



# **PROGRAMAÇÃO PARA INTERNET RICA**

## **FUNDAMENTOS DO JQUERY**

Prof. Dr. Daniel Caetano

2012 - 2

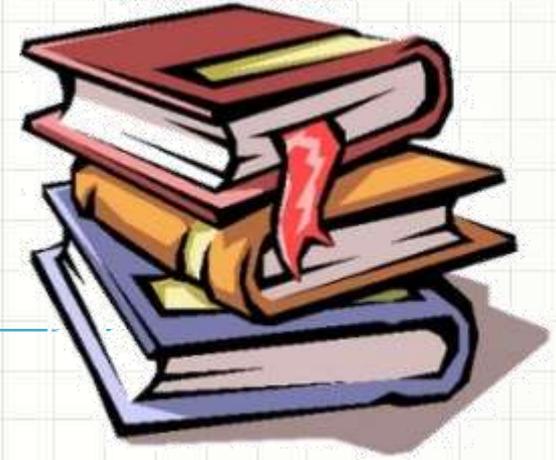
# Objetivos

- Introduzir o framework JQuery
- Capacitar para a construção de documentos JQuery simples
- Compreender os mecanismos mais complexos de acesso do JQuery

- **TRABALHO B!**



# Material de Estudo



## Material

## Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/>  
(Aula 9)

Apresentação

<http://www.caetano.eng.br/>  
(Aula 9)

