

前端开发基础知识回顾

小豪

maisui99@gmail.com

今年

- * 单元测试
- * HTML5
- * 手机端Web应用开发
- * Node.js迅速发展

前端

- * 运行环境

- * 排版引擎

- * JavaScript解释器

- * 设备API

前端

* 技术

- * HTML

- * CSS

- * JavaScript, (各种类库)

前端

* 独特

- * 兼容性问题

- * 安全性

- * 更关注用户体验

前端

- * 标准

- * W3C

- * ECMAScript 5 (ECMA-262.pdf)

HTML设计

* 回到XHTML

- * 元素名称属性必须小写
- * 属性总是用引号进行包裹
- * 禁止属性的简化
- * 空元素写法
- * 标签的正确嵌套

* 标签的使用

- * Body: The body element represents the main content of the document.
- * Address: Represents the contact information for its enclosing section. If it is a child of the bodyelement, then it applies to the document as a whole.

* 页面的抽象结构

- * 类：div.module

- * 用于通用化

- * 对象：div#colorful-module

- * 用于特殊化

```
<div classs="module" id="colorful-module">
```


<http://www.f2es.com/oocss-css-rules/>

<http://www.f2es.com/web-standard-oocss-share/>

* 避免错误的嵌套

```
<div class="test">  
  <p>这个页面是用以测试主机是否有正确的content-type。  
</div></p>
```

Closing tag matches nothing

-
- * 先考虑语义再考虑样式
 - * 内容优先
- 

- * CSS控制样式，而不是HTML

- *

- *

* 可访问性

- * Alt

- * Title

- * Tabindex

- * 减少窗口

- * ...


<http://rexsong.com/webdesign/accessibility/>

<http://dev.opera.com/articles/view/25-accessibility-basics/>

* 校验

- * W3C has an online validator available

CSS设计

- * 选择器从右到左
 - * 结构从左到右
- 

* 优先级与继承

- * 模块样式
 - * 页面级样式
 - * Reset样式
 - * 浏览器默认样式
- * !important > style > id > class > target
- * 慎用!important

* 盒模型 - 文档流中的布局方式

- * Block

- * Inline

- * Inline-block

- * None

- * more

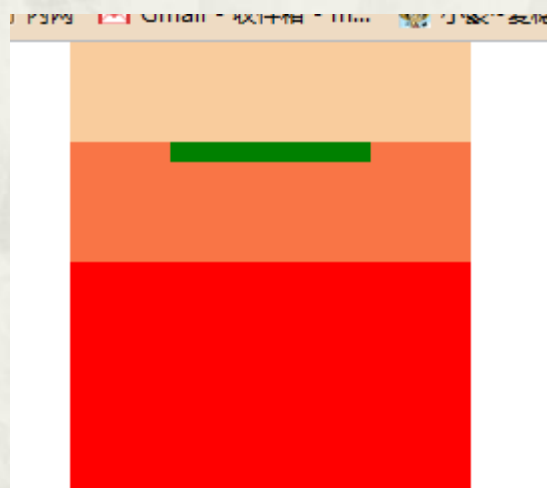
* Block

- * 垂直布局

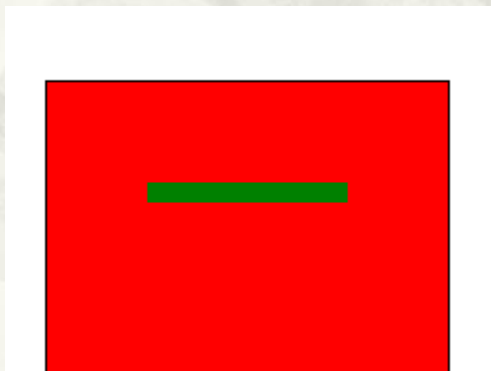
- * Margin控制间距

- * 并列和嵌套的block元素都存在margin合并的问题

```
body>  
<div style="background:red;height:200px;margin:50px;width:200px">  
| <div style="margin:50px;background:green;height:10px"></div>  
</div>  
/body>
```



```
body>  
<div style="background:red;height:200px;margin:50px;width:200px;border:1px solid #000">  
| <div style="margin:50px;background:green;height:10px"></div>  
</div>  
/body>  
/html>
```



