

讲义 11: Prototype 与 DOM

本课程将探讨 JavaScript 框架的学习方法，在框架的基础上动态的处理 DOM 模型，实现动态网页。最后，学习利用正则表达式在客户端验证用户的输入。

课程目标：

- 1) 掌握 Prototype 框架，有效处理 DOM 模型。
- 2) 正则表达式的使用。

使用工具：

Notepad++ or sublime;
chrome browser;
python2.7; Tornado 3.1;

课程资源与要点：

- 1) W3 教程 JS: **JS**

<http://www.w3schools.com/js/default.asp>

<http://w3school.com.cn/js/index.asp> (中文)

--- JS Advanced: 对象。

--- JS HTML DOM: JavaScript RegExp 对象，正则表达式语法及应用

- 2) Google Hosted Web Lib: **google web**

<https://developers.google.com/speed/libraries/devguide?hl=zh-CN>

- 3) Prototype 入门教程: **prototype**

<http://prototypejs.org/learn/>

--- Defining classes and inheritance

--- How Prototype extends the DOM

--- Event delegation

任务 1: 准备知识

通过前面的课程，如果你认为 JavaScript 和 DOM 模型比较简单，那就大错特错了。JavaScript 和 DOM 是应用超前标准的典型代表。经过多年的发展，它的“碎片”化程度不是安卓这样的系统可比拟的。为了把程序员从主流浏览器平台差异中解放出来，第一流行的“补鞋匠”出现了，它就是 Prototype。

如果你认为学了 Prototype 就可以开发高质量的 Web 应用，那也不对。更多更新的一推 JavaScript 框架在等你，它们各有秋季，你甚至无法选择。所以，现在普通程序员中，**JavaScript 程序员工资是最高的（易学难精）**。怎么办，如何制定学习目标呢？一般来说，掌握 **jQuery 是比较实用的道路**。难道那么多框架存在没有理由吗？“存在及价值”的真理颠覆了吗？

任务 1.1: 看看维基百科

[http://zh.wikipedia.org/wiki/Prototype_\(%E6%A1%86%E6%9E%B6\)](http://zh.wikipedia.org/wiki/Prototype_(%E6%A1%86%E6%9E%B6))

这里仅有一些 JavaScript 框架的不完全列表。这是学不完的。

- 1) **掌握学习方法才是正途。**
- 2) Prototype 的源代码是最少、最容易看懂的。
- 3) Prototype 的思想和方法，几乎是后续框架的基础。

如果你打算称为JavaScript大神级别的专家,请不断尝试阅读Prototype源代码。
它的思想方法,是你称为客户端专家的基础。
让我们学一些准备知识,以方便我们的学习。

任务 1.2: 练习、调试平台

w3school 练习平台,你可以试一下:

http://w3school.com.cn/tiy/t.asp?f=js_library_prototype2

在这个平台上可以测试许多学到的东西

调试平台: 开发者工具 (Console)

任务 1.3: JavaScript 对象与函数

http://w3school.com.cn/js/js_objects.asp

本部分要点:

- 1) JavaScript 内置了一些标准对象
- 2) 定义对象属性的基本方法 (new Object() / 字典法)
- 3) 函数构造法!!! 重点。this 的含义 (注意 DOM 绑定事件后, 也有 this)
- 4) for .. in .. 循环

任务 1.4: JavaScript RegExp 对象

正则表达式用法, 参考: http://w3school.com.cn/js/js_obj_regexp.asp

本部分要点:

- 1) 正则表达式对象的定义, 例如

```
var word = /[a-z]+/i;
```

- 2) test, exec, compile 三个方法

正则表达式语法, 参考: http://w3school.com.cn/js/jsref_obj_regexp.asp

本部分要点: 简单的正则表达式

- 3) 如何检测一个邮箱、URL、邮编的合法性

百度: “正则表达式大全”

任务 2: Prototype 入门

任务 2.1: Defining classes and inheritance

参见: <http://prototypejs.org/learn/class-inheritance.html>

本部分要点:

- 1) 运用对象 “prototype” 属性, 构建类似 java 等面向对象语言的定义属性和方法
的用法。
- 2) 对象 “继承”

```
Pirate.prototype = Object.extend(dest,source);
```

注意: JavaScript 对象就是一个字典, Object.extend 就是合并这个字典到目标对象

源代码阅读 (提高要求): 点击 “`Object.extend`” 链接, 进入 prototype APIs 参考手册。在 GitHub 上阅读源代码。

```

56  /**
57   * Object.extend(destination, source) -> Object
58   * - destination (Object): The object to receive the new properties.
59   * - source (Object): The object whose properties will be duplicated.
60   *
61   * Copies all properties from the source to the destination object. Used by Prototype
62   * to simulate inheritance (rather statically) by copying to prototypes.
63   *
64   * Documentation should soon become available that describes how Prototype implements
65   * OOP, where you will find further details on how Prototype uses [[Object.extend]] and
66   * [[Class.create]] (something that may well change in version 2.0). It will be linked
67   * from here.
68   *
69   * Do not mistake this method with its quasi-namesake [[Element.extend]],
70   * which implements Prototype's (much more complex) DOM extension mechanism.
71   */
72  function extend(destination, source) {
73    for (var property in source)
74      destination[property] = source[property];
75    return destination;
76  }

```

注释比代码多，也更难写!!!（二年级同学一定要学习阿）

3) The \$super argument in method definitions

注意：JavaScript 就是大杂烩，编程要点很多，请你仔细看后面官方资料

任务 2.2: How Prototype extends the DOM

参见：<http://prototypejs.org/learn/extensions.html>

本部分要点：

1) \$函数源代码阅读（提高要求）。

<http://api.prototypejs.org/dom/dollar/>

```

60  function $(element) {
61    if (arguments.length > 1) {
62      for (var i = 0, elements = [], length = arguments.length; i < length; i++)
63        elements.push($(arguments[i]));
64      return elements;
65    }
66    if (Object.isString(element))
67      element = document.getElementById(element);
68    return Element.extend(element);
69  }

```

2) 将 Element 类的内容合并到 DOM 对象中

Element.extend(element)

你可以参考源代码，定义自己处理 DOM 元素的类，把它注入到元素中？

注意：千万不要认为你很牛，学了 Object.extend(element, Element)就这样搞定了，看了别人的代码，就知道自己的浅薄了。

3) Adding your own methods with Element.addMethods

任务 2.3: Event delegation in Prototype 1.7

参见：<http://prototypejs.org/learn/event-delegation.html>

本部分要点：

1) \$\$函数

先猜功能，然后自己看 API

2) Prototype's DOM element 事件绑定的方法

百度“观察者模式”（提高要求）

Element.observe()可以在一个事件 handler 上绑定一组处理函数。

注意：请暂不要研究 event 对象，后面有的

任务 3：DOM 遍历与事件

任务 3.1：DOM 遍历

<https://courses.cs.washington.edu/courses/cse190m/09sp/lectures.shtml>

--- walking the DOM tree

任务 3.2：DOM 事件

<https://courses.cs.washington.edu/courses/cse190m/09sp/lectures.shtml>

--- events

任务 4：正则表达式的应用

<https://courses.cs.washington.edu/courses/cse190m/09sp/lectures.shtml>

---more events; client-side validation

作业与练习：

1、无

二、操作与实践

1、Lab 6: Mouse Maze (DOM and Events)

2、Homework 6 (Fifteen Puzzle)