

# JS高级 -- 原型

## 题目

- 说一个原型的实际应用
- 原型如何体现它的扩展性

## 原型的实际应用

- `jquery` 和 `zepto` 的简单使用
  - `jquery` 实例

```
<body>
  <p>jquery test 1</p>
  <p>jquery test 2</p>
  <p>jquery test 3</p>
  <p>jquery test 4</p>
  <div id="div1">
    <p>jquery test in div</p>
  </div>
  <script type="text/javascript" src="https://cdn.bootcss.com/jquery/3.2.1/jquery.js">
</script>
  <script type="text/javascript">
    var $p = $('p');
    $p.css('font-size', '40px'); //css是原型方法
    alert($p.html()) // html也是原型方法
    var $div1 = $('#div1');
    $div1.css('color', 'blue');
    alert($div1.html())
  </script>
</body>
```

- `zepto` 如何使用原型

```
// my-zepto.js
(function(window) {
  var zepto = {};
  function Z(dom, selector) {
    var i, len = dom ? dom.length : 0;
    for (i = 0; i < len; i++) {
      this[i] = dom[i];
    }
    this.length = len;
    this.selector = selector || '';
  }
  zepto.Z = function(dom, selector) {
    return new Z(dom, selector);
  }
  zepto.init = function(selector) {
    var slice = Array.prototype.slice;
    var dom = slice.call(document.querySelectorAll(selector));
    return zepto.Z(dom, selector);
  }
  var $ = function(selector) {
    return zepto.init(selector);
  }
}
```

```

window.$ = $;
$.fn = {
  constructor:zepto.Z,
  css: function(key, value) {
    alert('css')
  },
  html: function(value) {
    return 'html'
  }
}
Z.prototype = $.fn;
})(window)

```

- **jquery** 如何使用原型

```

// my-jquery 简易代码
(function(window) {
  var jQuery = function(selector) {
    return new jQuery.fn.init(selector);
  }
  jQuery.fn = {
    constructor:jQuery,
    css: function(key, value) {
      alert('css')
    },
    html: function(value) {
      return 'html'
    }
  };
  var init = jQuery.fn.init = function(selector) {
    var slice = Array.prototype.slice;
    var dom = slice.call(document.querySelectorAll(selector));
    var i, len = dom ? dom.length : 0;
    for (i = 0; i < len; i++) {
      this[i] = dom[i];
      this.length = len;
      this.selector = selector || '';
    }
  }
  init.prototype = jQuery.fn;
  window.$ = jQuery;
})(window)

```

- 为何要把原型方法放在 **\$.fn** 上?

这样可以实现插件扩展

```

// 插件扩展的例子
$.fn.getNodeName = function(){
  return this[0].nodeName;
}

```

- 好处
  - 只有 **\$** 会暴露在 **window** 全局变量
  - 将插件扩展统一到 **\$.fn.xxx** 这一接口，方便使用