Material Didático

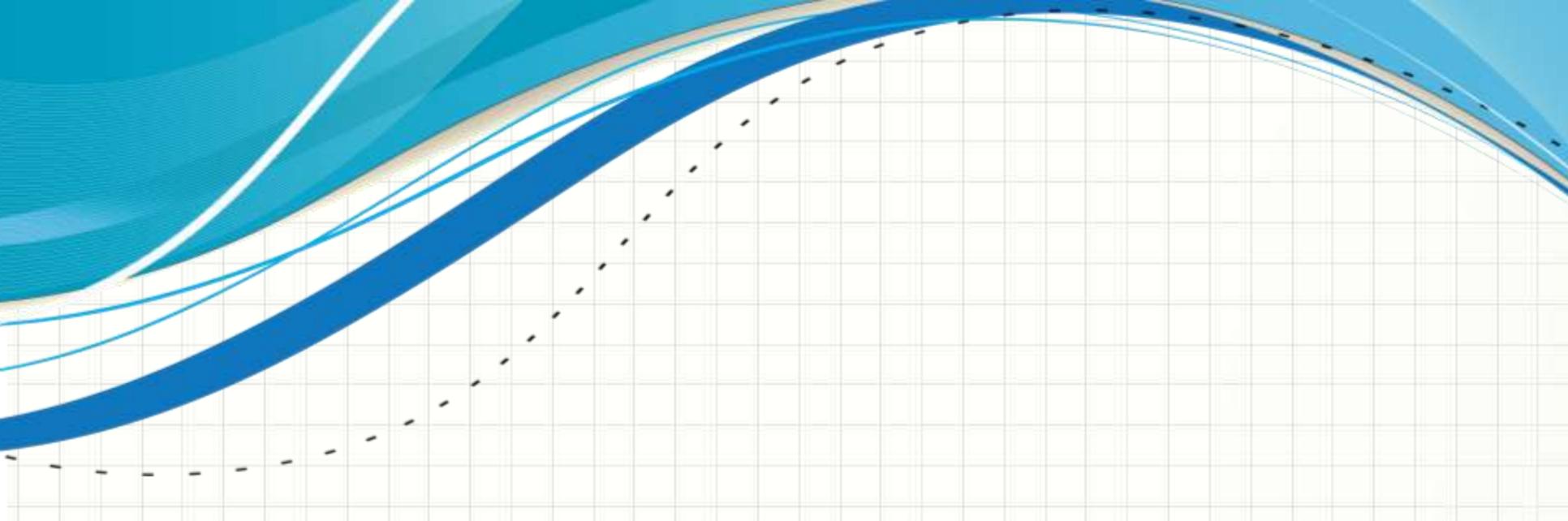
-

Google

+“jQuery” +tutorial

Web Sites

<http://www.jquery.com/>



# O QUE É JQUERY?

# Introdução

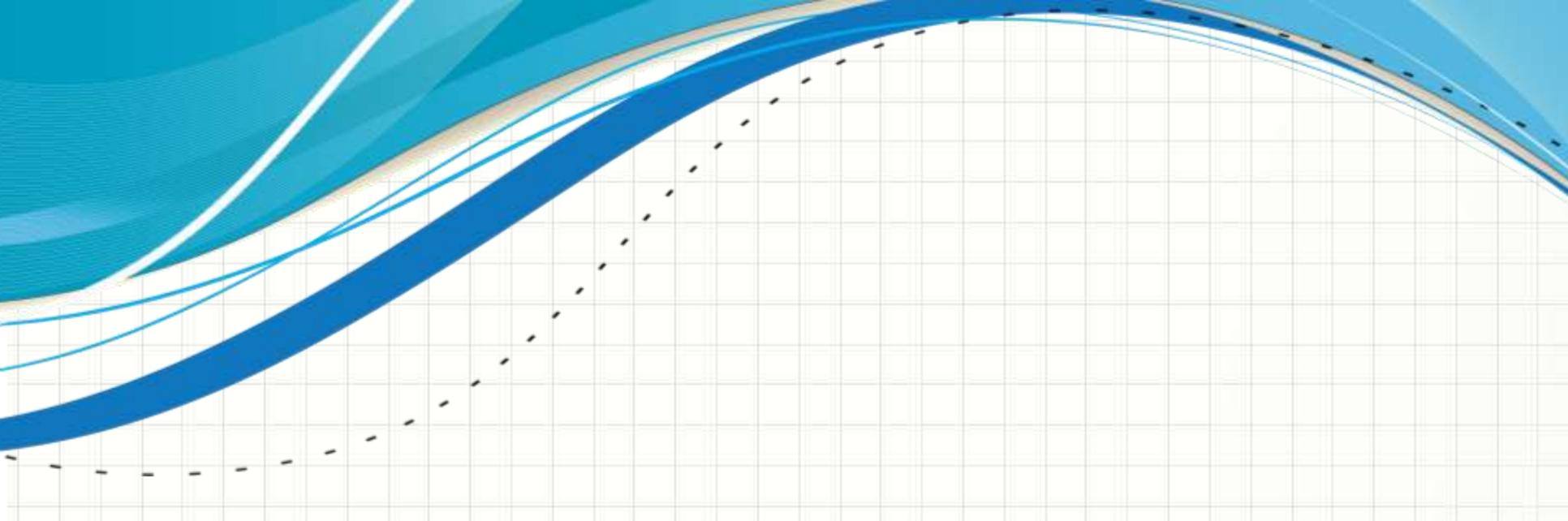
- Métodos diretos para manipular DOM/ CSS
- Tarefa simples?
- Em muitos casos, sim...
- Para todos os outros, use JQuery!
  
- Framework de Desenvolvimento JavaScript
  - Biblioteca x Framework
  - Framework: impõe regras sobre fluxo da aplicação

# Introdução

- JQuery foi criada para **simplificar**
- Cerca de 1/3 dos sites mais visitados usam
- 6 Características Básicas
  - Cross Browser
  - Controle total sobre o DOM
  - Manipula eventos JavaScript
  - Manipula regras do CSS
  - Aplica efeitos visuais
  - Facilita o uso de AJAX

# Introdução

- JQuery implementa buscas na árvore DOM
  - JQuery = JavaScript **Query** = Busca JavaScript
- Busca de elementos DOM facilitada
- Aplicar propriedades a diversos elementos simultaneamente (resultados da busca)



# TRABALHANDO COM JQUERY

# Trabalhando com o JQuery

- **Primeiro** passo: download

<http://jquery.com/>

- Development: cerca de 250KB
- Production: cerca de 30KB
- Verifique sempre: existência de novas versões
  - Corrigem problemas de compatibilidade

# Trabalhando com o JQuery

- **Segundo** passo: inclusão no código

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

- Algumas bibliotecas já incluem o JQuery

**Ex.: Google AJAX Libraries API**

```
<script type="text/javascript"  
    src="http://www.google.com/jsapi/"></script>
```

# Trabalhando com o JQuery

- **Terceiro** passo: comando de busca
- **\$()** → Seletor JQuery

```
$("#body").css("background-color","blue");
```