当外层的元素存在border和padding的时候，边距合并的现象就会消失

* Inline

- * 水平布局
- * 垂直的margin,padding,border无效
- * 不能指定宽高

- * BFC — block formatting context
 - * 浮动元素(float: left | right);
 - * 绝对定位元素(position: absolute | fixed);
 - * 行内块元素(display: inline-block);
 - * 表格的单元格(display: table-cells, TD、TH);
 - * 表格的标题(display: table-captions, CAPTION);
 - * 'overflow' 特性不为 visible 的元素
 - * 表格元素创建的 “匿名框” 。

* BFC

- * 不存在margin合并的问题
- * 清除内部浮动
- * 不会和float的元素重叠

* IE layout

- * Images
- * Tables, TableRows, TableCells
- * HR
- * Input elements: text, button, file, select
- * Marquee
- * Framesets, Frames
- * Objects, applets, plugins
- * Absolute positioned elements
- * Floated elements
- * Inline-block elements
- * Filters (rotation, dropshadow, etc.)
- * Body

* 激发layout

- * 在strict模式下设置了宽度或者高度的块级元素
- * 在兼容模式下指定了宽度或者高度的元素
- * 在css中设置了zoom属性的
- * 浮动的方向和父级元素浮动的方向不一样

* Layout特点

- * 不围绕其他元素布局
- * 不自适应子元素（子元素也不适应layout元素）
- * 自适应父元素

* 普通定位

- * Position为static, top/left/bottom/right, z-index无效
- * Position为relative, top/left/bottom/right, z-index有效并按照原位置偏移


```
8 <body>
9 <div style="float:left;height:100px;width:100px;background:yellow">
10 </div>
11 <div style="overflow:hidden;*zoom:1">
12 <p>店铺地址：西湖区华星路170号</p>
13 <p>送餐范围：西到紫金港路，北到德胜路，东到新塘路，南到之江路，滨江主城区</p>
14 <p>送餐范围：西到紫金港路，北到德胜路，东到新塘路，南到之江路，滨江主城区</p>
15 <p>送餐范围：西到紫金港路，北到德胜路，东到新塘路，南到之江路，滨江主城区</p>
16 <p>送餐范围：西到紫金港路，北到德胜路，东到新塘路，南到之江路，滨江主城区</p>
17 <p>送餐范围：西到紫金港路，北到德胜路，东到新塘路，南到之江路，滨江主城区</p>
18 <p>送餐范围：西到紫金港路，北到德胜路，东到新塘路，南到之江路，滨江主城区</p>
19 <p>送餐范围：西到紫金港路，北到德胜路，东到新塘路，南到之江路，滨江主城区</p>
20 <p>送餐范围：西到紫金港路，北到德胜路，东到新塘路，南到之江路，滨江主城区</p>
21 </div>
22 </body>
```

* 浮动定位

- * 紧贴容器边缘，没有空间后下移
- * 没有bfc的容器会围绕浮动元素
- * 宽度适应子元素（不设宽度的情况下）
- * 元素内部属性为block
- * top/left/right/bottom, z-index无效

* 清除浮动

```
.clearfix:after { /* 利用生成内容清理浮动 for IE8(标准模式) 和 非IE浏览器 */
  content: "";
  clear: both;
  height: 0;
  visibility: hidden;
  display: block;
}
```

```
.clearfix{ /* 激发IE中的hasLayout属性 for IE8(Quirks)模式和IE6\7浏览器*/
  *zoom:1;
}
```

```
.clear{
  zoom:1;/*激活ie6/7/8(quirks mode)中的layout*/
  overflow:hidden;/*激活Block Formatting Context, 清楚内部元素浮动*/
}
```

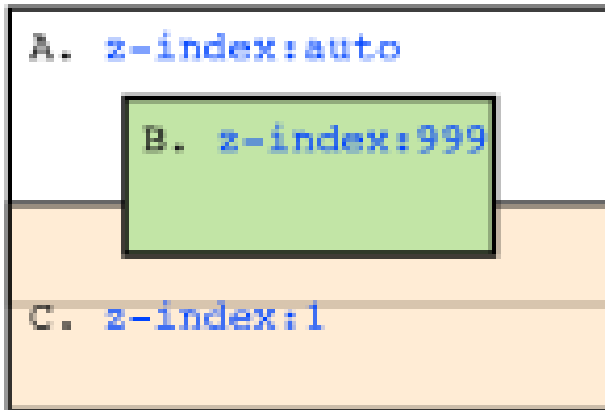
* Absolute Position

- * Float失效
- * 脱离文档流
- * 元素内部属性为block

* Z-index

- * 顺序优先级

- * Auto则保持和父级一致 (ie6/7下hack)



