



链滴

Java Web 开发中一些实用的技术点

作者: iTanken

原文链接: <https://ld246.com/article/1483519924205>

来源网站: 链滴

许可协议: 署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

1. Java Integer的缓存:-128 to 127

```
Integer.valueOf(10) == Integer.valueOf(10); // true
Integer.valueOf(128) == Integer.valueOf(128); // false
```

2. Java 中的小数运算, 2-1.1=?

我们知道, $2-1.1=0.9$, 但是在java中如果直接这样运算, 会得到一个意外的结果。一般可以用 `java.math.BigDecimal` 来解决这类问题, 但如果不想写太多代码, 还有一种比较简单方便的解决方式, 直接在小数后面添加一个 `f`:

```
System.out.println(2-1.1); // 0.8999999999999999
System.out.println(2-1.1f); // 0.9
```

3. JS 中, 删除一个数组, 应该使用 `arr.length = 0`, 而不是 `delete a`

```
var arr = [1,2,3,4];
console.log(arr);
arr.length = 0;
console.log(arr);
```

4. JS 关闭当前页面

```
function close() {
  window.opener=null;
  window.open('','_self');
  window.close();
}
```

5. JS 中, 将数字字符串转换成数字, 使用 `parseInt` 时注意不要忘传第二个参数, 否则可能会出现一些问题

```
parseInt("42")      //=> 42
parseInt("042")     //=> 34 (Firefox) || 42 (Chrome)
parseInt("09")      //=> NaN (Firefox) || 9 (Chrome)
parseInt("the 42")  //=> NaN
parseInt("42px")    //=> 42 <- Advantage of using parseInt
parseInt("09", 10)  //=> 返回十进制的 9
```

当第二个参数的值为 0, 或没有设置该参数时, `parseInt()` 会根据 string 来判断数字的基数。

举例, 如果 string 以 "0x" 开头, `parseInt()` 会把 string 的其余部分解析为十六进制的整数。如果 string 以 0 开头, 那么 `ECMAScript v3` 允许 `parseInt()` 的一个实现把其后的字符解析为八进制或十六进制的数字。如果 string 以 1 ~ 9 的数字开头, `parseInt()` 将把它解析为十进制的整数。

6. `opacity < 1` 的元素会显示在最上层, 所以有的时候使用 `opacity: 99`; 可以方便的解决很多问题

```
.top {
  opacity: .99;
}
```

7. CSS 中 `position: absolute, relative;` 元素会晚于正常文档流元素的渲染, 所以显示的层级会比较高

```
a:hover {  
  position: relative;  
}
```

8. JS 中, 为 String 对象添加实用属性方法

```
// 判断字符串是否为空  
String.prototype.isEmpty = function() { return /\s*$/.test(this); }
```

```
// 删除字符串两边的空格  
String.prototype.trim = function() { return this.replace(/(^\\s*)|(\\s*$)/g, ""); }
```

```
// 删除字符串左边的空格  
String.prototype.ltrim = function() { return this.replace(/(^\\s*)/g, ""); }
```

```
// 删除字符串右边的空格  
String.prototype.rtrim = function() { return this.replace(/(\\s*$)/g, ""); }
```

9. JS 获取 URL 参数值

- 方式一 `substring`:

```
/**  
 * 获取 url 参数值: url?name1=value1&name2=value2  
 * @param {Object} name 参数名  
 * @return {TypeName} 参数值  
 * @author Tanken·L  
 */  
function getParam(name) {  
  // 若服务器端未设置 URL 编码为 UTF-8, 传递中文值时默认编码为 ISO-8859-1, 使用 decodeUR  
  方法处理中文乱码。  
  var url = decodeURI(location.href);  
  var paramString = url.substring(url.indexOf("?") + 1, url.length).split("&");  
  var paraObj = {};  
  for (i = 0; j = paramString[i]; i++) {  
    paraObj[j.substring(0, j.indexOf("=")).toLowerCase()] = j.substring(j.indexOf("=") + 1, j.leng  
h);  
  }  
  var returnValue = paraObj[name.toLowerCase()];  
  if (typeof (returnValue) == "undefined") {  
    return "";  
  } else {  
    return returnValue;  
  }  
}
```

- 方式二 `RegExp` (腾讯体验中心写法-正则表达式):

```
function getUrlParam(name) {  
  var reg = new RegExp("(^|&)" + name + "=(^[&]*)(&|$)"); // 构造一个含有目标参数的正则表达
```

对象

```
var r = window.location.search.substr(1).match(reg); // 匹配目标参数
if (r!=null) return unescape(r[2]); return null; // 返回参数值
}
```

10. JS 中使用倒序 for 循环实现高效遍历

```
var arr = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
// 高效倒序遍历，若要正序可先用 reverse 方法对数组进行反转再遍历
for (var i = arr.length; i--;) {
    // 依次打印 9 8 7 6 5 4 3 2 1 0
    console.log(arr[i])
}
```

11. JS 中使用 **charCodeAt** 和 **fromCharCode** 实现字符串的简单解密

```
// 编码
var enUnicode = function(s) {
    for (var arr = [], i = s.length; i--;) {
        arr.push(s.charCodeAt(i));
    }
    // 可将 D 改为其他字符
    return arr.join('D');
}

// 解码
var deUnicode = function(s) {
    for (var arr = [], code = s.split('D'), i = code.length; i--;) {
        arr.push(String.fromCharCode(code[i]));
    }
    return arr.join('');
}

// 使用示例
enUnicode('123123');
// > "51D50D49D51D50D49"

deUnicode("51D50D49D51D50D49");
// > "123123"
```

12. 自定义 WebKit 内核浏览器滚动条样式

• 样式一

```
::-webkit-scrollbar { width: 6px; height: 6px; }
::-webkit-scrollbar-track-piece { background-color: #eee; -webkit-border-radius: 3px; }
::-webkit-scrollbar-thumb { background-color: #999; outline: 0; border: 1px solid #999; -webkit-border-radius: 3px; }
::-webkit-scrollbar-thumb:hover { background-color: #666; }
```

• 样式二：

```
@media(-webkit-max-device-pixel-ratio:1) {  
  ::-webkit-scrollbar-track-piece { background-color:#fff }  
  ::-webkit-scrollbar { width:6px; height:6px }  
  ::-webkit-scrollbar-thumb { background-color:#c2c2c2; background-clip:padding-box; min-height:28px }  
  ::-webkit-scrollbar-thumb:hover { background-color:#a0a0a0 }  
}
```

- Google 登录页面滚动条样式:

```
::-webkit-scrollbar { width:8px }  
::-webkit-scrollbar-track { visibility:hidden }  
::-webkit-scrollbar-thumb { background:rgba(0, 0, 0, 0.12) }  
::-webkit-scrollbar-thumb:hover { background:rgba(0, 0, 0, 0.26) }
```

持续更新中...