

# jQuery 简介

# jQuery 是什么？

## jQuery ( Write less, do more.)

jQuery是一个快速、小巧且功能丰富的JavaScript库。它使HTML文档的遍历和操作、事件处理、动画和Ajax等操作变得更加简单，并提供了一个易于使用的API，可以跨多种浏览器工作。jQuery集多功能性和可扩展性于一身，改变了数百万人编写JavaScript的方式。

changed the way that millions of people write JavaScript.

# 通过ID获取一个HTML元素

## JavaScript

```
var obj = document.getElementById(`IdValue`);
```

## jQuery

```
var obj = $('#IdValue');
```

# 将一个HTML元素隐藏

## JavaScript

```
document.getElementById(`IdValue`).style.display = `none`;
```

## jQuery

```
$('#IdValue').hide();
```

# 把一个HTML元素的宽度变成200px

## JavaScript

```
document.getElementById(`IdValue`).style.width = `200px`;
```

## jQuery

```
$('#IdValue').css(`width`, `200px`);
```

# jQuery 到底是什么？

jQuery是一个JavaScript函数库

jQuery是一个轻量级的“写得少，做的多”的JavaScript库

# jQuery的历史

## jQuery之父

John Resig , jQuery的创始人和技术领袖 , 目前在Mozilla担任JavaScript工具开发工程师。



# jQuery的历史

**2005年8月**

**John Resig提议改进Prototype的“Behaviour”库，于是他在blog上发表了自己的想法，并用了3个例子做简单的流程说明。**



# jQuery的历史

## 案例一：为元素注册一个事件

```
Behaviour.register({  
  '#example li': function(e){  
    e.onclick = function(){  
      this.parentNode.removeChild(this);  
    }  
  }  
});
```



```
$('#example li').bind('click',function(){  
  this.parentNode.removeChild(this);  
});
```

# jQuery的历史

## 案例二：为不同的元素注册不同的事件

```
Behaviour.register({  
  'b.someclass' : function(e){  
    e.onclick = function(){  
      alert(this.innerHTML);  
    }  
  },  
  '#someid u' : function(e){  
    e.onmouseover = function(){  
      this.innerHTML = "BLAH!";  
    }  
  }  
});
```



```
$('b.someclass').bind('click',function(){  
  alert(this.innerHTML);  
});  
$('#someid u').bind('mouseover',function(){  
  this.innerHTML = 'BLAH!';  
});
```

# jQuery的历史

## 案例三：为不断变化的元素注册不同的事件

```
Behaviour.register({  
  '#foo ol li': function(a) {  
    a.title = "List Items!";  
    a.onclick =  
function(){ alert('Hello!'); };  
  },  
  '#foo ol li.tmp': function(a) {  
    a.style.color = 'white';  
  },  
  '#foo ol li.tmp .foo': function(a) {  
    a.style.background = 'red';  
  }  
});
```



```
$('#foo ol li')  
  .set('title','List Items!')  
  .bind('click',function(){ alert('Hello!'); })  
  .select('.tmp')  
  .style('color','white')  
  .select('.foo')  
  .style('background','red');
```

# jQuery的历史

**2006年1月14日**

**John Resig 正式宣布以jQuery的名称发布自己的程序库。**

**2006年8月**

**jQuery的第一个稳定版本，并且已经支持CSS选择符、事件处理和AJAX交互。**

# jQuery的作用

◆ HTML 元素选取

◆ CSS 操作

◆ JavaScript 特效和动画

◆ AJAX

◆ HTML 元素操作

◆ HTML 事件函数

◆ HTML DOM 遍历和修改

◆ Utilities

# jQuery的目的

**化大为小，化繁为简**

**简化代码，使程序更高效。**

**JavaScript**

**jQuery主要是用来替代原生JavaScript的。**

# jQuery的版本



# 学前准备

- ◆ 掌握 HTML 基本知识
- ◆ 具备 CSS 基础
- ◆ 熟练使用 JavaScript 及其 DOM 操作
- ◆ 有耐心和毅力，勤于动手



# jQuery 的优势

- ◆ 轻量级
- ◆ 强大的选择器
- ◆ 出色的DOM操作及其封装
- ◆ 可靠地事件处理机制
- ◆ 完善的Ajax

# jQuery 的优势

◆ 不污染顶级变量

◆ 出色的浏览器兼容性

◆ 链式操作方式

# jQuery 的优势

◆ 隐式迭代

◆ 行为层和结构层的分离

◆ 丰富的插件支持

# 课程概述

◆ 使用 jQuery

◆ 操作 DOM

◆ jQuery 动画

◆ jQuery 扩展

◆ 选择器

◆ 操作事件

◆ Ajax

# 学习目标

- 1、掌握 jQuery 的核心功能特点，并熟练应用
- 2、更加深入了解网页结构层、表示层以及行为层的功能特性
- 3、有效提高开发效率，并学会组件化与模块化的封装思路
- 4、为面向对象开发学习打好基础