

Основни примери с webMathematica за преподаватели

Всяко изчисление, направено с *Mathematica* може да се направи с *webMathematica* при две важни ограничения. Първо, когато се работи с *webMathematica*, нямате достъп до пълните възможности на потребителския интерфейс (front end) на *Mathematica*. Второ, лицензни ограничения могат да Ви забранят извършването на някои изчисления с *webMathematica*.

В тази глава ще дадем няколко прости примера за *webMathematica* jsp страници. Повечето от тях можете да копирате и да ползвате като база за Ваша собствена разработка. Можете да тествате тяхната функционалност в подходящи уеб-страници.

Описанията, които даваме тук, ще работят като набор от примерни JSP страници, всяка от които ще демонстрира някои детайли на отделна характеристика на *webMathematica*. Изходните кодове на всички дадени тук примери са напълно изпълними и читателят може да ги изпробва на сайта на EVLM. След прочитането на тази глава ще поискате да погледнете към следващата: Разработка на Ваши собствени страници. Ние Ви даваме някои идеи как да започнете да създавате свои собствени сайтове.

Пример 1 – Страница с текст и статично изчисление

Този пример създава проста html страница и статично дефинирано изчисление. В страницата ще изчислим интеграл от някаква функция и ще решим едно линейно уравнение. Първо да покажем изчисленията и текстовете както са направени с *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}}
```

Същото изчисление може да се осъществи със следващата jsp страница. Тази страница използва стандартни HTML тагове, както и специални тагове на *webMathematica*. Последните имат вида `<msp:tag>`. Таговете на *webMathematica* се изпълняват отгоре надолу. Тагът `<msp:allocate>` възлага изчисление на ядрото на *Mathematica*. Съдържанието на таговете `<msp:evaluate>` се изпраща към *Mathematica* за изчисляване. Резултатът ще се вмъкне в уеб-страницата и ще

се изобрази на екрана. Тагът `</msp:allocate>` освобождава ядрото на *Mathematica* за друго изчисление.

Исходен код:

```
<?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>
```

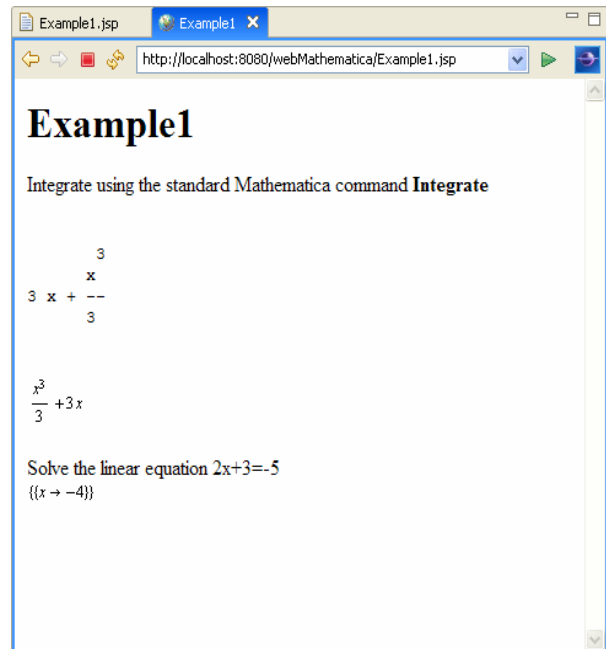
```
<p> Integrate using the standard Mathematica command <b>Integrate</b> </p>
```

```
    <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>
```



Можете да тествате този пример на адрес:

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

Пример 2 – Страница с просто онлайн изчисление

Този пример изчислява функцията на *Mathematica* `Date[]` и развива по степени произволно генериран полином с функцията `Expand[]`. Резултатът при всяко отваряне на веб-страницата е различен, т.е. изчислението е реален динамичен процес. Първо ще покажем изчисленията, както се правят с *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
```

Същото изчисление ще се извърши със следващата jsp страница. HTML таговете `<form>` и `<input>` служат за изпращане на потребителската заявка към сървъра. При всяко отваряне на страницата се предоставя ново изчисление от ядрото на *Mathematica*. Потребителят го получава с натискане на бутона Submit.

