

Basic Examples of web $Mathematica$ for Teachers

Any calculation done in $Mathematica$ can be done using $webMathematica$, with two noteworthy limitations. First, when using $webMathematica$, you will not have access to the full capabilities of $Mathematica$'s front end. Second, license restrictions may prohibit you from enabling certain calculations when you use $webMathematica$.

In this section we will show several simple examples of $webMathematica$ jsp pages. Many of these can be copied and used as the basic for your own work. You can test their functionality on appropriate web pages.

The description given here will work through a collection of sample JSPs, each of which will demonstrate some details of feature. The sources for all these examples are included in this chapter and fully working JSPs pages reader can find on EVLM web. When you finished this chapter wish to look at the next chapter: Developing your own Pages. This gives you some ideas for getting started to develop your own site.

Example 1 – Page with Text and Statically Given Evaluation

This example will create a simple html page with static text and statically defined evaluation. On this page we will evaluate the integral of some function and solve a linear equation. First we will show calculations and texts done in $Mathematica$

Integrate using the standard $Mathematica$ command **Integrate**

```
In[1]:= Integrate[x^2 + 3, x]
```

```
Out[1]= 3 x +  $\frac{x^3}{3}$ 
```

Solve the linear equation $2x+3=-5$

```
In[2]:= Solve[2 x + 3 == -5, x]
```

```
Out[2]= {{x -> -4}}
```

The same calculation can be provided by the next jsp page. This page uses standard HTML tags as well as special $webMathematica$ tags. These have the form `<msp:tag>`. The $webMathematica$ tags are executed from the top to the bottom. The tag `<msp:allocate>` allocates the $Mathematica$ kernel to use for computations. The contents of the `<msp:evaluate>` tags are sent to $Mathematica$ for computation. The result will be inserted into the final page and displayed. The tag `</msp:allocate>` frees the $Mathematica$ kernel to be used for another computation.

The source code:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="HTMLFiles/pmathml.xsl"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.1 plus MathML 2.0//EN"
"HTMLFiles/xhtml1-math11-f.dtd">

<%@ page language="java" %>
<%@ taglib uri="/webMathematica-taglib"
prefix="msp" %>

<html>
<head>
<title> Example1 </title>
</head>

<body>
<h1> Example1 </h1>
<msp:allocateKernel>

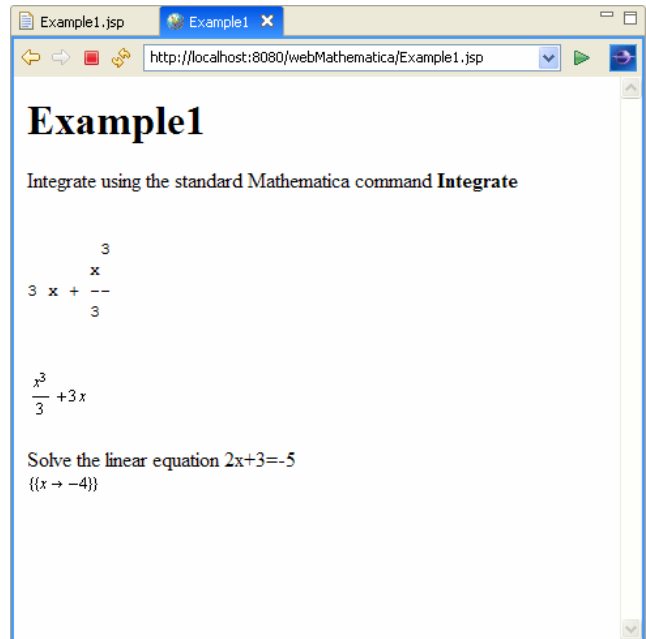
<p> Integrate using the standard
Mathematica command <b>Integrate</b> </p>

    <msp:evaluate>
        Integrate[x^2+3,x]
    </msp:evaluate>
<br/>
    <msp:evaluate>
        MSPFormat[Integrate[x^2+3,x], TraditionalForm]
    </msp:evaluate>
<br/>

<p> Solve the linear equation 2x+3=-5 <br/>
    <msp:evaluate>
        MSPFormat[Solve[2x+3==-5,x], TraditionalForm]
    </msp:evaluate>

</p>

</msp:allocateKernel>
</body>
</html>
```



You can check this example on

<http://webmathematica.mathematica.sk/webMathematica/evIm/book/Example1.jsp>

Example 2 – Page with Simple Evaluation

This example evaluates the `Date[]` function of Mathematica and **Expand** the random generated polynomial. The result changes each time the page is accessed, evaluation this really is dynamic process. First we will show calculations done in *Mathematica*.

