

思远(BonaShen) 博客

朋友们，我有了新家，就在<http://www.bonashen.com>

BEANSHELL 简介

简介：
BeanShell是一种脚本语言，一种完全符合java语法的java脚本语言，并且又拥有自己的一些语法和方法，beanShell是一种松散类型的脚本语言（这点和JS类似）。
下载地址：<http://www.beanshell.org>

设置环境
I 把;bsh-xx.jar放到\$JAVA_HOME/jre/lib/ext文件夹下
I unix: export CLASSPATH=\$CLASSPATH:bsh-xx.jar
I windows: set classpath %classpath%;bsh-xx.jar

运行方式：
I 界面UI方式：java bsh.Console
I 命令行方式：java bsh.Interpreter
I 运行脚本文件：java bsh.Interpreter filename [args]

简单举例：
在classpath中设置好环境变量，打开dos窗口，键入：java bsh.Console命令
出现BeanShell图片代表设置成功，beanshell开始运行

测试内容：
设置变量
foo = "Foo";
four = (2 + 2)*2/2;
打印变量
print(foo + " = " + four);
循环
for (i=0; i<5; i++)
print(i);
在窗口中打印按钮
button = new JButton("My Button");
frame = new JFrame("My Frame");
frame.getContentPane().add(button, "Center");
frame.pack();
frame.setVisible(true);

完整代码：
foo = "Foo";
four = (2 + 2)*2/2;
print(foo + " = " + four);

for (i=0; i<5; i++)
print(i);

MY LINKS

[首页](#)
[联系](#)
[订阅](#) [XML](#)
[管理](#)

BLOG STATS

Posts - 111
Stories - 1
Comments - 492
Trackbacks - 1

NEWS



BLOG中信息，目的是用于学术交流，而非商业运作，如有版权问题请及时和我联系！欢迎大家来交流。如要转载请注明并联系我！

昵称：[Bona shen](#)
园龄：[5年7个月](#)
粉丝：[2](#)
关注：[0](#)

搜索

常用链接

[我的随笔](#)
[我的空间](#)
[我的短信](#)
[我的评论](#)
[更多链接](#)

我的标签

[shipbuilding](#)(14)
[IT](#)(8)
[Design](#)(3)

```
button = new JButton( "My Button" );
frame = new JFrame( "My Frame" );
frame.getContentPane().add( button, "Center" );
frame.pack();
frame.setVisible(true);
```

在窗口中输入上面的代码

敲回车执行，运行结果如图

```
说明：
因为beanshell是松散类型的脚本语言因此可以直接写
foo = "Foo";
four = (2 + 2)*2/2;
```

print是beanshell提供一种简单的打印命令相当于java中的System.out.println()

其他的beanshell脚本命令

- source(), run() – 读取，或者运行一个脚本文件
- frame() – 显示一个窗口
- load(), save() – 读取或者保存一个脚本对象到文件
- cd(), cat(), dir(), pwd(), etc. 使用Unix下面的命令
- exec() – 运行一个本地的方法
- javap() –使用javap命令.
- setAccessibility() – 设置可以接收private和protected类型的变量

BeanShell命令不一定是内置的脚本命令，脚本方法会自动从classpath中取方法使用，因此你可以添加你自己的脚本到classpath中来扩充基本的命令

脚本方法

一般的方法：

```
int addTwoNumbers( int a, int b ) {
return a + b;
}
```

```
sum = addTwoNumbers( 5, 7 ); // 12
也可以使用动态的变量类型（无状态）方法
add( a, b ) {
return a + b;
}
```

```
foo = add(1, 2); // 3
foo = add(1, "2"); //"12" 只要有一个为字符串全部按照字符串处理，系统不会根据1是数字在前把“2”转换成数字处理（特别注意）
foo = add("Oh", " baby"); // "Oh baby"
```

- [Gantt](#)(3)
- [licence](#)(2)
- [ERP](#)(2)
- [管理](#)(2)
- [精细化](#)(2)
- [SC](#)(2)
- [PWBS](#)(1)
- [更多](#)

收藏夹(8)

- [.Net](#)(2) [\(rss\)](#)
- [Ajax](#) [\(rss\)](#)
- [Cracks](#)(3) [\(rss\)](#)
- [ERP](#) [\(rss\)](#)
- [framework](#) [\(rss\)](#)
- [JAVA](#)(2) [\(rss\)](#)
- [MDA](#) [\(rss\)](#)
- [webservice](#) [\(rss\)](#)
- [workflow](#) [\(rss\)](#)
- [哲理](#)(1) [\(rss\)](#)

