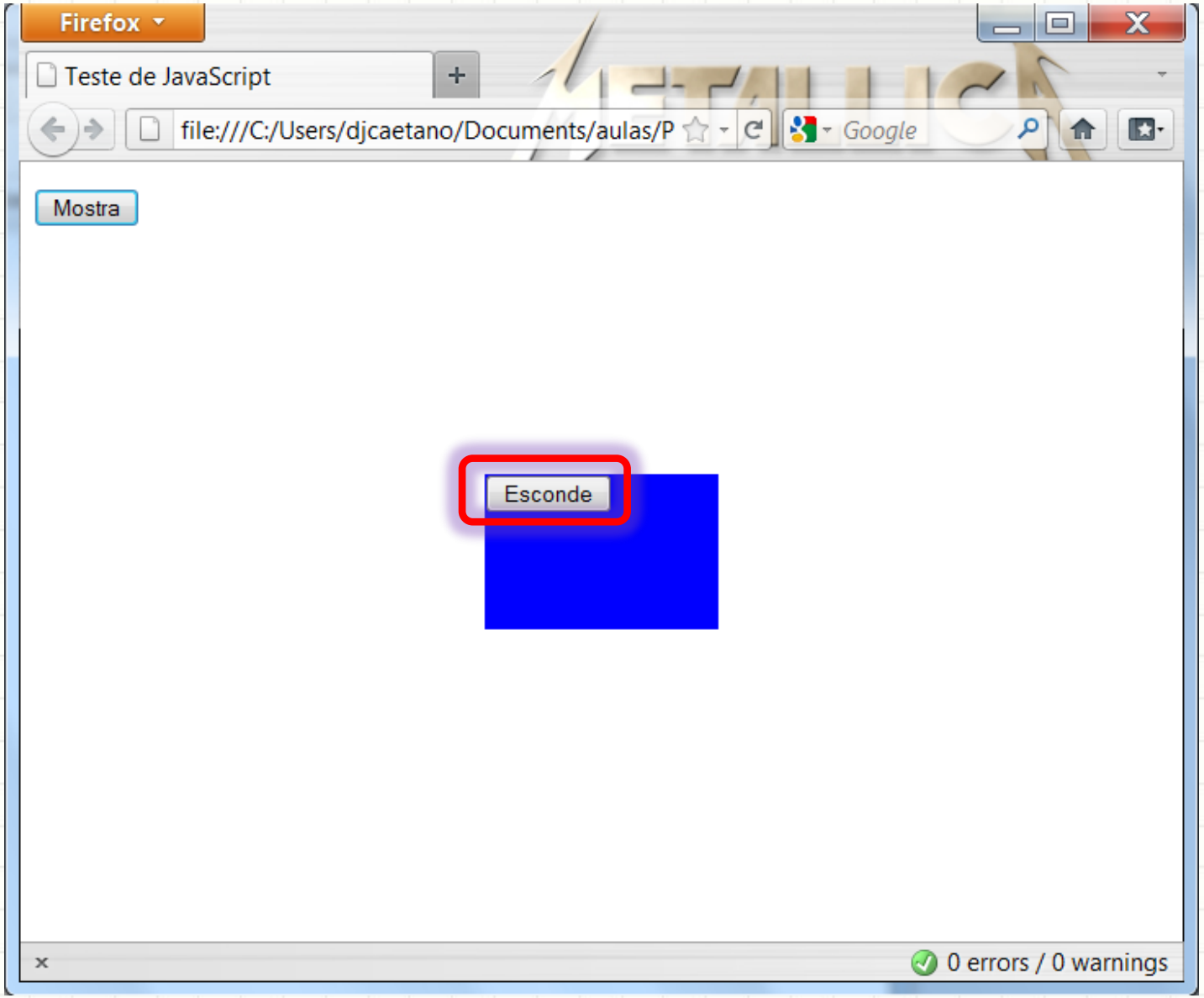


TUTORIAL POPUP

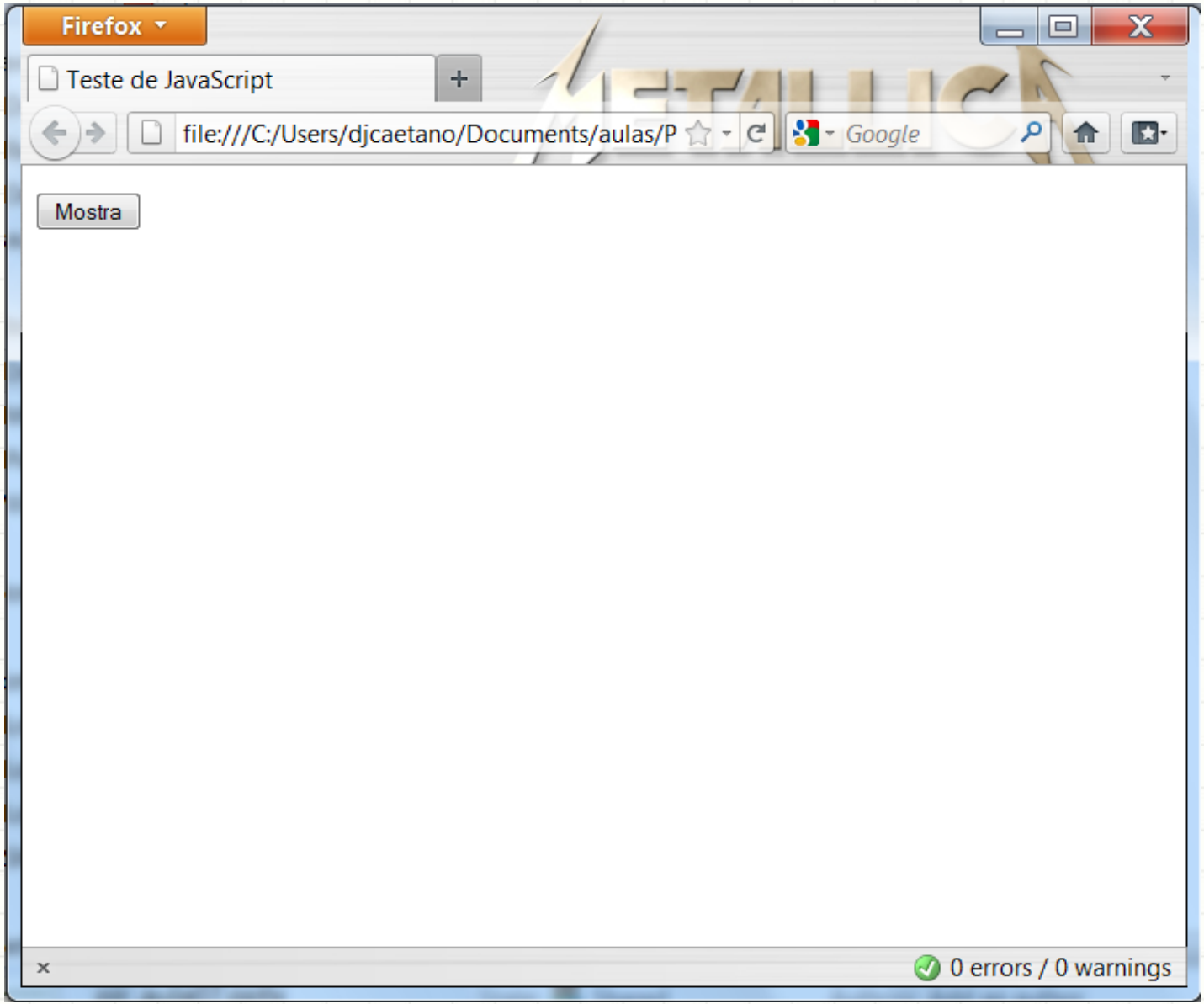
Um PopUp em JavaScript+CSS



Um PopUp em JavaScript+CSS



Um PopUp em JavaScript+CSS



Associando Funcionalidades a Outros Elementos

Funcionalidades em Elementos

- Quando associamos funções direto no HTML
  - Só podemos associar ações a links e botões...
- No JavaScript, entretanto, não há limitação!
- Podemos associar ações a um **div**, por exemplo!

A. Fechar o PopUp quando clicarmos duas vezes no mesmo;
B. Fechar o PopUp quando passarmos o mouse por cima do mesmo.

Funcionalidades em Elementos

- Quanc o HTML
  - Só p tões...
- No Jav itação!
- Podemos associar ações a um **div**, por exemplo!