- Fim do “getElementById” para mudar CSS

Seletor	CSS	JQuery	Descrição
Elemento	p	\$("#p")	Afeta todos os parágrafos do documento
ID	#umId	\$("#umId")	Afeta o elemento que possui o id “umId”
Classe	.aClasse	\$(".aClasse")	Afeta todos os elementos que possuam a classe “aClasse”

# Trabalhando com o JQuery

- **Quarto** passo: garantindo funcionalidade

```
$(document).ready( function() {  
});
```

- Código dentro dessa função...
  - Executada após completo carregamento do DOM
  - Não espera carregamento total do CSS ou imagens
- Similar a...
  - `window.onload = init;`
  - `function init() { ... }`

# Trabalhando com o JQuery

- Modificando valores do CSS

```
seletorJQ.css("propriedade","valor");
```

- Modifica estilos **inline**

- Códigos similares:

```
$("#id").css("background-color","red");
```

```
document.getElementById("id").style.backgroundColor = "red";
```

# Trabalhando com o JQuery

- Modificando valores de atributos HTML

```
seletorJQ.attr("propriedade","valor");
```

```
$("#id").attr("title","Clique para obter ajuda");
```

```
document.getElementById("id").title = "Clique para obter ajuda";
```

# Trabalhando com o JQuery

- Adicionando Classes

```
seletorJQ.addClass("classe");
```

- Removendo Classes

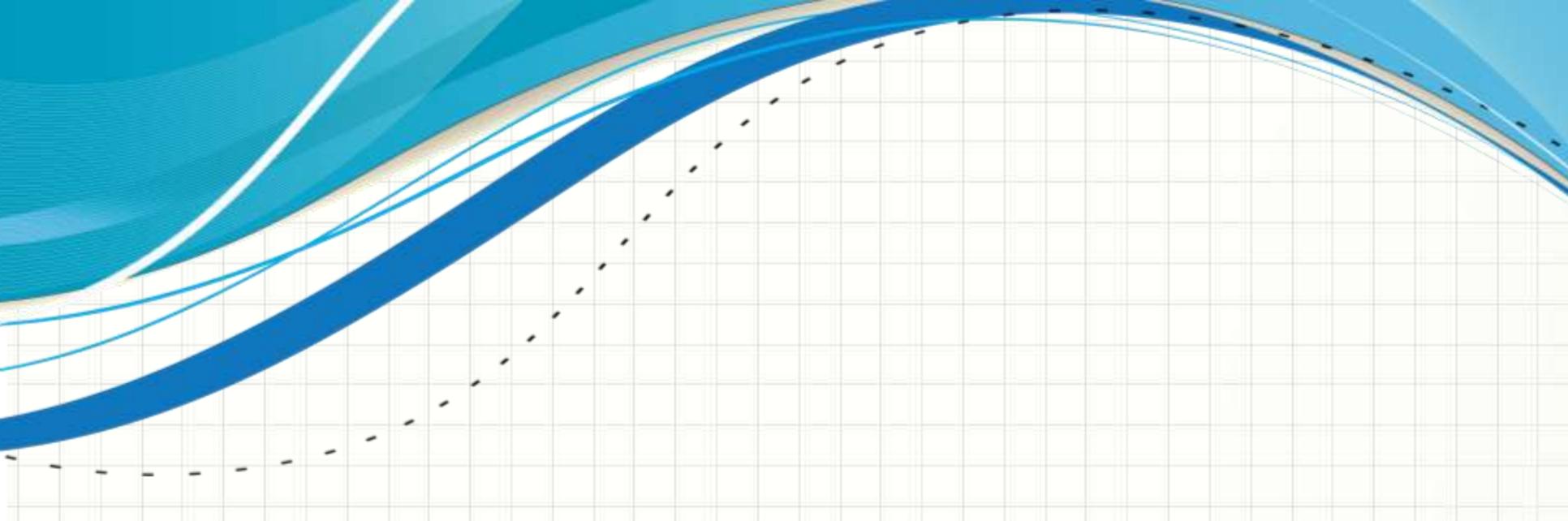
```
seletorJQ.removeClass("classe");
```

# Exemplo JQuery com #id/.class/el

- Acompanhe o Professor
  - index.html
  - estilo.css
  - efeitos.js

# Atividade JQuery

- O método **fadeIn()** varia a opacidade de um elemento de 0 a 100%, além de ativar o **display: block**
- O método **fadeOut()** varia essa opacidade de 100% a 0%, além de ativar o **display: none**
- Sabendo isso, construa um exemplo de PopUp similar ao da aula 6 (mostra/esconde) usando os efeitos **fadeIn()** e **fadeOut()** do JQuery.