CRACKS

- [OllyDBG Cracking Tutorials](#)
- [Reverse Engineering Team](#)
- [看雪技术论坛](#)
- [邪恶八进制信息安全团队](#)
- [一蓑烟雨 - Powered by Unpack !](#)

DATA BANK

- [CNKI](#)
- [中国期刊网](#)
- [盟动力个人主页](#)

DEVELOPMENT WEB

- [AJAX中国](#)
- <http://www.javaresearch.org/>
- [JavaWorld@TW](#)
- [JAVA视线论坛](#)
- [Onlysail](#)
- [Matrix\(与JAVA共舞\)](#)
- [W3China](#)
- [中国XML论坛](#)

DOWNLOAD RESOURCE

- [全方位下载](#)

MANAGEMENT WEB

- [宁波项目管理网](#)

SAP

- [macroerp](#)
- [sap online help](#)
- [sap-directory](#)
- [SAP-IMG](#)
- [SAP Basis, ABAP](#)

实现接口

实现任何接口需要java1.3或者更高

可以使用缺省的java匿名类的语法实现一个接口类，例如：

```
ActionListener scriptedListener = new ActionListener() {  
    actionPerformed( event ) { ... }  
}
```

不需要实现接口的所有的方法，只需要实现你使用的方法即可，如果使用你没有实现的方法，beanshell将抛出一个错误，

```
ml = new MouseListener() {  
    mousePressed( event ) { ... }  
    // handle the rest  
    invoke( name, args ) { print("Method: "+name+" invoked!");  
    }
```

脚本对象

使用特殊的关键字this可以创建一个对象（根JS类似）

```
foo() {  
    print("foo");  
    x=5;
```

```
bar() {  
    print("bar");  
}
```

```
return this;  
}
```

```
myfoo = foo(); // prints "foo"  
print( myfoo.x ); // prints "5"  
myfoo.bar(); // prints "bar"
```

从应用程序中调用BeanShell

创建一个BeanShell的解释器(interpreter)用eval（）和source()命令可以对一个字符串求值和运行一个脚本文件

使用set（）方法可以给一个对象传入一个变量的参考

使用get（）方法可以重新得到一个变量的结果

完整代码：

```
package cn.com.sparknet.util;
```

```
import bsh.*;  
import java.util.*;
```

Programming and Other IMG Stuff

[SAP-Mit](#)

[FI，CO一步一步学](#)

[SAP俱乐部](#)

[SAP网站排名](#)

STUDY

[PPT模板库](#)

[XML-RPC](#)

哲理

[<<道德经>>句解](#)

[南怀瑾老师著作](#)

积分与排名

积分 - 155087

排名 - 582

```
public class BeanShell {
public static void main(String[] args) {
try {
Interpreter interpreter = new Interpreter(); // 构造一个解释器
interpreter.set("foo", 5); // 设置变量
interpreter.set("date", new Date()); //设置一个时间对象
Date date = (Date) interpreter.get("date"); // 重新得到时间变量
interpreter.println(date.toString()); //打印时间变量
interpreter.eval("bar = foo*10"); // 对一段脚本求值，并得到结果
System.out.println(interpreter.get("bar")); //打印变量
interpreter.source("d:\\helloWorld.bsh"); // 导入并执行一个脚本文件
}
catch (Exception e) {
//如果发生异常，写入日志文件
Log.error(new BeanShell(), "main", FormatDate.getCurrDate(), e.getMessage());
}
}
}
```

BeanShell语法

BeanShell是一种最原始的java解释器。

标准的java语法

/*

Standard Java syntax

*/

// Use a hashtable

Hashtable hashtable = new Hashtable();

Date date = new Date();

hashtable.put("today", date);

// Print the current clock value

print(System.currentTimeMillis());

// Loop

for (int i=0; i<5; i++)

print(i);

// Pop up a frame with a button in it

JButton button = new JButton("My Button");

JFrame frame = new JFrame("My Frame");

frame.getContentPane().add(button, "Center");

frame.pack();

frame.setVisible(true);

