

highlight.js:轻量级的Web代码语法高亮库

highlight.js是一款轻量级的Web代码语法高亮库，它主要有以下几个特点：

(1)、支持118种语言（看这里<https://github.com/isagalaev/highlight.js/tree/master/src/languages>）和54中样式（看这里<https://github.com/isagalaev/highlight.js/tree/master/src/styles>）；

(2)、可以自动检测编程语言；

(3)、同时为多种编程语言代码高亮；

(4)、支持在node.js上是哟个

(5)、支持任何标签

(6)、与任何 js 框架兼容。

在你的网站上面使用highlight.js也是很简单的，我们只需要引入样式文件和highlight.js即可。这里以monokai_sublime.min.css样式为例进行说明：

一、首先我们需要引入相关的文件，并调用initHighlightingOnLoad方法：

```
<link rel="stylesheet" href="/wp-content/monokai_sublime.min.css"/>
<script src="/wp-content/highlight.min.js"></script>
<script>hljs.initHighlightingOnLoad();</script>
```

```
hljs.initHighlightingOnLoad();
```

引入了上面的文件之后，highlight.js将在HTML中找到<pre><code>标签，并将它之间的代码高亮（上面的就是一个例子），并且可以自动推断里面的语言（下面的就是让highlight.js自动推断的）。不过如果自动推断无效，你可以加上class属性，并表明其中的语言：

```
<pre><code class="html">...</code></pre>
```

如果你不想让<pre><code>标签之间的内容语法高亮，可以加上nohighlight属性：

```
<pre><code class="nohighlight">...</code></pre>
```

下面再举一个对Scala代码高亮的例子：

```
package spark

import org.apache.spark.{SparkContext, SparkConf}
import org.apache.spark.SparkContext._
```

```
/**
 * Created by www.iteblog.com on 2015/04/17.
 */
object Test {
  def main(args: Array[String]) {
    val conf = new SparkConf().setAppName("cpc").setMaster("local")
    val sc = new SparkContext(conf)
    val ip = sc.textFile("file:///E:/1.txt")
    val map = ip.map(line => {
      val arr = line.split("\\t")
      println(arr(1) + "\\t " + arr(0))
      (arr(1), arr(0))
    }).groupByKey().collectAsMap()

    println(map)
  }
}
```

Java的语法高亮：

```
/**
 * Created by www.iteblog.com on 2015/04/17.
 */
package l2f.gameserver.model;

import java.util.ArrayList;

public abstract class L2Character extends L2Object {
  public static final Short ABNORMAL_EFFECT_BLEEDING = 0x0_0_0_1; // not sure

  public void moveTo(int x, int y, int z) {
    _ai = null;
    _log.warning("Should not be called");
    if (1 > 5) {
      return;
    }
  }
}

/** Task of AI notification */
@SuppressWarnings( { "nls", "unqualified-field-access", "boxing" })
public class NotifyAITask implements Runnable {
  private final CtrlEvent _evt;
```

```
List mList = new ArrayList()

public void run() {
    try {
        getAI().notifyEvent(_evt, _evt.class, null);
    } catch (Throwable t) {
        t.printStackTrace();
    }
}
}
```

关于更多highlightjs的使用，可以参见highlightjs的文档：<http://highlightjs.readthedocs.org/en/latest/api.html#configure-options>。

highlightjs的例子可以参见<https://highlightjs.org/static/demo>

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: **【】** ()