# SINTAXE AVANÇADA

# Acessando elementos filhos

- **selectorJQ.children("tag")**
- Modifica a propriedade de todos os elementos "tag" que sejam "filhos" de **selectorJQ**.

```
$("#id").children("p").css("color","green");
```

# Acessando elementos filhos

- Seletores compostos... como no CSS
- Todos os parágrafos dentro do elemento #id  
`$("#id p").css("color","green");`
- Parágrafos filhos diretos do elemento #id  
`$("#id > p").css("color","green");`
- Parágrafo adjacente ao elemento #id  
`$("#id + p").css("color","green");`

# Encadeamento

- Podemos encadear elementos JQuery:

```
selectorJQ.children("tag").fadeOut();
```

- Exemplo

```
$("#id").children("p").fadeOut();
```

- Aplicará um fade-out em todos os parágrafos do div "id".

# Agrupamento de Seletores

- Podemos somar o resultado de várias buscas

```
$(sel1, sel2, sel3, sel4, ..., selN)
```

- Exemplo

```
$("#id, p, .centro").css("color","blue");
```

- Muda a cor do texto dos parágrafos, do elemento de id "id" e dos elementos cuja classe "centro" está definida



**FILTROS**

# Filtrando Elementos na Busca

- Por padrão, JQuery busca todos os elementos de uma tag ou classe
- Como filtrar?
  - Primeiro elemento
  - Elementos pares
  - Etc...?
- Vejamos alguns deles!

# Filtrando Elementos na Busca

- Primeiro elemento

```
$(“p:first”).css(“color”,“green”);
```

- Último elemento

```
$(“p:last”).css(“color”,“green”);
```

- Elementos Pares

```
$(“p:even”).css(“color”,“green”);
```

- Elementos Ímpares

```
$(“p:odd”).css(“color”,“green”);
```

# Filtrando Elementos na Busca

- Elemento específico (4º, por exemplo)

```
$(“p:eq(4)” ).css(“color”, “green”);
```

- Elemento que contém um texto

```
$(“p:contains(texto)” ).css(“color”, “green”);
```

- Elementos Ocultos

```
$(“p:hidden” ).css(“color”, “green”);
```

- Elementos Visíveis

```
$(“p:visible” ).css(“color”, “green”);
```

# Filtrando Elementos na Busca

- Elementos vazios

```
$(“p:empty”).css(“color”,“green”);
```

- Elementos que contenham outro elemento (**ênfase**, por exemplo)

```
$(“p:has(em)”).css(“color”,“green”);
```

# Filtrando Por Atributos

- Filtrar por atributos como id, class, href... etc.
- Todos elementos que contenham um atributo (**id** por exemplo)

```
$(“p[id]”).css(“color”,“green”);
```

- Todos elementos que contenham um atributo com um valor específico

```
$(“p[class=‘aClasse’]”).css(“color”,“green”);
```

# Filtrando Por Atributos

- Todos elementos que contenham um atributo com um valor **diferente** do especificado

```
$(“p[class!=‘aClasse’]”).css(“color”,“green”);
```

- Acumulação de atributos: vários ao mesmo tempo

```
$(“p[class=‘aClasse’][href=‘link’]”).css(“color”,“green”);
```