```
In[3]:= Date[]
```

```
Out[3]= {2007, 11, 17, 0, 6, 44.8281250}
```

```
In[4]:= Expand[(x+y)^Random[Integer, {0, 10}]]
```

```
Out[4]= x3 + 3 x2 y + 3 x y2 + y3
```

The same calculation can be provided by next jsp page. The HTML tags **<form>** and **<input>** provide to sent user request to the server. Each time the page is accessed, the new evaluation is provide by *Mathematica* kernel. User can done this by click on the submit button.

The source code:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="HTMLFiles/pmathml.xsl"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.1 plus MathML 2.0//EN"
"HTMLFiles/xhtml1-math11-f.dtd">

<%@ page language="java" %>
<%@ taglib uri="/webMathematica-taglib"
prefix="msp" %>

<html>
<head>
<title> Example2 </title>
</head>

<body>
<h1> Example2 </h1>

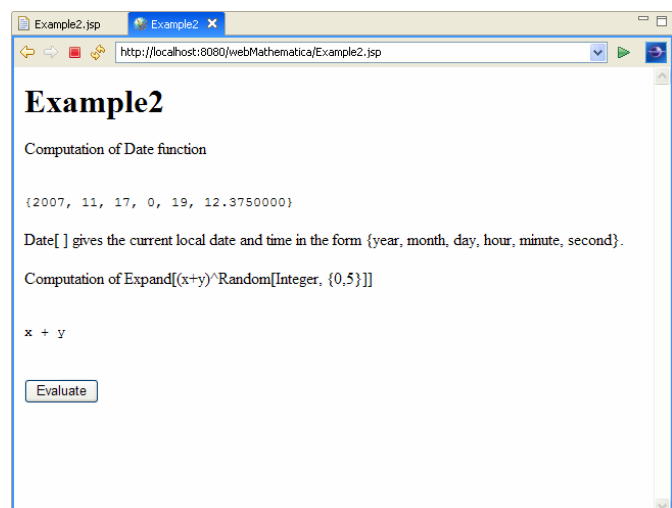
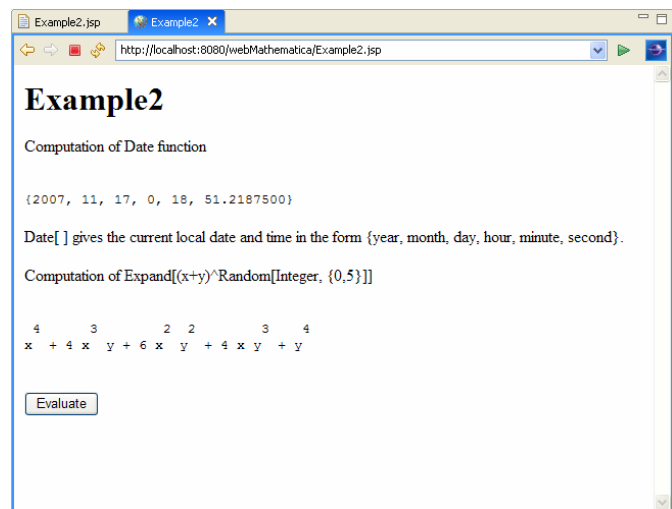
<form action="Example2.jsp"
method="post">

<msp:allocateKernel>

<p>
Computation of Date function
</p>
<msp:evaluate>
Date[ ]
</msp:evaluate>

<msp:evaluate>
Date::usage
</msp:evaluate>
<p>
Computation of
Expand[(x+y)^Random[Integer, {0,5}]]
</p>
<msp:evaluate>
Expand[(x+y)^Random[Integer, {0,5}]]
</msp:evaluate>

</msp:allocateKernel>
```



```

<br/>
<input type="submit" name="button1" value="Evaluate">
</form>
</body>
</html>

```

You can check this example on

<http://webmathematica.mathematica.sk/webMathematica/evlm/book/Example2.jsp>

Example 3 – Page with Various Graphical Outputs

This example generates a various plots. We will show the use of the MSP function **MSPShow**. **MSPShow** takes the Mathematica graphics object from the **Plot** (**ListPlot**, **Plot3D**...) command and generates a GIF image, which is stored on the server, returning an HTML **** tag.