Изходен код:

```
<?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" %>

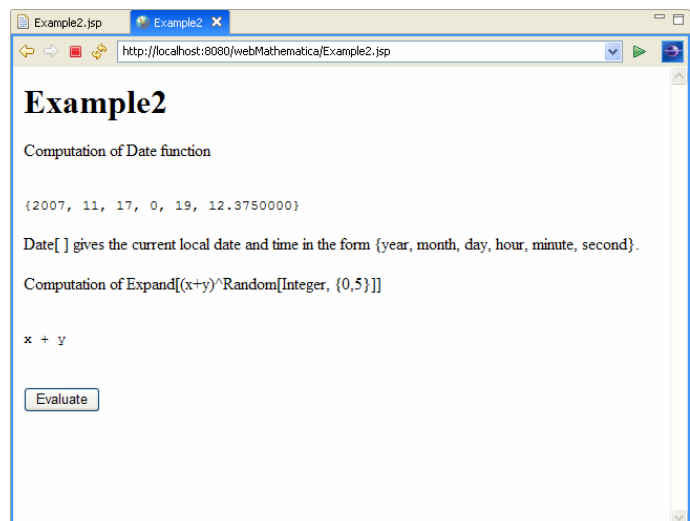
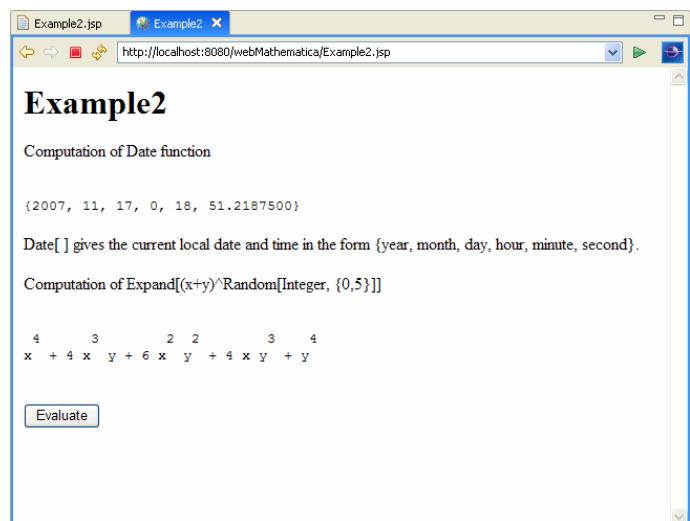
<html>
<head>
<title> Example2 </title>
</head>

<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>

```

Можете да тествате този пример на адрес
<http://webmathematica.mathematica.sk/webMathematica/evlm/book/Example2.jsp>

Пример 3 – Страница с променливи графични изходи

Този пример създава различни графики. Ще покажем как се използва MSP функцията `MSPShow`. `MSPShow`, която взима графичен обект от командата на *Mathematica* `Plot` (`ListPlot`, `Plot3D`...) и генерира GIF изображение, което се запомня в сървъра, като връща HTML тага ``.

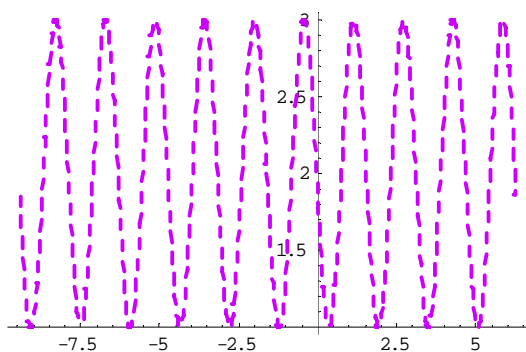
Освен тези, има и други възможни елементи при изпращането на изображение към сървъра, но тук няма да ги дискутираме. *Mathematica* може да извежда графики в най-различни формати, включително GIF, анимиран GIF, JPEG, SVG. С използване на Java аплета `LiveGraphics3D`, също можете да създавате и манипулирате интерактивно тримерни графики на Вашия браузър.