# Filtros para Formulários

- Filtrar por “type” e outros elementos

- Campos de entrada

```
$(“:input”).css(“color”,“green”);
```

- Campos de texto

```
$(“:text”).css(“color”,“green”);
```

- Campos de Password

```
$(“:password”).css(“color”,“green”);
```

# Filtros para Formulários

- Campos checkbox

```
$(":checkbox").css("color","green");
```

- Campos habilitados

```
$(":enabled").css("color","green");
```

- Campos desabilitados

```
$(":disabled").css("color","green");
```

- Campos marcados

```
$(":checked").css("color","green");
```

- Campos seleccionados

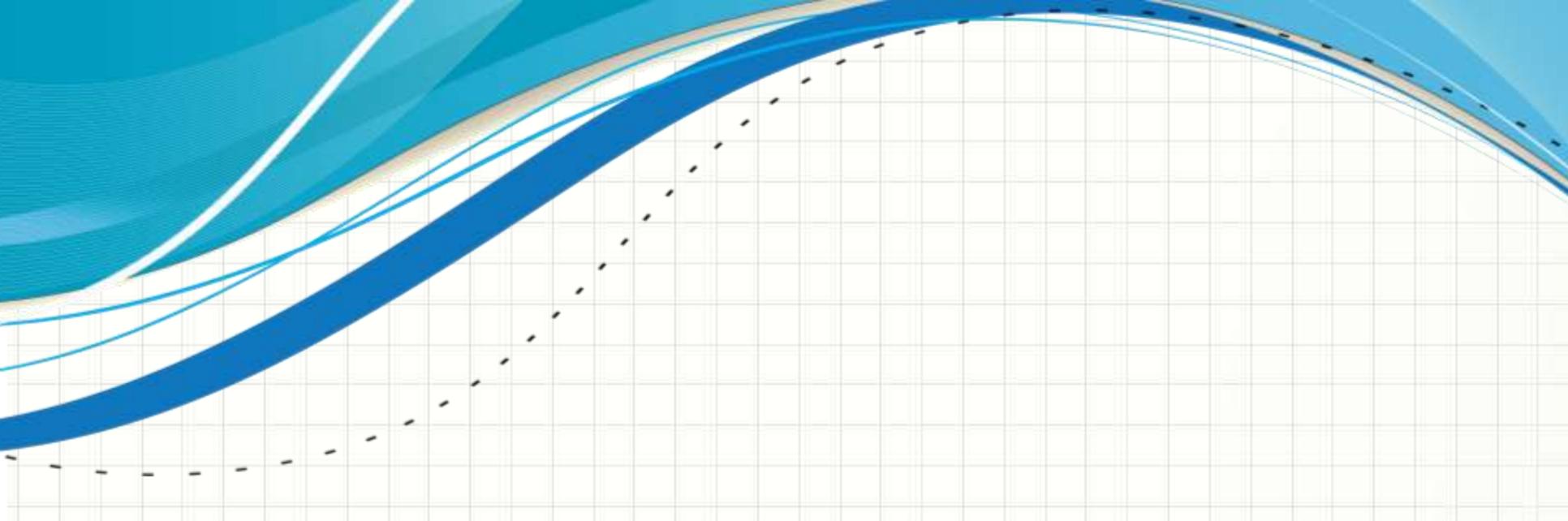
```
$(":selected").css("color","green");
```



**ATIVIDADE**

# Atividade

- 1) Acompanhe o professor na construção do PopUp com imagem e rollUp
- 2) Acompanhe o professor na construção dos diversos exemplos de busca
- 3) Modifique o Jogo da Velha da aula passada para que a janela apareça com FadeIn / FadeOut