Vamos testar!

A. Fechar o PopUp quando clicarmos duas vezes no mesmo;
B. Fechar o PopUp quando passarmos o mouse por cima do mesmo.

Trocando a Funcionalidade de Elementos da Página

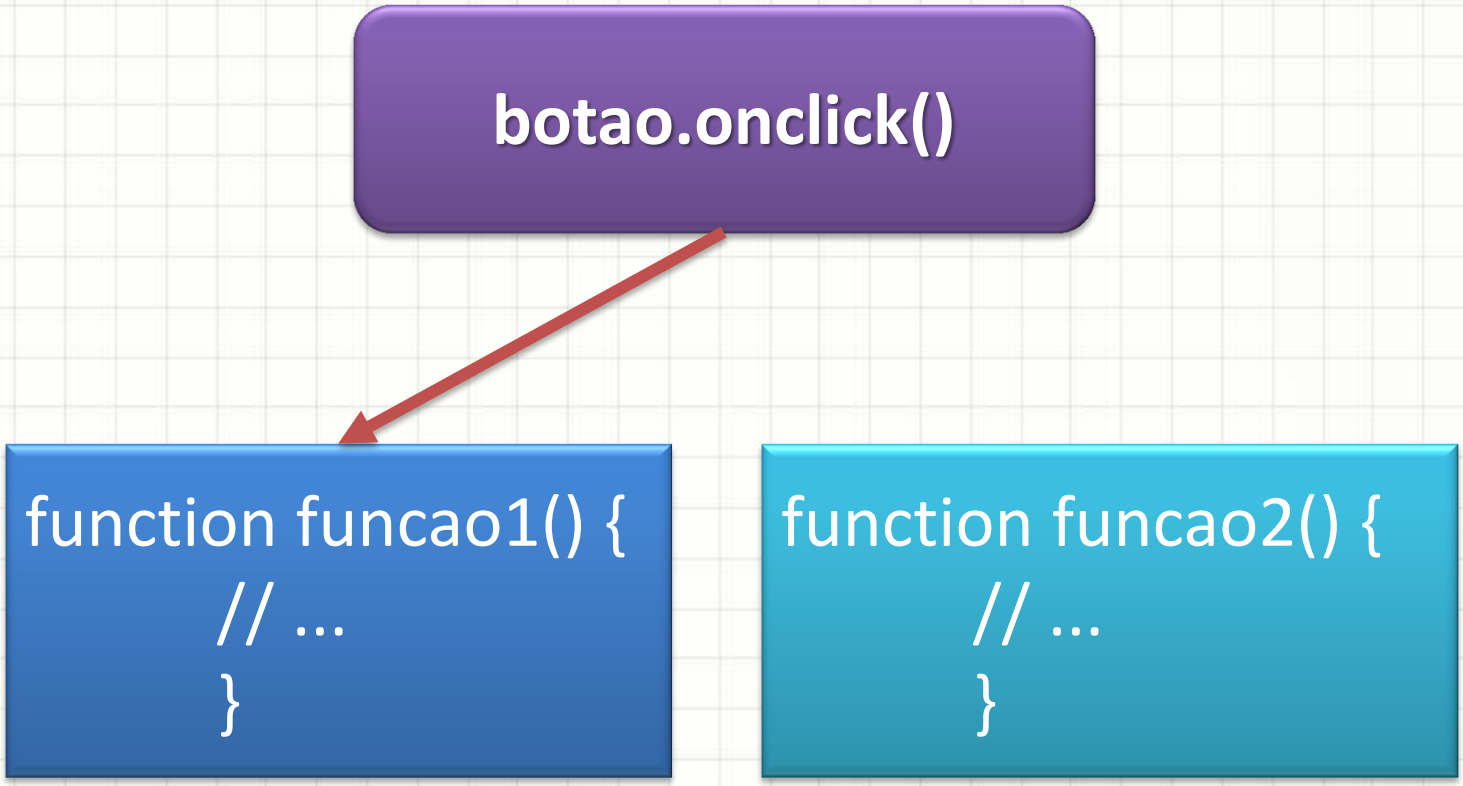
Trocar a Funcionalidade

- Na função init()...
  - Associamos uma função ao botão “mostra”
- É possível mudar a função associada!
- Basta indicar um novo nome

```
botao.onclick = funcao1;
(...)
botao.onclick = funcao2;
```

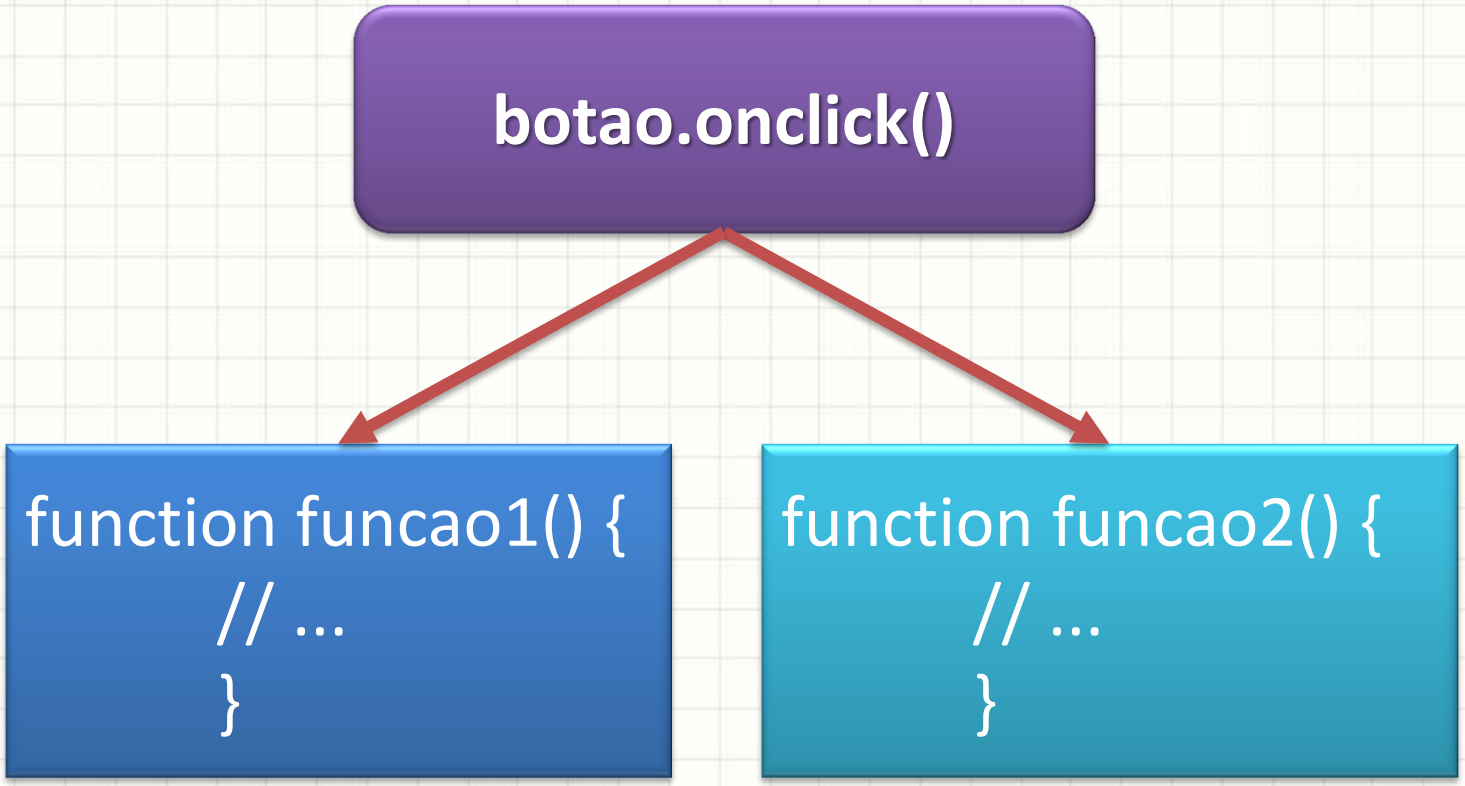
Trocar a Funcionalidade

botao.onclick = funcao1;