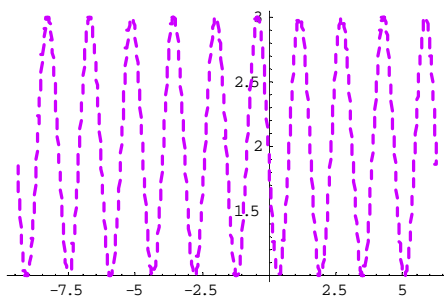
There are also several problems and questions with sending image to the browser, but in this simple example we will not discuss. on it. *Mathematica* can output graphics in a number of formats including GIF, animated GIF, and JPEG, SVG. Using the Java applet LiveGraphics3D, you can also create and manipulate three-dimensional graphics interactively in your web browser.

First we will show calculations done in *Mathematica*.

```

In[1]:= Plot[Sin[4 x - 3] + 2, {x, -3 Pi, 2 Pi},
PlotStyle -> {Dashing[{0.02}], Thickness[0.01],
Hue[0.8]}]

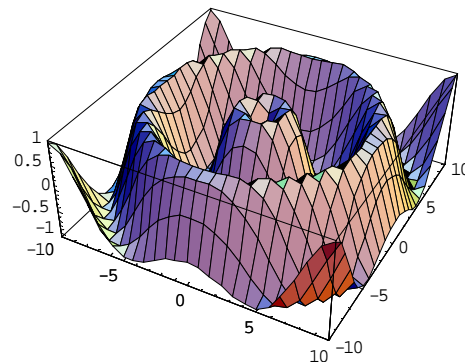
```



```

In[2]:= Plot3D[Sin[Sqrt[x^2 + y^2]], {x, -10, 10},
{y, -10, 10}]

```



The same calculation can be provided by next jsp page.

The source code:

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="HTMLFiles/pmathml.xsl"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN"
"HTMLFiles/xhtml1-math11-f.dtd">

<%@ page language="java" %>
<%@ taglib uri="/webMathematica-taglib" prefix="msp" %>

```

```
<html>
<head>
<title> Example3 </title>
</head>

<body>
<h1> Example3 </h1>

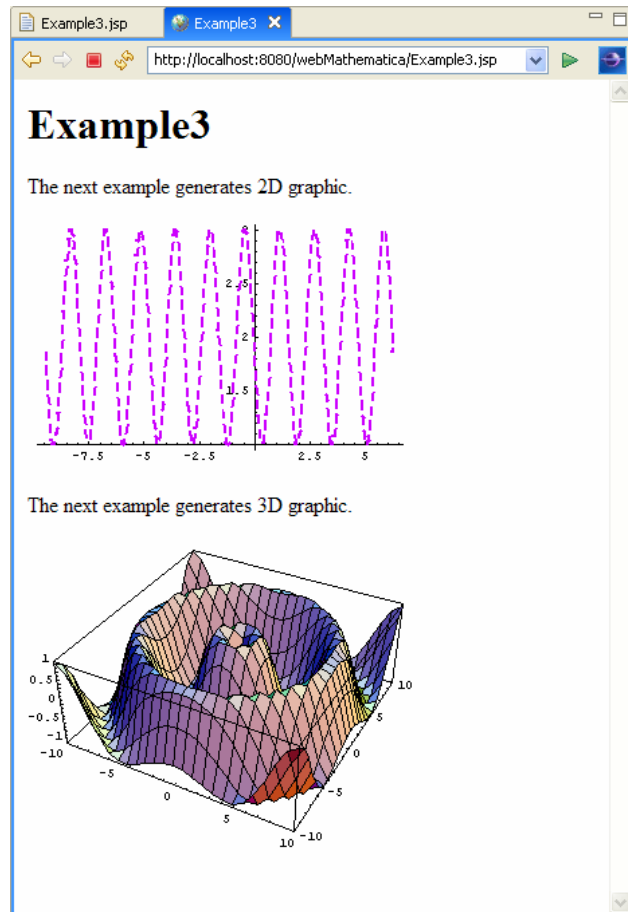
<msp:allocateKernel>

<p> The next example generates 2D graphic.</p>
<msp:evaluate>
  MSPShow[Plot[Sin[4x - 3] + 2, {x, -3Pi, 2Pi},
    PlotStyle -> {Dashing[{0.02}], Thickness[0.01], Hue[0.8]}]]
</msp:evaluate>

<p> The next example generates 3D graphic.</p>
<msp:evaluate>
  MSPShow[Plot3D[Sin[Sqrt[x^2 + y^2]], {x, -10, 10}, {y, -10, 10}]
</msp:evaluate>