JavaScript

- * JavaScript
 - * Netscap版本
 - * Mozilla
- * Jscript
 - * Microsoft
- * ECMAScript
 - * 参考标准

- * 原始类型

- * Number, String, Boolean

- * 引用类型

- * Function, Array, Date, RegExp

- * Null

- * undefined

* 弱类型

* +new Date = 1324462799561

* + "3" = 3

* 64位二进制表示

```
> 5 * 1.015  
5.0749999999999999  
>
```

* 布尔型数据

- * 0, “”, NaN, null, undefined为false, 其他都为true

- * `![] == []`

- * `ToPrimitive([]) == 0`

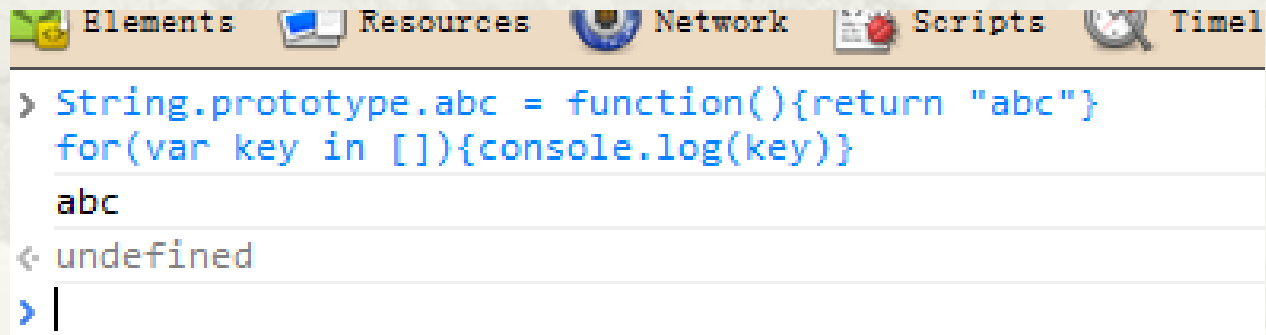
- * 基础类型valueOf, 引用类型toString

- * !运算如果后面为对象则返回false

* Prototype污染

* hasOwnProperty检查

- * 避免将prototype上的属性遍历出来
- * 也可以用于检查某个属性是否存在



```
Elements Resources Network Scripts TimeL
> String.prototype.abc = function(){return "abc"}
  for(var key in []){console.log(key)}
  abc
< undefined
> |
```

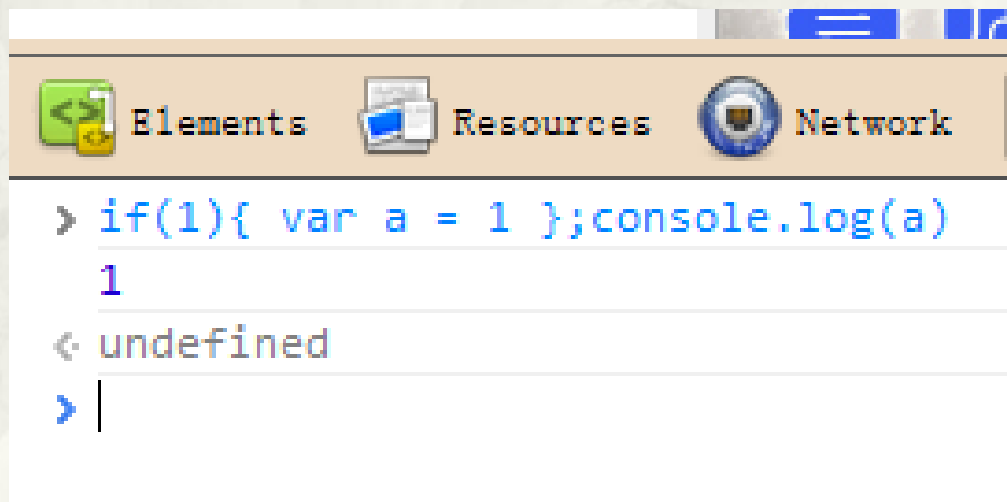
* Window

- * 全局变量

- * 所有没有用var定义的变量直接挂载在window上
(未定义也可以使用的根源)