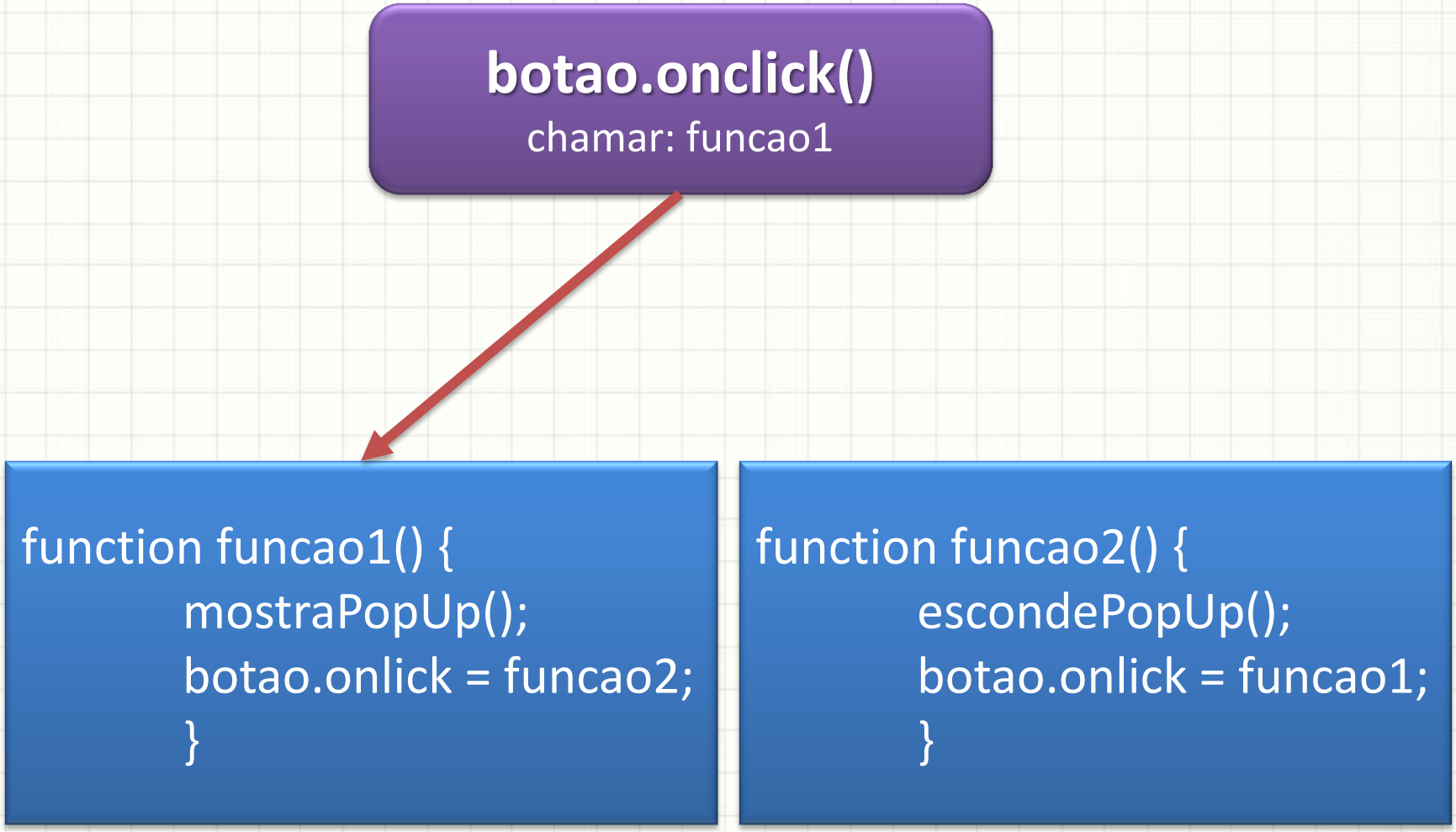
Trocar a Funcionalidade

```
botao.onclick = funcao1;
botao.onclick = funcao2;
```



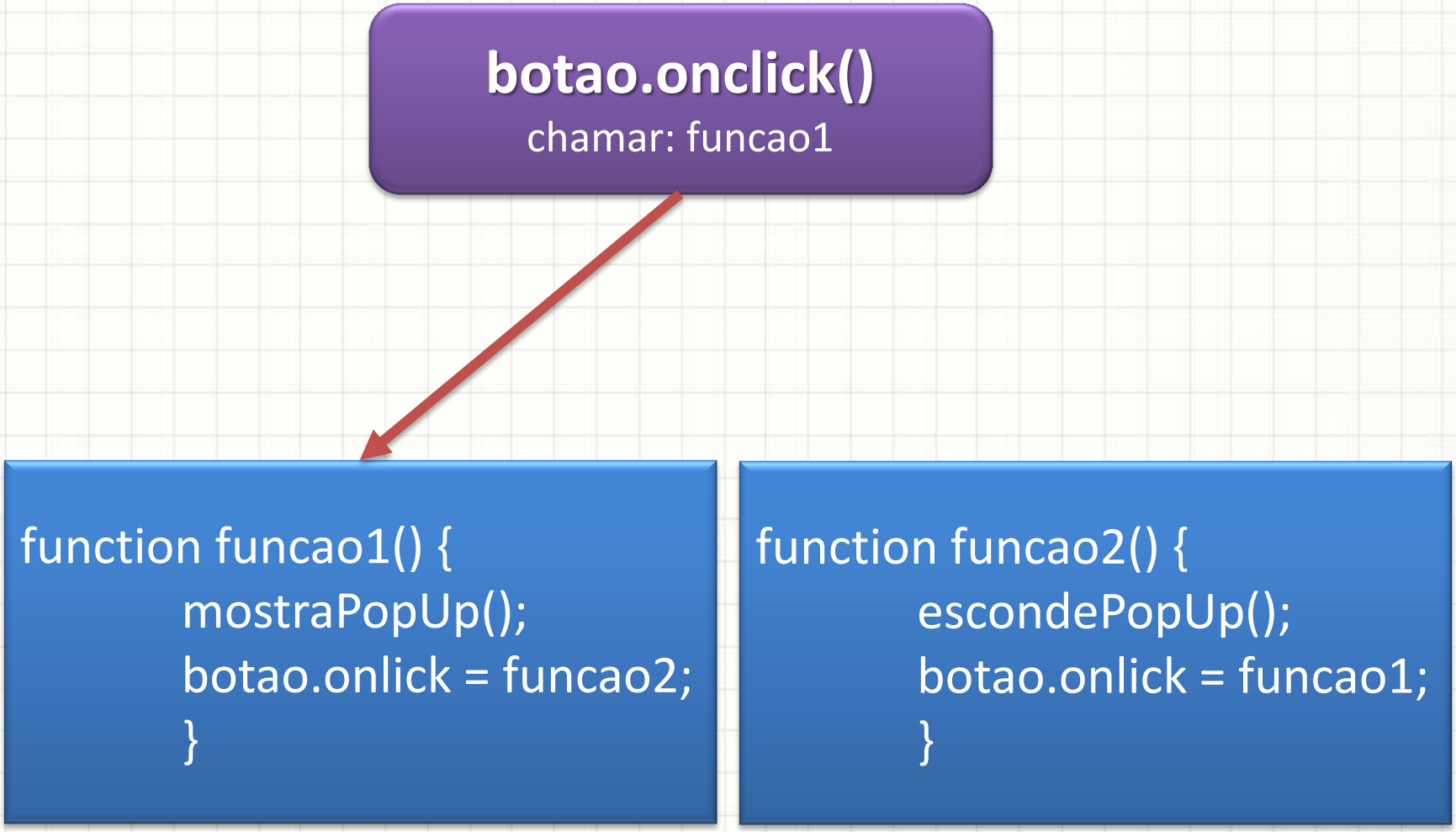
Trocar a Funcionalidade

- Exemplo:



Trocar a Funcionalidade

- Exemplo:



Trocar a Funcionalidade

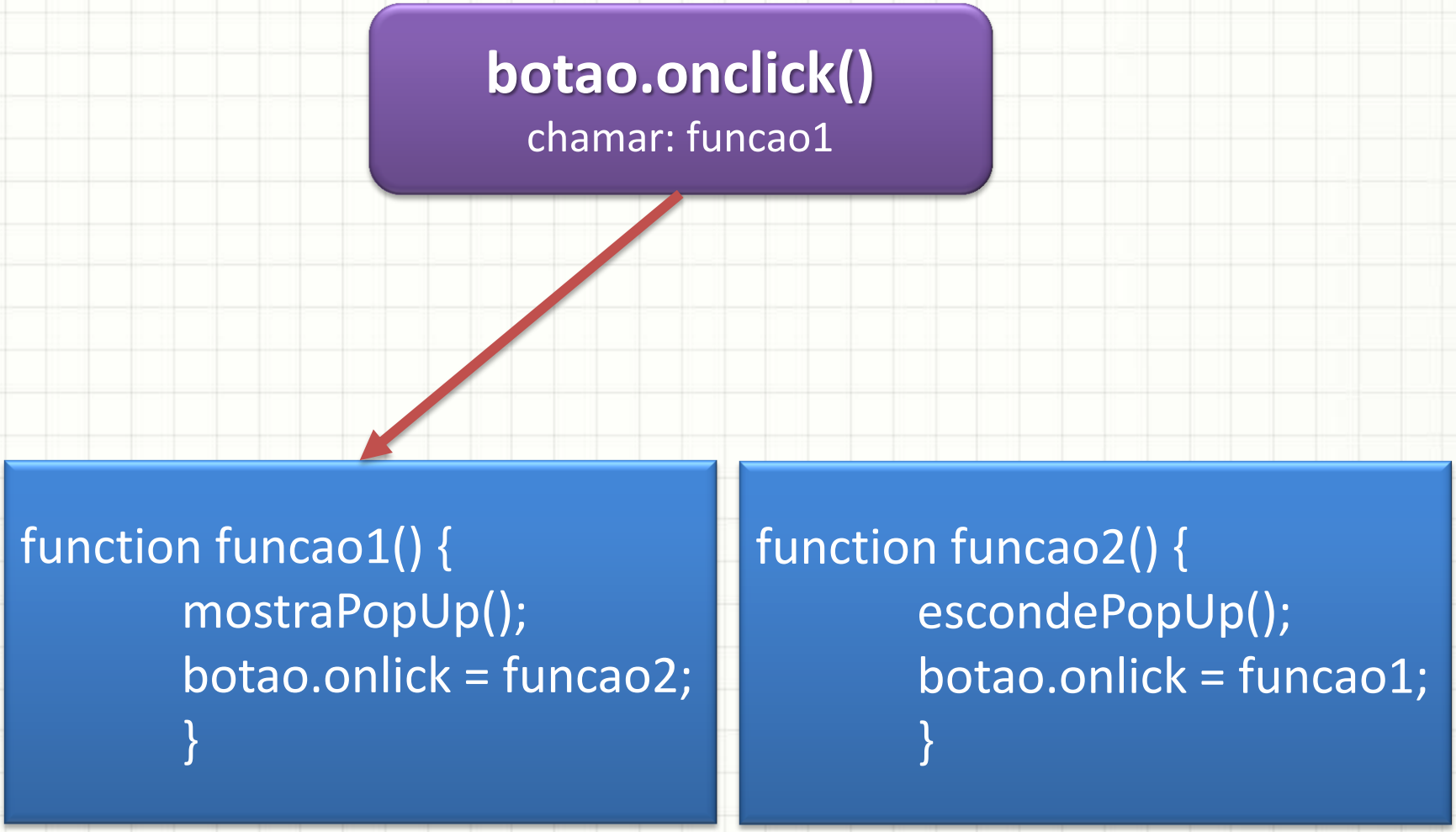
- Exemplo:


```
function funcao1() {
        mostraPopUp();
        botao.onlick = funcao2;
        }
```

```
function funcao2() {
        escondePopUp();
        botao.onlick = funcao1;
        }
```

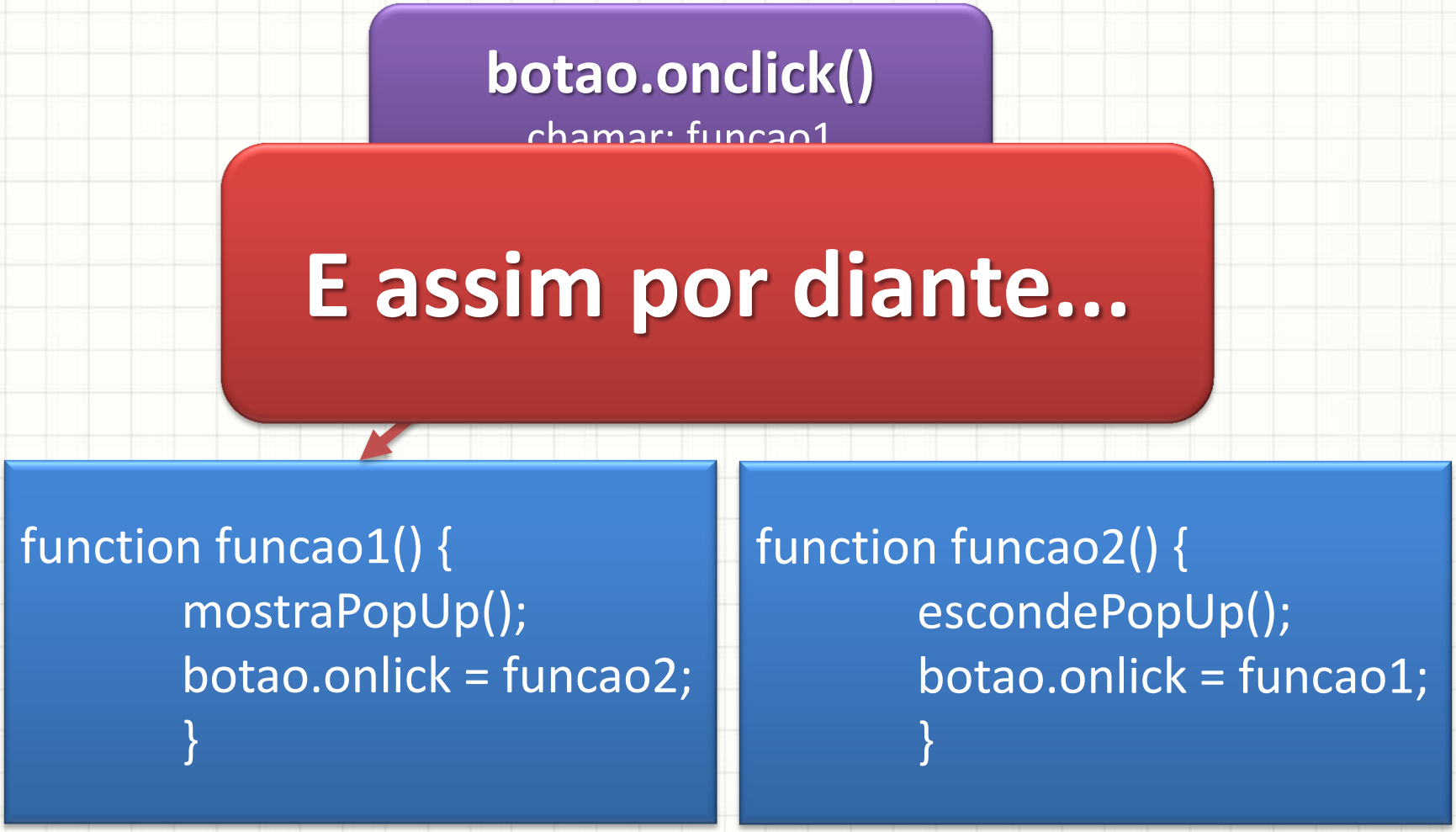
Trocar a Funcionalidade

- Exemplo:



Trocar a Funcionalidade

- Exemplo:



Trocar a Funcionalidade

- Vamos testar?

  A. Modifique o botão que abre o PopUp para que, após abri-lo, sirva também para fechá-lo;
  B. Incremente o código para que, quando a função do botão mudar, mude também seu texto.

Vários Eventos, Uma Única Função

Função para Múltiplos Eventos

- Uma mesma função JS pode ser associada a vários eventos e elementos...

```
function mudaCor() {
        document.body.style.color=“green”;
        }

button1.onclick = mudaCor;
div1.onclick = mudaCor;
td1.onclick = mudaCor;
```

Função para Múltiplos Eventos

- Mas essa função muda sempre o texto global!

```
function mudaCor() {
    document.body.style.color = “green”;
    }
```

- E se quisermos mudar apenas a cor do texto do elemento clicado?