</msp:allocateKernel>

</body>
</html>
```



You can check this example on

<http://webmathematica.mathematica.sk/webMathematica/evlm/book/Example3.jsp>

Example 4 – Simple Interactive Communication

This example allows to create a 2D plot of the function specified by user. This page is more elaborate because it contains **<form>** and **<input>** HTML elements. These are important ways for allowing interaction from client.

A **<form>** element is a block of HTML that contains input elements. A form may be activated with an **<input>** of type submit, this send the name and values associated with each input tag to the server.

Here, the opening tag of the form element contains two attributes. The action refers to a URL that is accessed when the form is activated. In this case, it is a relative URL that refers to the original script. The method attribute tells the browser what HTTP method to use, in this case , a post method.

This example has four input tags, the first three tags allow the user of the page enter text, and the fourth specifies a button that, when pressed, will submit the form. When form is submitted, it will send the information from input elements to the URL specified by the action attribute (in this case the same jsp file).

Text entered into the input tag, which uses the name **func**, will be assigned to the input variable **\$\$func**. Text entered into the input tag, which uses the name **min**, will be assigned to the input variable **\$\$min** etc.

The first time the page is accessed there is no value for **\$\$func**. When a value is entered in the text field and the **Evaluate** button is pressed, **\$\$func** gets a value which will be send to the server. The same process will be perform for the next two text input tag. If you want the input to be interpreted and evaluated by *Mathematica*, you need to use one of the MSP functions. **MSPValue** assign *Mathematica* interpreted value to the variable.

Note : **name** is the name of the html variable

\$\$name is the name of the *Mathematica* variable

The **MSPBlock** command is a programming construct, which here inspects to the three variables, **\$\$func**, **\$\$min**, **\$\$max**. If either of these has no value, **MSPBlock** returns a null string, which is why the first time you acces the page, you do not see a result. The values of all three variables are then interpreted by *Mathematica*. If successful, the result of interpretation are substituted into second argument or body of **MSPBlock**. In this example all instances of **func**, **min** and **max** are substituted with the parsed value of **\$\$func**, **\$\$min** and **\$\$max**. The result is then evaluated, formatted, and places in the html page, which is returned to the client.

The next example generates a plot of some function **func** (given by user) in range given by user on the text fields **min** and **max**. For the first time of calculation the default values were assigned to the used variables.

The source code:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="HTMLFiles/pmathml.xsl"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN"
      "HTMLFiles/xhtml1-math11-f.dtd">

<%@ page language="java" %>
<%@ taglib uri="/webMathematica-taglib" prefix="msp" %>

<html>
<head>
<title> Example4 </title>
</head>

<body>
<h1> Example4 </h1>

<msp:allocateKernel>

<p> The next example generates 2D graphic.</p>

<form action="Example4.jsp" method="post">

Enter a function <i>f(x)</i> :
<input type="text" name="func" size="20"
      value="<msp:evaluate>MSPValue[ $\text{Sin}[x]^2$ ]" /><br/>
Enter a min value for plot <i>min</i> :
<input type="text" name="min" size="20"
      value="<msp:evaluate>MSPValue[ $0$ ]" /><br/>
Enter a max value for plot <i>max</i> :
<input type="text" name="max" size="20"
      value="<msp:evaluate>MSPValue[ $10$ ]" /><br/>

<br/>
<input type="submit" name="button1" value="Evaluate">

<msp:evaluate>
MSPBlock[ $\{\text{func}, \text{min}, \text{max}\}$ ,
      MSPShow[Plot[ $\{\text{func}, \{x, \text{min}, \text{max}\}$ ,
        PlotStyle -> {Dashing[{0.02}], Thickness[0.01], Hue[0.6]}]}
      ]
</msp:evaluate>

</msp:allocateKernel>
</form>
</body>
</html>
```

You can check this example on

<http://webmathematica.mathematica.sk/webMathematica/evlm/book/Example4.jsp>

Example4

The next example generates 2D graphic.

Enter a function $f(x)$:

Enter a min value for plot min :

Enter a max value for plot max :

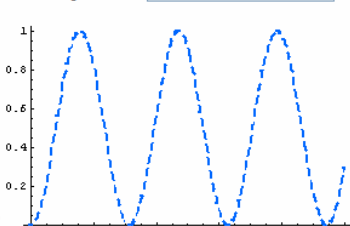
Example4

The next example generates 2D graphic.

Enter a function $f(x)$:

Enter a min value for plot min :

Enter a max value for plot max :



User can change the values of variables **func**, **min** and **max** and sent the new request for evaluation to the server. If successful, the result will appeared in the same window.

Example4

The next example generates 2D graphic.

Enter a function $f(x)$:

Enter a min value for plot min :

Enter a max value for plot max :