* 局部变量

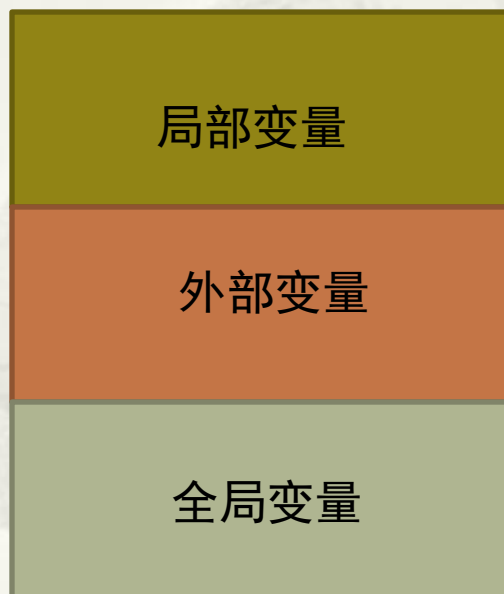
- * 函数体内
- * 非块级作用域



The screenshot shows a browser's developer console with three tabs: Elements, Resources, and Network. The console contains the following text:

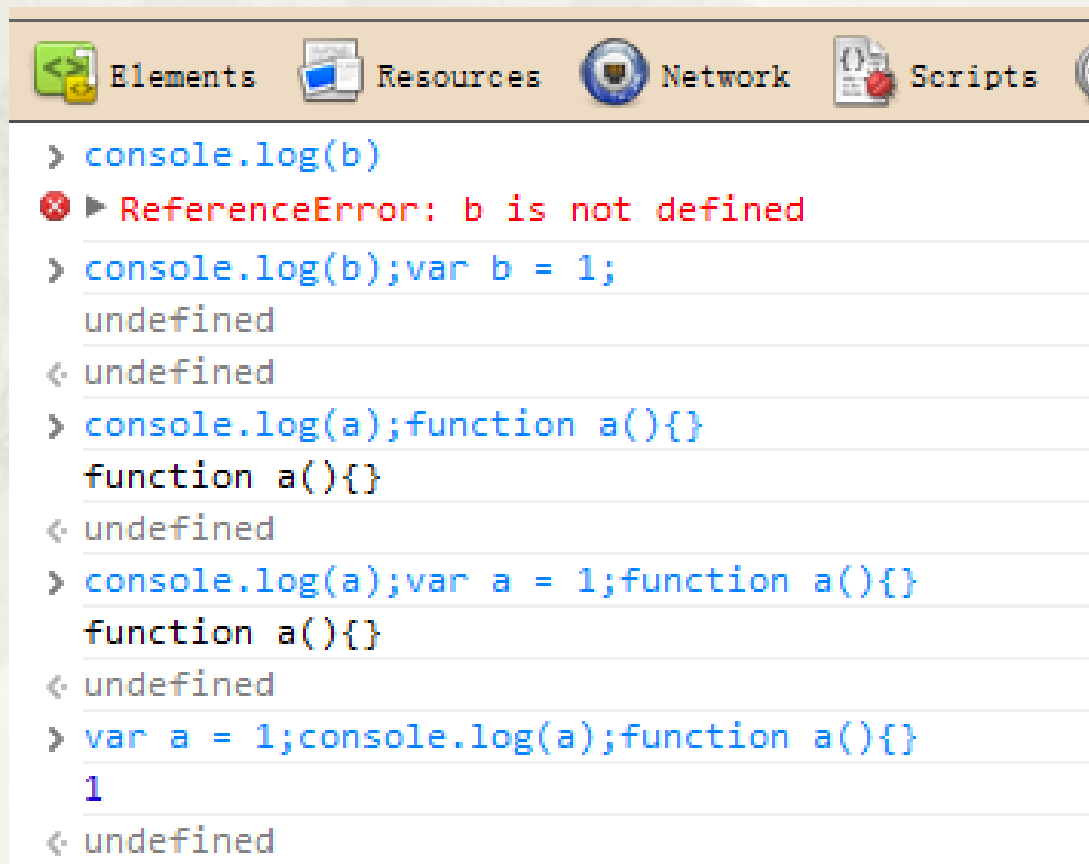
```
> if(1){ var a = 1 };console.log(a)  
1  
← undefined  
> |
```

