المدرسة الوطنية للعلوم التطبيقية
ⵜⵉⵎⵙⵙⵓⵜ ⵜⴰⵏⴰⵟⵓⵔⵉⵜ ⵜⵓⵙⵙⴰⵏⵉⵏ
Ecole Nationale des Sciences Appliquées de Fès

جامعة سيدي محمد بن عبد الله
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵙⵉⴷⵉ ⵎⵓⵃⴰⵎⴻⴷ ⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ
Université Sidi Mohamed Ben Abdellah

Rapport de Tp1

Prise en main de l'environnement JEE dans Apache Tomcat

Realisé par:

Salma fassi

G.Info 2

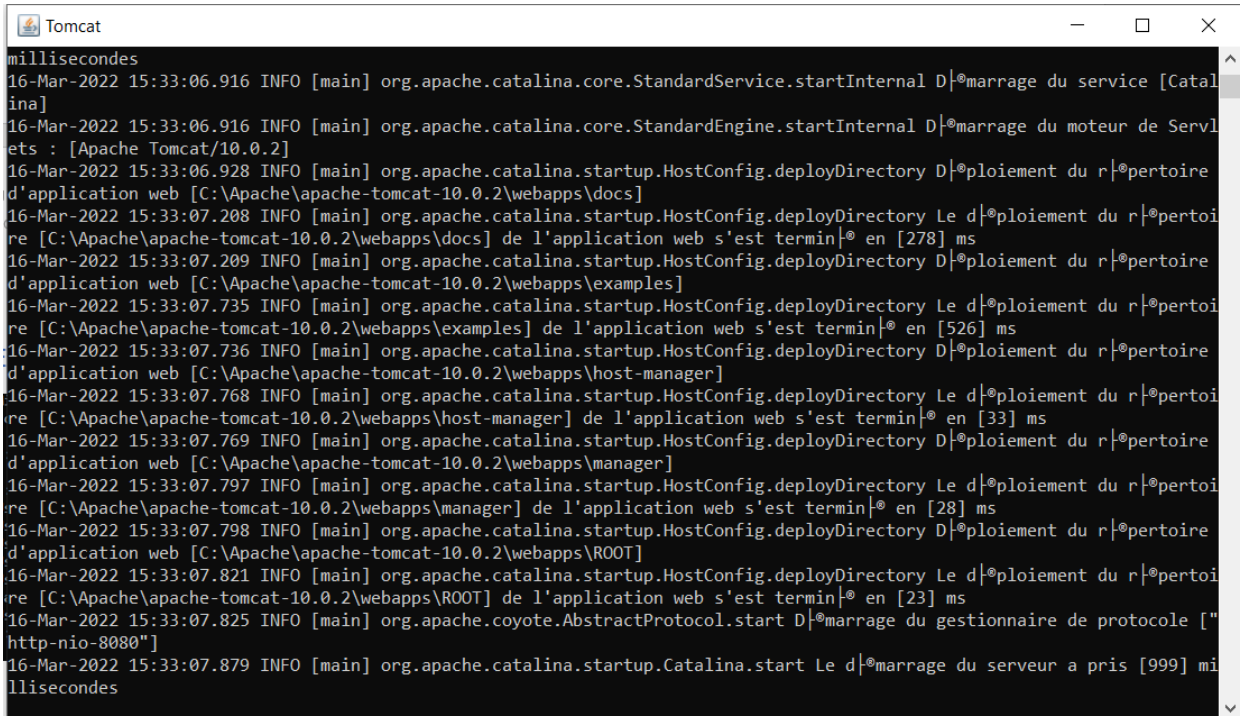
Encadré par:

Pr. Youness IDRISSI KHAMLICHI

Exercice 1 :