松散类型的java语法

/*

Loosely Typed Java syntax

*/

// Use a hashtable

```
hashtable = new Hashtable();
date = new Date();
hashtable.put( "today", date );

// Print the current clock value
print( System.currentTimeMillis() );

// Loop
for (i=0; i<5; i++)
print(i);

// Pop up a frame with a button in it
button = new JButton( "My Button" );
frame = new JFrame( "My Frame" );
frame.getContentPane().add( button, "Center" );
frame.pack();
frame.setVisible(true);
异常处理
标准的Java异常
try {
int i = 1/0;
} catch ( ArithmeticException e ) {
print( e );
}
松散的异常处理（类似JS）
try {
...
} catch ( e ) {
...
}
松散类型变量的作用范围
标准的Java程序的变量作用范围是在一个模块中的（在模块中声明的变量），而在松散类型的语言中如果在一个模块中没有指定一个变量的类型，则认为是一个 全局变量（只有它以后的代码可以使用该变量，系统在调用该变量的时候自动生成一个全局变量，也就为什么在调用模块之前不能使用该变量的原因）
// Arbitrary code block
{
y = 2; // Untyped variable assigned
int x = 1; // Typed variable assigned
}
print( y ); // 2
print( x ); // Error! x is undefined.

// Same with any block statement: if, while, try/catch, etc.
if ( true ) {
y = 2; // Untyped variable assigned
int x = 1; // Typed variable assigned
}
print( y ); // 2
print( x ); // Error! x is undefined.

同样也使用于for-loop, if-else等循环语句
for( int i=0; i<10; i++ ) { // typed for-init variable
j=42;
}
print( i ); // Error! 'i' is undefined.
print( j ); // 42
```

```
for( z=0; z<10; z++ ) { } // untyped for-init variable
print( z ); // 10
```

方便灵活的语法

标准的java语法

```
java.awt.Button button = new java.awt.Button();
button.setLabel("javaButton");
```

松散的语法

```
button = new java.awt.Button();
button.label = "my button";
```

你也可以使用{ } 来对一个对象设置属性

```
b = new java.awt.Button();
b{"label"} = "my button"; // Equivalent to: b.setLabel("my button");
```

```
h = new Hashtable();
h{"foo"} = "bar"; // Equivalent to: h.put("foo", "bar");
```

自动装箱和自动拆箱 (box和unbox)

BeanShell自动转为简单类型

```
i=5;
iw=new Integer(5);
print( i * iw ); // 25
```

导入类和包

```
import javax.xml.parsers.*;
import mypackage.MyClass;
```

超级导入法:

```
import *;
```

BeanShell默认导入下面的包

- java.lang
- java.io
- java.util
- java.net
- java.awt
- java.awt.event
- javax.swing
- javax.swing.event

友好文档实体

BeanShell支持特殊的文档操作类型内容

```
@gt > @lt <
@lteq <= @gteq >=
@or || @and &&
@bitwise_and & @bitwise_or |
@left_shift << @right_shift >>
```

```
@right_unsigned_shift >>> @and_assign &=
@or_assign |= @left_shift_assign <<=
@right_shift_assign >>= @right_unsigned_shift_assign >>>=
```

脚本方法

你可以定义方法象java中的定义方法一样

```
int addTwoNumbers( int a, int b ) {
return a + b;
}
```

你可以使用内嵌的BeanShell方法使用他们

```
sum = addTwoNumbers( 5, 7 );
```

只有BeanShell变量可以被动态定义为动态类型，方法可以有动态的参数以及返回类型

```
add( a, b ) {
return a + b;
}
```

在这个方法中，BeanShell将动态的决定类型当这个方法被调用时并且能够准确的计算出你想要的结果

```
foo = add(1, 2);
print( foo ); // 3
```



```
foo = add("Oh", " baby");
print( foo ); // Oh baby
```

在第一个例子中BeanShell将把参数定义为数字型，并返回数字型

在第二个例子中BeanShell将把参数定义为字符型，并返回字符对象

变量和方法的可见范围

就像您所预期的那样，在方法内您可以参考到上下文中上面的变量和方法

```
a = 42;
someMethod() { ... }
```



```
foo() {
print( a );
someMethod(); // invoke someMethod()
}
```



```
// invoke foo()
foo(); // prints 42
```

如果一个变量只有在方法内使用请定义成局部变量，即加上类型，如果是全局变量请在方法外定义

```
var = "global";
foo() {
print(var);
String var = "local";
print(var);
}
foo();
print(var);
```

将打印出

```
global
local
global
```

方法内的var（第四行）变量属于局部变量，不会覆盖全局变量var（第一行）的因此改变var（第四行）变量不会影响到全局变量var（第一行）

范围参考：super

使用super关键字可以在局部参考到全局变量

```
var = "global";
foo() {
String var = "local";
print(var);
```

```
print(super.var);
}
foo();
将输出
local
global
```