Função para Múltiplos Eventos

- Exemplo:
  - Clicou em um botão, muda cor texto deste botão
  - Clicou em um div, muda cor deste div

- Dois jeitos: o “força bruta” e o “sei o que faço”
- Força Bruta?
  - Uma função diferente para cada elemento

- E o outro método?

Função para Múltiplos Eventos

- Jeito “inteligente”:

```
function mudaCor() {
    this.style.color = “green”;
    }
```

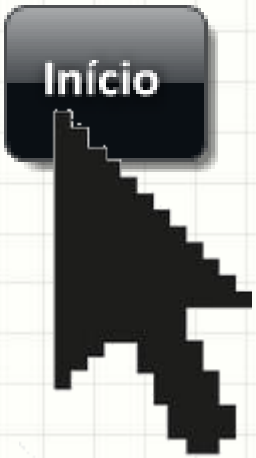
- *this*?
- this: referencia ao objeto “atual” (clicado)

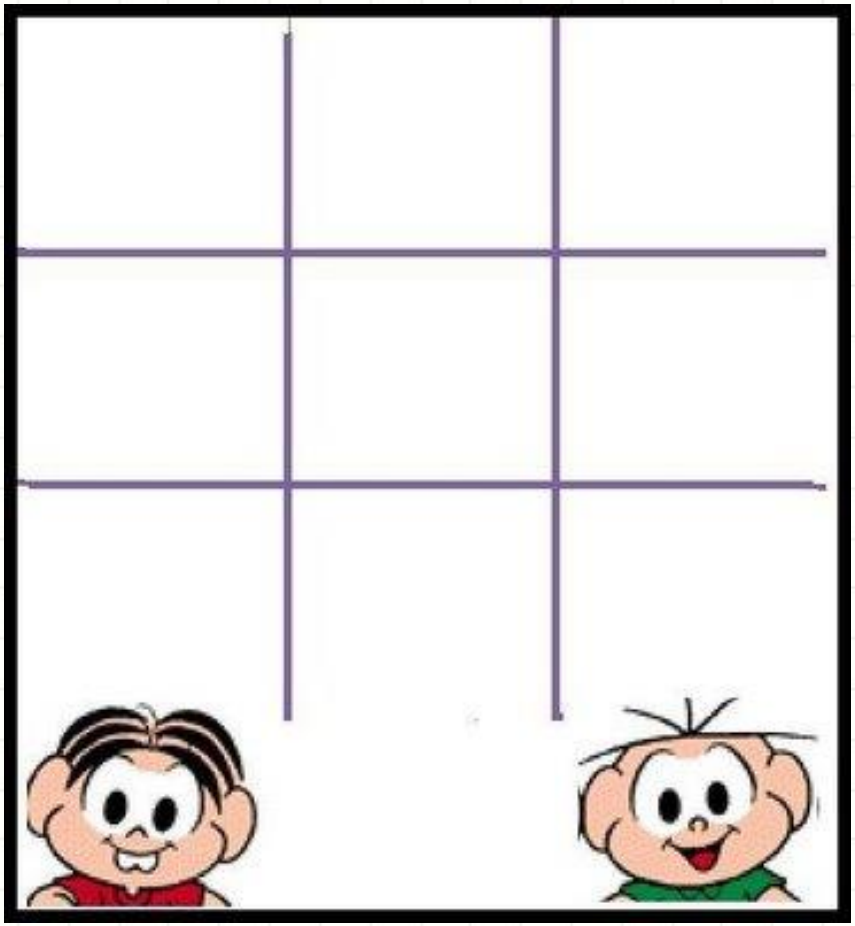
```
button1.onclick = mudaCor;
div1.onclick = mudaCor;
td1.onclick = mudaCor;
```