1. *Lancez Tomcat startup.bat dans le répertoire TOMCAT_HOME/bin :*

```
C:\Users\lenovo>cd C:\Apache\apache-tomcat-10.0.10\bin

C:\Apache\apache-tomcat-10.0.10\bin>startup.bat
Using CATALINA_BASE:   "C:\Apache\apache-tomcat-10.0.10"
Using CATALINA_HOME:   "C:\Apache\apache-tomcat-10.0.10"
Using CATALINA_TMPDIR: "C:\Apache\apache-tomcat-10.0.10\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk-17.0.1"
Using CLASSPATH:       "C:\Apache\apache-tomcat-10.0.10\bin\bootstrap.jar;C:\Apache\apache-tomcat-10.0.10\bin\tomcat-ju
li.jar"
Using CATALINA_OPTS:   ""
```

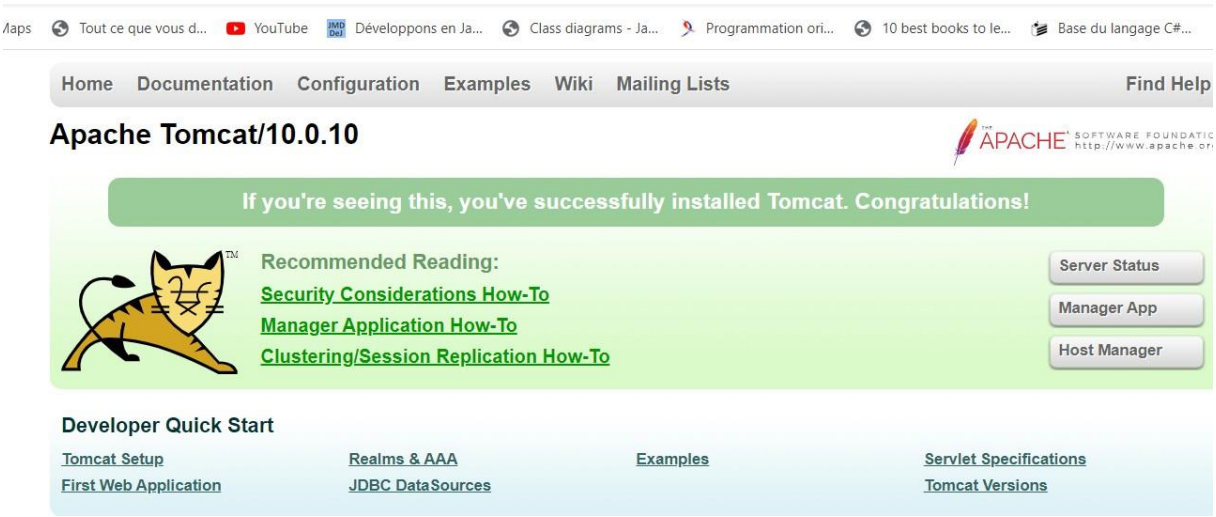


2. *le fichier server.xml :*

```
  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!-- Note:  A "Server" is not itself a "Container", so you may not
     define subcomponents such as "Valves" at this level.
     Documentation at /docs/config/server.html
 -->
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener" />
  -->
  <!-- APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
```

La balise principale	`<Server port="8005" shutdown="SHUTDOWN">`
Les balises filles et le nombre de leurs apparitions	*Listner, GlobalNamingRsources, Resource, Service, Engine, Engine, Realm, Host.*
La balise connector et ces attributs	*<connector executor= ”tomcat ThreadPool” port=”8080” protocol=”Http/1.1” connectionTimeOut=”20000” redirectPort=”8443” />*