**CONCLUSÕES**

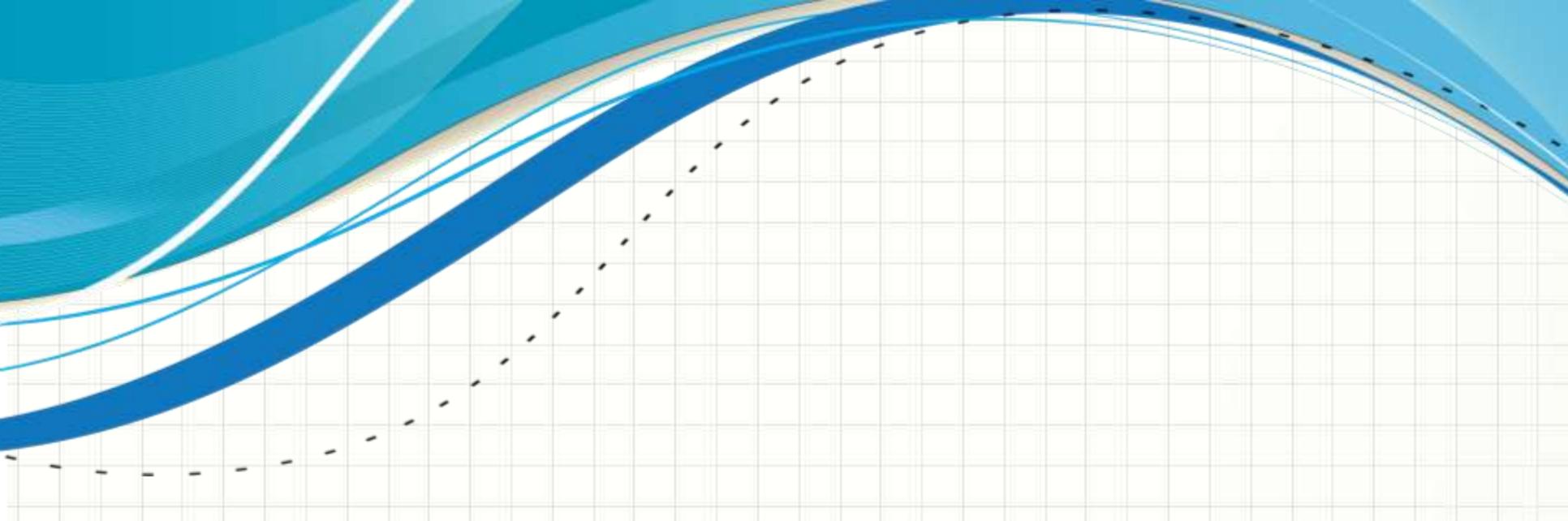
# Resumo

- JQuery: framework para desenvolvimento JS
- Facilita o acesso a elementos
  - Modificar múltiplos elementos simultaneamente
- Acesso padronizado aos elementos
  - Funcionam em todos os navegadores modernos
- **TAREFA**
  - Trabalho B!

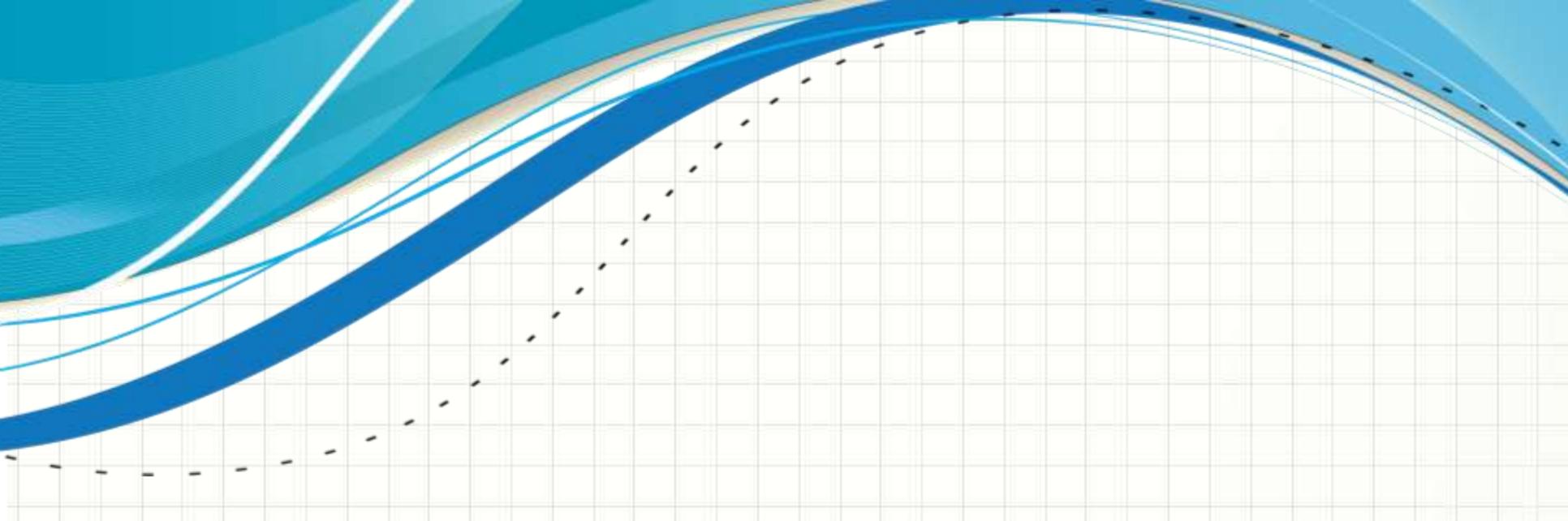
# Próxima Aula



- Como usar o JQuery com ainda mais flexibilidade?
  - Construir a Interface com o Usuário
  - Aplicação Web: Jogo da Velha



**PERGUNTAS?**



**BOM DESCANSO  
A TODOS!**