Tutorial: Adicionando JavaScript no Jogo da Velha

Jogo da Velha

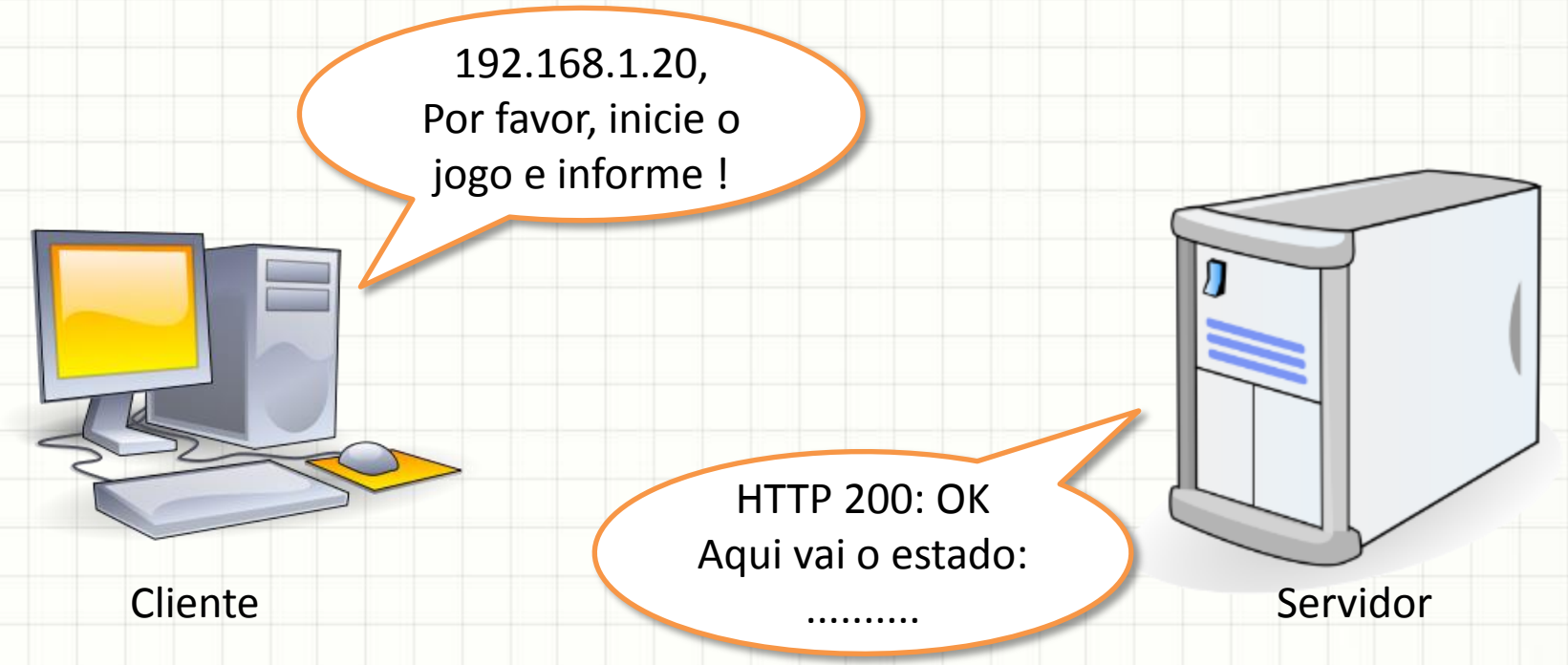
- Lógica de funcionamento





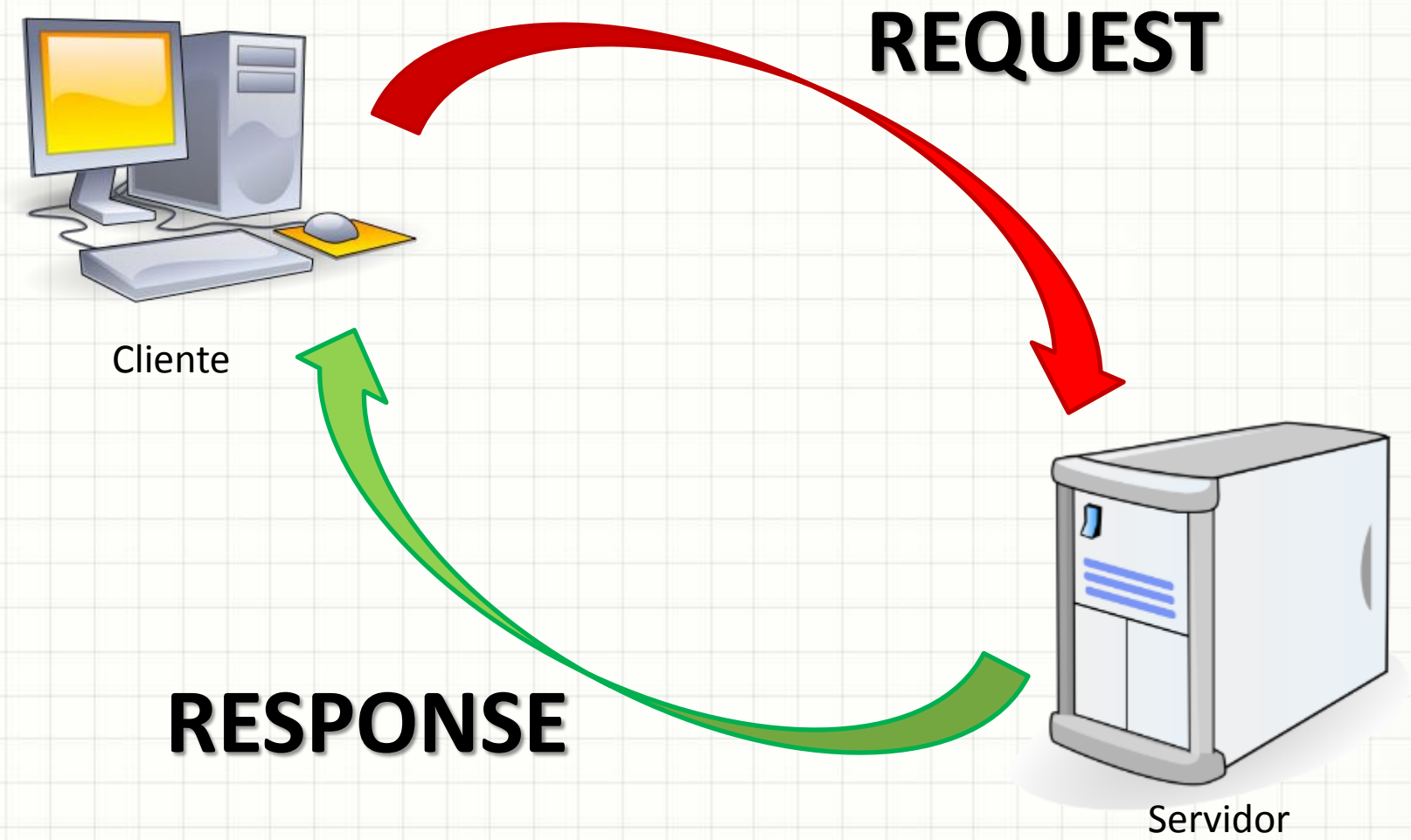
Jogo da Velha

- Navegador **requisita** ao servidor início do jogo:
- O servidor **responde** com o estado do jogo:



Jogo da Velha

- Requisição x Resposta



Jogo da Velha

- Requisição:

  start=x ou start =o

- Resposta:

_ _ _ _ _ _ _ _ _ _ _

Jogo da Velha

- Requisição:

  start=x ou start =o

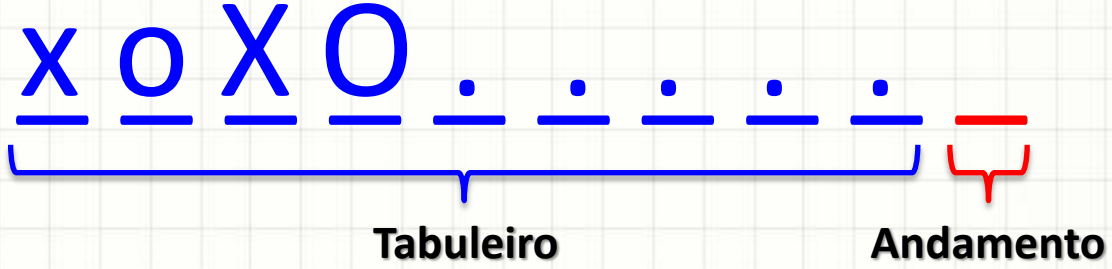
- Resposta:

Jogo da Velha

- Requisição:

  start=x ou start =o

- Resposta:



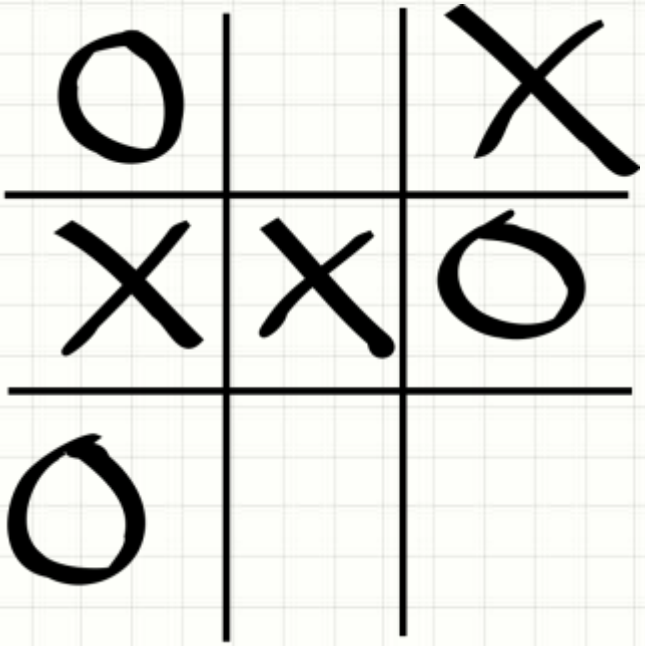
Jogo da Velha

- Navegador terá que pintar o tabuleiro!

O . X X X O O . .

Tabuleiro

Andamento



Jogo da Velha

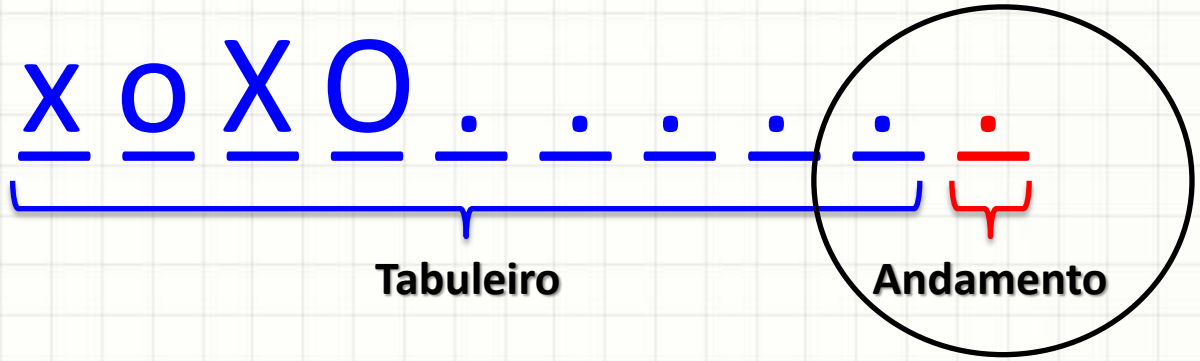
A. Sumir com fundo cinza das áreas de jogo
B. Tornar “invisível” elemento de seleção
C. Criar classes para diferentes marcações
D. Criar função pintaTabuleiro()
   - Pintar tabuleiro segundo texto
   - Ligar visibilidade da seleção só para campos vazios
E. Chamar pintaTabuleiro() no enviaRequest

Jogo da Velha

- Requisição:

  start=x ou start =o

- Resposta:

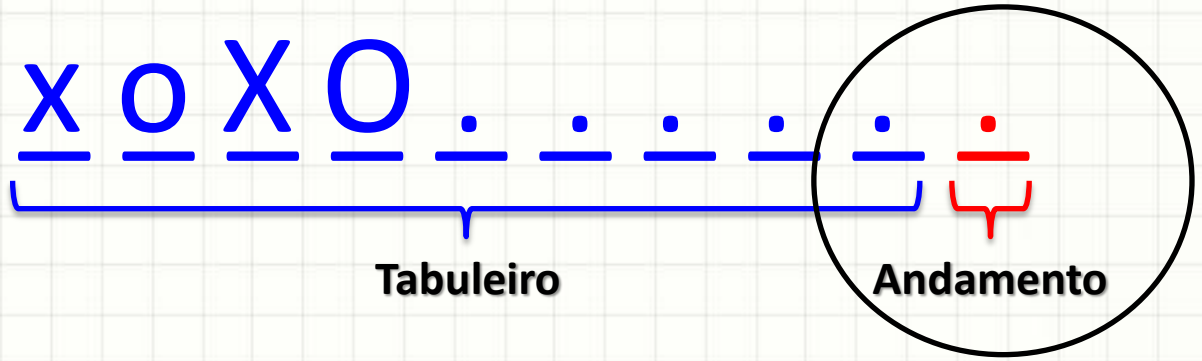


Jogo da Velha

- Requisição:

  start=x ou start =o

- Resposta:



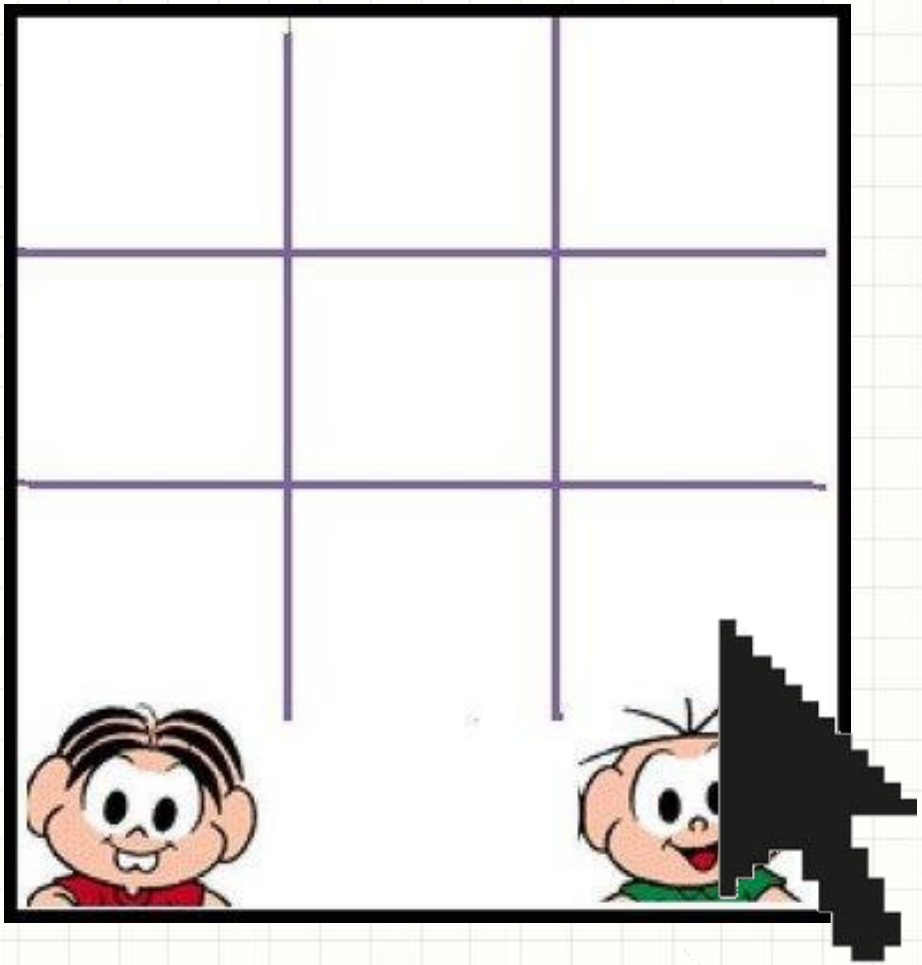
v d e

Jogo da Velha

A. mostraJanela com diferentes status de jogo

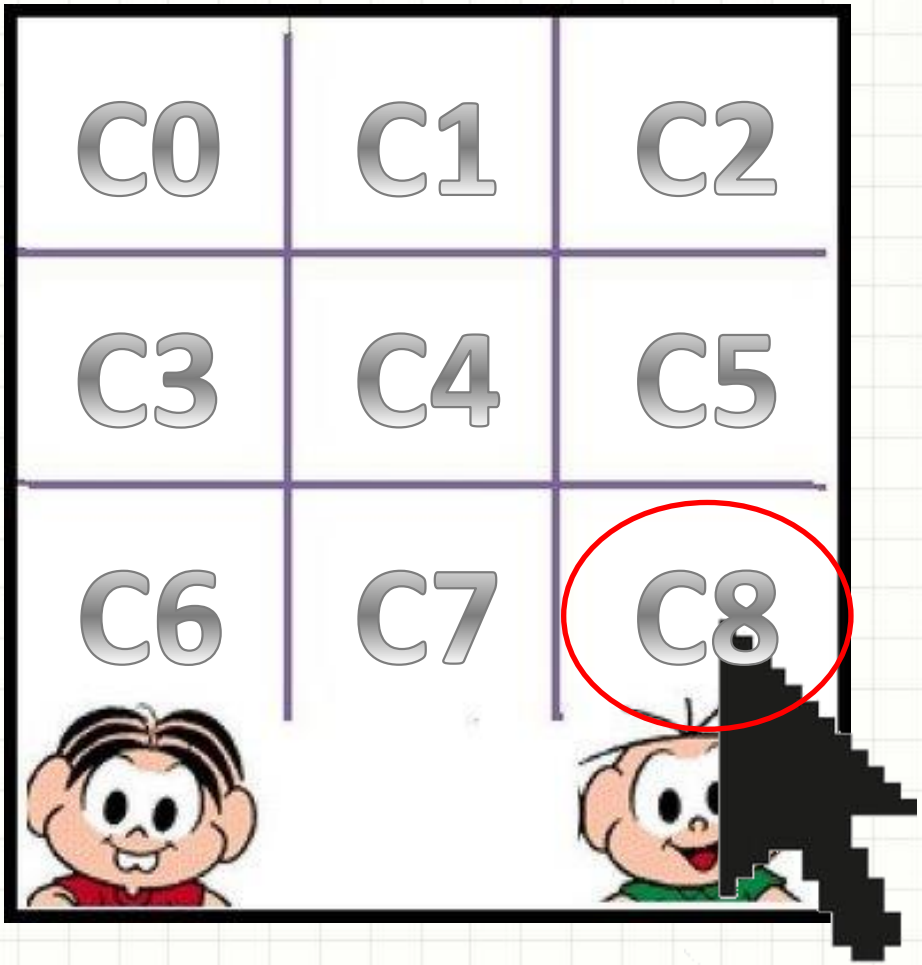
Jogo da Velha

- Lógica de funcionamento



Jogo da Velha

- Lógica de funcionamento



Jogo da Velha

- Requisição:

  cel=c**n**

- Resposta:

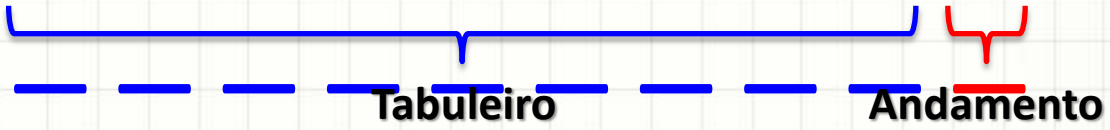
_ _ _ _ _ _ _ _ _ _ _

Jogo da Velha

- Requisição:

  cel=c**8** ... cel=c**n**

- Resposta:



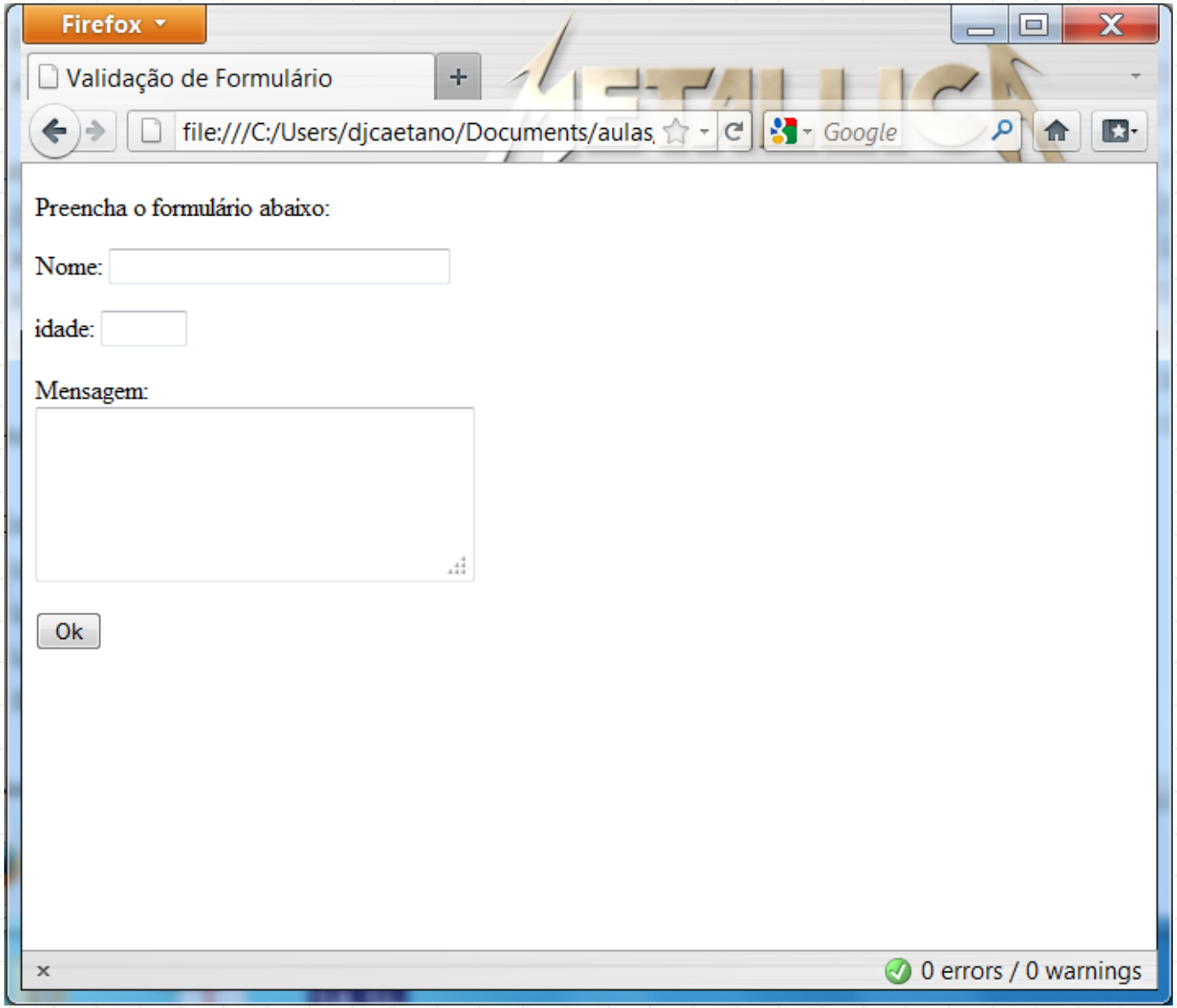
Jogo da Velha

A. Implementar função click (muda fundo)
   - Habilitar **click** só para campos vazios

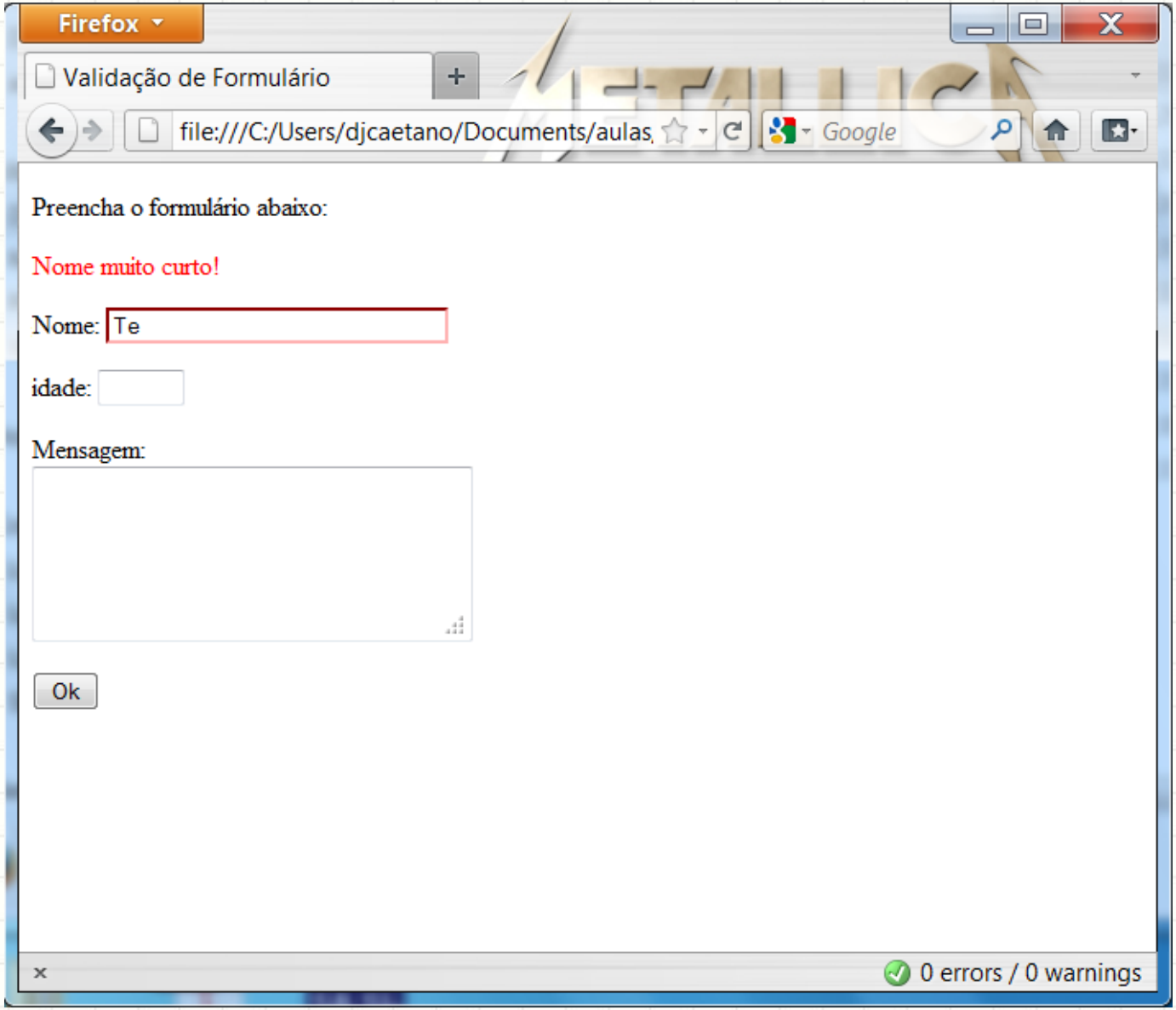
Tutorial: Validação de Formulário

(Opcional)

Validação com JavaScript + CSS



Validação com JavaScript + CSS



Validação com JavaScript + CSS

- Não esqueçamos de:

A. Marcar o campo ativo com “:focus”
B. Marcar o campo “errado” com borda vermelha
C. Ativar o foco no campo “errado” com “.focus()”
D. Marcar texto do campo “errado” com “.select()”
E. Desligar o botão de envio quando for clicado
F. Mudar o texto do botão para “Enviando...”
G. Opcional: contador de caracteres do TextArea
   - Evento .onkeyup

Conclusões

Resumo

- JavaScript: processam. simples no lado do cliente
- JavaScript para modificar CSS:
  - Torna as páginas mais dinâmicas
  - “Sensação”: evita os tempos de espera da rede
- RIA **sempre** terrão código no lado do cliente
  - Objetivo: aumentar velocidade e dinamismo!

- **TAREFA**
  - **Trabalho A!**

Próxima Aula

- JavaScript e CSS só fazem essas coisas?
  - Sim e não!
  - DHTML!
  - Não vai ficar sobre !

PERGUNTAS?

Bom Descanso a Todos!