* 作用域链



```
1 var a = 1;  
2 = function b(){  
3     var c = 1,  
4         w = window,  
5         innerA = a;  
6 }
```

- * 解析顺序
 - * 函数声明
 - * 变量生命
 - * 执行



The screenshot shows a browser's developer console with the following tabs: Elements, Resources, Network, and Scripts. The console output is as follows:

```
> console.log(b)
✖ ▶ ReferenceError: b is not defined
> console.log(b);var b = 1;
undefined
< undefined
> console.log(a);function a(){}
function a(){}
< undefined
> console.log(a);var a = 1;function a(){}
function a(){}
< undefined
> var a = 1;console.log(a);function a(){}
1
< undefined
```

* 函数声明与表达式

* `var a = function(){} // 函数表达式`

* `function a(){} // 函数声明`

* `var a = function b(){} // 这样算什么?`

```
> var a = function b(){}  
undefined  
> b  
✖ ▶ ReferenceError: b is not defined  
> var a = function b(){console.log(b)}  
undefined  
> a()  
function b(){console.log(b)}  
← undefined  
> |
```

* IE

```
1 var a = 1,  
2   b = function c(){  
3     a++;  
4   };
```

```
1 function c(){  
2   a++;  
3 }  
4 var a = 1,  
5   b = function(){  
6     a++;  
7   };
```

* 闭包

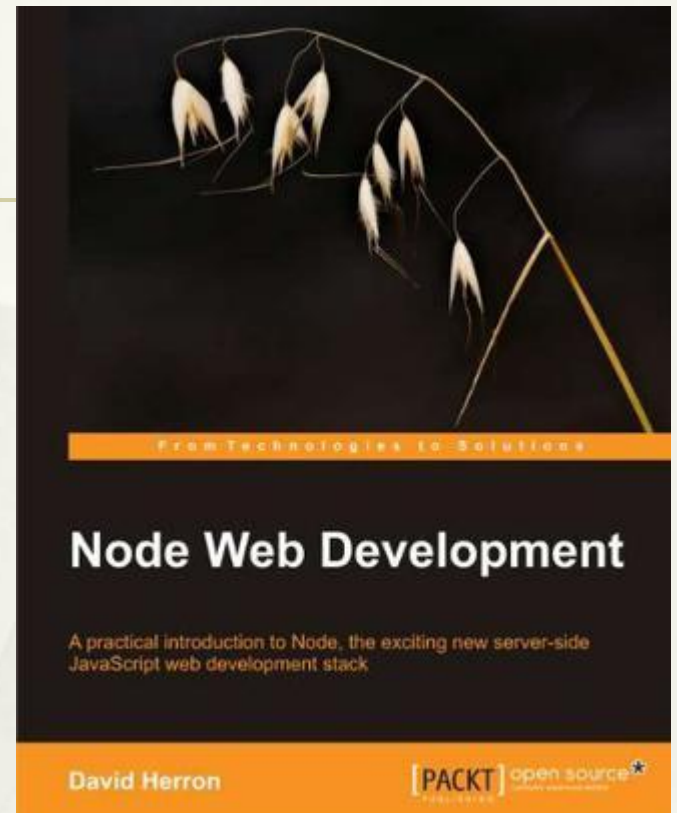
- * Context, 执行期上下文环境
- * 在执行的时候创建

```
1 var a = 1;
2   setTimeout(function(){
3     console.log(a);
4   }, 1000);
5   a = a + 1;
```

* 定时器

- * setTimeout 一定时间后插入执行队列
 - * 比预设的时间长
- * setInterval按照固定时间不断插入
 - * 比预设的时间短

- * V8 + 异步IO + HTTP实现
 - * 用单线程的方式处理请求
 - * 多核cpu
 - * 多个node进程负载均衡
 - * 具体关注 《node web development》



谢谢

小豪

@小豪_步天天天

maisui99@gmail.com