脚本对象

this对象

在java标准语言中可以使用**this**返回一个类的一个实例

```
// MyClass.java
MyClass {
Object getObject() {
return this; // return a reference to our object
}
}
```

在这个例子中**getObject()** 方法是返回**MyClass**类的一个实例

在BeanShell中对象中的变量只是局部的变量在对象内可以使用，在对象外是不可以使用（不同于前面**for-loop**,**if-else**中的使用）；

// Define the **foo()** method:

```
foo() {
bar = 42;
print( bar );
}
```

// Invoke the **foo()** method:

```
foo(); // prints 42
```

```
print( bar ); // Error, bar is undefined here
```

可以使用**this**返回对象，使用对象加上**"."**运算符参考属性（类似JS）

```
foo() {
bar = 42;
return this;
}
```

```
fooObj = foo();
```

```
print( fooObj.bar ); // prints 42!
```

同样对象中也可以定义一些方法

```
foo() {
bar() {
...
}
}
```

例如

```
foo() {
int a = 42;
bar() {
print("The bar is open!");
}
```



```
bar();
return this;
}

// Construct the foo object
fooObj = foo(); // prints "the bar is open!"
// Print a variable of the foo object
print ( fooObj.a ) // 42
// Invoke a method on the foo object
fooObj.bar(); // prints "the bar is open!"
也可以把bar () 和foo也可以代参数
foo() {
return this;
}
bar(int a) {
print("The bar is open!" + a);
}
foo().bar(1);
也可以把bar () 方法定义到对象外面

foo() {
bar(int a) {
print("The bar is open!" + a);
}
return this;
}
foo().bar(1);
BeanShell一种松散的脚本语言，有很多中声明的方法可以使用
This super global
This是参考当前对象
Super是参考父亲对象
Global是参考最上层对象
super.super.super...foo = 42; // Chain super. to reach the top
global.foo = 42;
```

简单例子：
文本拖动：

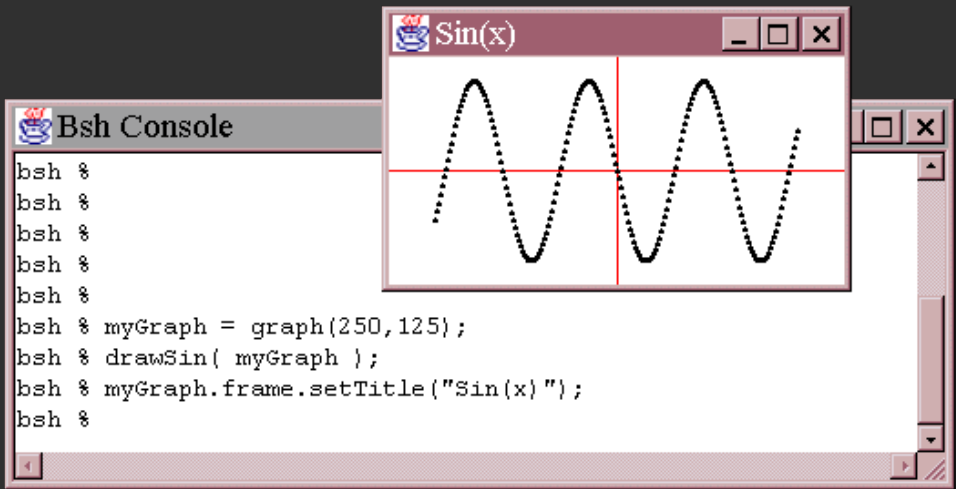
```
dragText() {
f = new Frame("Drag in the box");
f.setFont( new Font("Serif", Font.BOLD, 24) );
f.setSize(300, 300);
f.show();
gc = f.getGraphics();
gc.setColor(Color.cyan);
mouseDragged( e ) {
gc.drawString("Drag Me!", e.getX(), e.getY());
}
```

```
mouseMoved( e ) { }  
f.addMouseMotionListener( this );  
}
```