Първо ще покажем командите за изчисленията с *Mathematica*.

```

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

```

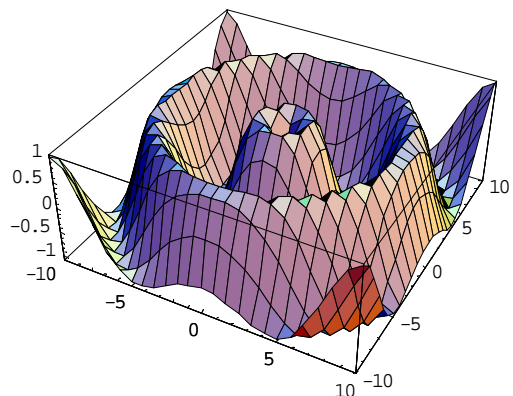


Out[1]= - Graphics -

```

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

```



Out[2]= - SurfaceGraphics -

Същото изчисление може да се получи със следващата jsp страница.

Изходен код:

```
<?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/xhtml-math11-
f.dtd">
```

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

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

```
<body>
<h1> Example3 </h1>
```

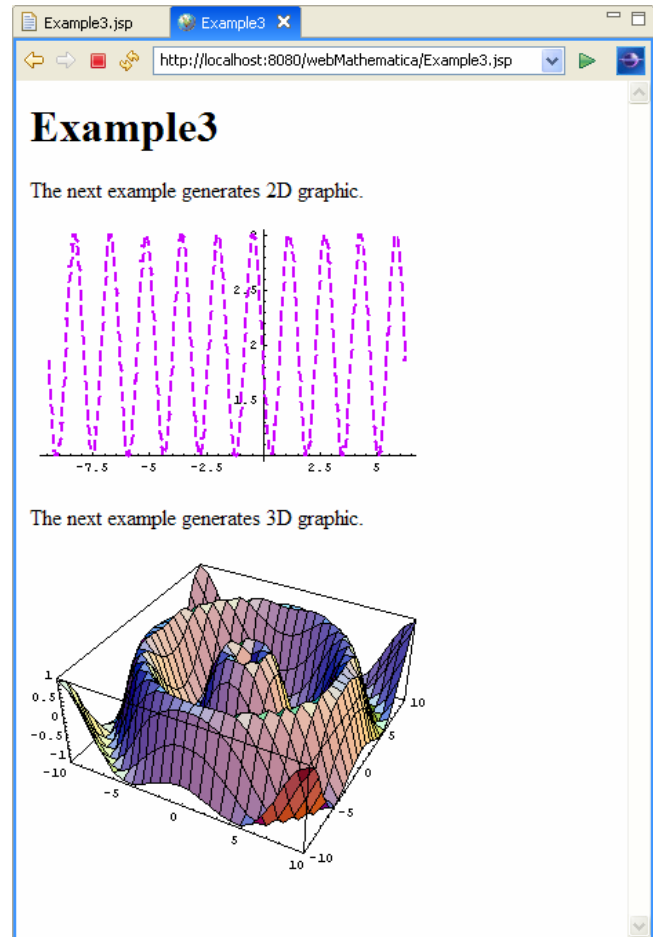
```
<msp:allocateKernel>
```

```
<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>
```



Можете да тествате този пример на интернет адрес:

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

Пример 4 - Проста интерактивна комуникация

Този пример рисува двумерна графика на функция, зададена от потребителя. Тази страница е по-сложна, защото съдържа HTML елементите **<form>** и **<input>**. Те са важни за предоставяне на интерактивност на клиента.

Елементът **<form>** е HTML блок, който съдържа входни елементи. Формата може да се активира с един **<input>** от тип `submit`, което праща името и стойностите, съответстващи на всеки `input` таг към сървъра.

Тук отварящият таг на формата съдържа два атрибута. Действието се насочва към URL, който се отваря с активирането на формата. В този случай, това е URL, който съответства на оригиналния скрипт. Методът атрибут съобщава на брауъра кой метод на HTTP да използва, тук е `post`-метод.

В дадения пример има четири входни тага, първите три тага позволяват на потребителя на страницата да въвежда текст, а четвъртият определя клавиша, при натискането на който ще се подава формата. Когато формата е подадена, информацията от входните елементи ще се изпрати към URL-то, определено от действието атрибут (в този пример това е същият `jsp` файл). Текстът, въведен в `input` тага, който използва името `func`, ще бъде присвоен на входната променлива `$$func`. Текстът, въведен в `input` тага, с името `min`, ще бъде присвоен на входната променлива `$$min` и т.н.

При първото отваряне на веб-страницата променливата `$$func` няма стойност. Когато се въведе текст в текстовото поле и се натисне бутонът **Evaluate**, `$$func` ще получи стойност, която се изпраща към сървъра. Същата операция ще се извърши и за другите два входни текстови тага. Ако искате входовете да се интерпретират и изпълнят от *Mathematica*, трябва да използвате една от MSP функциите. `mSPvalue` прехвърля към *Mathematica* интерпретирането на съдържанието на променливата.

Забележка : `name` е името на `html` променливата, а `$$name` е името на променливата на *Mathematica*

Командата `mSPBlock` е програмна конструкция, която тук проверява трите променливи `$$func`, `$$min`, `$$max`. Ако някоя от тях няма стойност, `mSPBlock` връща нулев стринг, което се получава, когато за пръв път отваряте страницата и няма да видите резултат. След това *Mathematica* интерпретира стойностите на трите променливи. Ако всичко е наред, резултатът от интерпретирането се замества във вторият аргумент или тялото на `mSPBlock`.

В този пример `func`, `min` и `max` се заместват с текущите стойности на `$$func`, `$$min` и `$$max`. След това се извършва изчислението, форматирането и поместването на резултата в `html` страницата, която се връща на клиента.

Следващият пример генерира графика на някаква функция `func`, зададена от потребителя в интервал, също зададен от потребителя в текстовите полета `min` и `max`. В първия момент на изчисленията са предвидени начални стойности по подразбиране, които са присвоени на променливите.

Исходен код:

```
<?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/xhtml-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[ $\sin[x]^2$ ]"</msp:evaluate>"
/><br/>
Enter a min value for plot <i>min</i> :
<input      type="text" name="min" size="20"
            value="<msp:evaluate>MSPValue[ $\sin[x]^2$ , "0"]</msp:evaluate>" /><br/>
Enter a max value for plot <i>max</i> :
<input      type="text" name="max" size="20"
            value="<msp:evaluate>MSPValue[ $\sin[x]^2$ , "10"]</msp:evaluate>" /><br/>

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

<msp:evaluate>
MSPBlock[ $\sin[x]^2$ ,  $\sin[x]^2$ ,  $\sin[x]^2$ ],
        MSPShow[Plot[ $\sin[x]^2$ , {x,  $\sin[x]^2$ ,  $\sin[x]^2$ },
                    PlotStyle -> {Dashing[{0.02}], Thickness[0.01], Hue[0.6]}]]
        ]
</msp:evaluate>

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

Може да тествате този пример на:

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

Example4.jsp Example4

http://localhost:8080/webMathematica/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.jsp Example4

http://localhost:8080/webMathematica/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 :

Потребителят може да променя стойностите на променливите **func**, **min** и **max** и да изпраща нова заявка за изчисление към сървъра. Ако всичко е наред, резултатът ще се покаже с същия прозорец.

Example4.jsp Example4

http://localhost:8080/webMathematica/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 :