简单画图

```
import bsh.util.BshCanvas; // BshCanvas simply buffers graphics
```

```
graph( int width, int height ) {  
    canvas=new BshCanvas();  
    canvas.setSize( width, height );  
    frame=frame( canvas );  
    graphics=canvas.getBufferedGraphics();  
    // draw axis  
    graphics.setColor( Color.red );  
    graphics.drawLine( 0, height/2, width, height/2 );  
    graphics.drawLine( width/2, 0, width/2, height );  
    graphics.setColor( Color.black );  
  
    plot(int x, int y) {  
        graphics.fillOval( (x+width/2-1), (y+height/2-1), 3, 3);  
        canvas.repaint();  
    }  
  
    return this;  
}  
  
drawSin( graph ) {  
    for (int x=-100; x<100; x++) {  
        y=(int)(50*Math.sin( x/10.0 ));  
        graph.plot( x, y );  
    }  
}
```



简单web浏览器

```
import javax.swing.*;  
import javax.swing.event.*;  
import javax.swing.text.*;  
import java.awt.event.*;  
import java.awt.*;
```

```
JFrame browser( startingUrl ) {
    invoke( method, args ) {}

    windowClosing(WindowEvent we) {
        we.getWindow().setVisible(false);
    }

    setPage( url ) {
        try {
            pane.setPage( url );
        } catch(Exception e) {
            statusBar.setText("Error opening page: "+url);
        }
    }

    hyperlinkUpdate( HyperlinkEvent he ) {
        type = he.getEventType();
        if (type == HyperlinkEvent.EventType.ENTERED) {
            pane.setCursor(
                Cursor.getPredefinedCursor( Cursor.HAND_CURSOR ) );
            statusBar.setText(he.getURL().toString());
        } else
        if (type == HyperlinkEvent.EventType.EXITED) {
            pane.setCursor( Cursor.getDefaultCursor() );
            statusBar.setText(" ");
        } else {
            setPage( he.getURL() );
            if (urlField != null)
                urlField.setText(he.getURL().toString());
        }
    }

    frame = new JFrame("Browser");
    frame.setSize(400,300);
    frame.addWindowListener( this );

    urlPanel = new JPanel();
    urlPanel.setLayout(new BorderLayout());
    urlField = new JTextField(startingUrl);
    urlPanel.add(new JLabel("Site: "), BorderLayout.WEST);
    urlPanel.add(urlField, BorderLayout.CENTER);

    statusBar = new JLabel(" ");
    pane = new JEditorPane();
    pane.setEditable(false);
    setPage( startingUrl );
    jsp = new JScrollPane(pane);

    frame.getContentPane().add(jsp, BorderLayout.CENTER);
    frame.getContentPane().add(urlPanel, BorderLayout.SOUTH);
    frame.getContentPane().add(statusBar, BorderLayout.NORTH);

    // This is the equivalent of an inner class in bsh.
    urlTextHandler() {
        actionPerformed(ActionEvent ae) {
            setPage( ae.getActionCommand() );
        }
    }
}
```

```
}
return this;
}

urlField.addActionListener( urlTextHandler() );
pane.addHyperlinkListener( (HyperlinkListener)this );

return frame;
}
browser = browser("http://java.sun.com/");
browser.show();
```

writed by bonashen

posted on 2006-10-23 11:31 Bona shen 阅读(2054) 评论(0) 编辑 收藏

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[返回博客园首页](#)。

[简洁阅读版式](#)

[不3k就业不给1分学费\(java, .net, php, android\)](#)

最新IT新闻:

- [诺基亚中国裁员追踪：被裁员工对新方案仍不满](#)
- [消息称淘宝商城下周将推出开放平台 针对独立B2C](#)
- [盛大创新院发布光速搜索工具 将不涉足全网搜索](#)
- [中电信部分省市已接到iPhone 5销售准备通知](#)
- [Windows To Go：在 USB 存储设备中运行 Windows 8，附演示视频](#)

» [更多新闻...](#)

最新知识库文章:

- [软件开发应该以史为鉴，还是从头再来？](#)
- [Web应用程序的开发步骤](#)
- [利用Httponly提升web应用程序安全性](#)
- [理解RESTful架构](#)
- [我不是一个工程师 - 我是一个软件开发者](#)

» [更多知识库文章...](#)



找个老外学英语 迅速提高英语口语 口语就要这样练 www.MarsEnglish.com.cn	上海招聘求职 到百姓网 招聘求职信息真实有效,上万岗位 免费查询,企业直招,快速上岗,无中介费 www.Baixing.com
---	--

Google 提供的广告



[China-pub](#) 2011秋季教材巡展

[China-Pub](#) 计算机绝版图书按需印刷服务

[网站首页](#) [博问](#) [知识库](#) [IT新闻](#) [我的园子](#) [闪存](#) [程序员招聘](#)

Powered by:

[博客园](#)

Copyright © Bona shen