3. *le fichier « tomcat-users.xml » :*

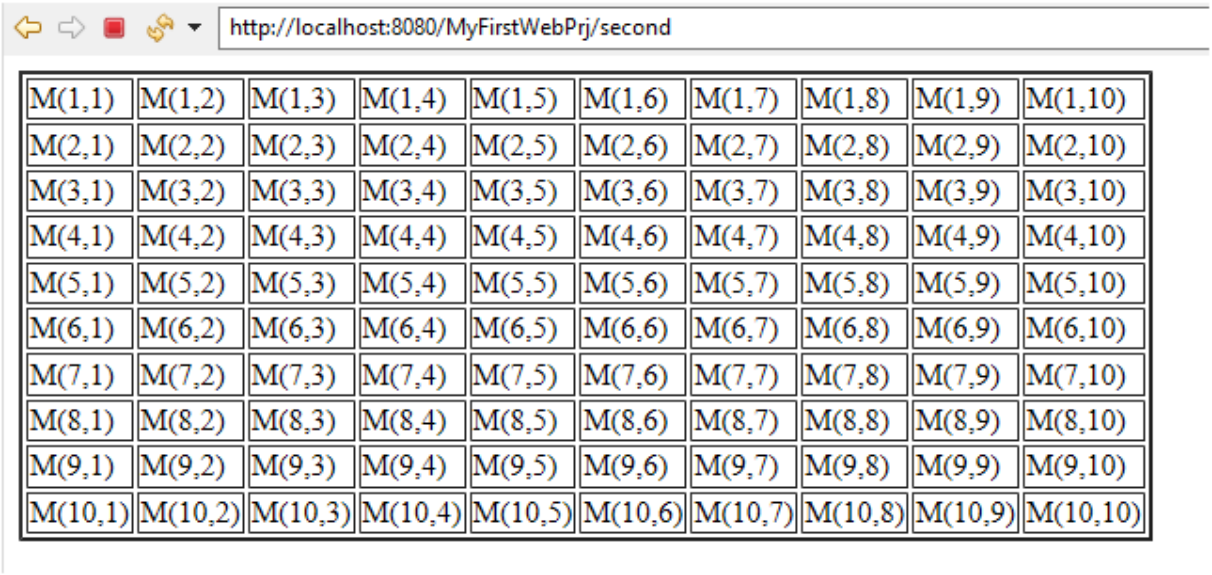


Exercice 2 :

1. un tableau de 10 cases par 10 :

```java
import java.io.IOException;
import java.io.PrintWriter;

import javax.sql.rowset.serial.SerialException;

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class servlet1 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */


    /**
     * @throws IOException
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        // TODO Auto-generated method stub
        //response.getWriter().append("Served at: ").append(request.getContextPath());

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<table border='2'>");
        for(var i=1; i<11; i++) {
            out.println("<tr>");
            for(var j=1; j<11 ; j++) {

                out.println("<td>"+ "M("+i+","+j+")"+"</td>");
            }
            out.println("</tr>");
        }
        out.println("</table>");

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws  IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
```

- *Affichage :*



M(1,1)	M(1,2)	M(1,3)	M(1,4)	M(1,5)	M(1,6)	M(1,7)	M(1,8)	M(1,9)	M(1,10)
M(2,1)	M(2,2)	M(2,3)	M(2,4)	M(2,5)	M(2,6)	M(2,7)	M(2,8)	M(2,9)	M(2,10)
M(3,1)	M(3,2)	M(3,3)	M(3,4)	M(3,5)	M(3,6)	M(3,7)	M(3,8)	M(3,9)	M(3,10)
M(4,1)	M(4,2)	M(4,3)	M(4,4)	M(4,5)	M(4,6)	M(4,7)	M(4,8)	M(4,9)	M(4,10)
M(5,1)	M(5,2)	M(5,3)	M(5,4)	M(5,5)	M(5,6)	M(5,7)	M(5,8)	M(5,9)	M(5,10)
M(6,1)	M(6,2)	M(6,3)	M(6,4)	M(6,5)	M(6,6)	M(6,7)	M(6,8)	M(6,9)	M(6,10)
M(7,1)	M(7,2)	M(7,3)	M(7,4)	M(7,5)	M(7,6)	M(7,7)	M(7,8)	M(7,9)	M(7,10)
M(8,1)	M(8,2)	M(8,3)	M(8,4)	M(8,5)	M(8,6)	M(8,7)	M(8,8)	M(8,9)	M(8,10)
M(9,1)	M(9,2)	M(9,3)	M(9,4)	M(9,5)	M(9,6)	M(9,7)	M(9,8)	M(9,9)	M(9,10)
M(10,1)	M(10,2)	M(10,3)	M(10,4)	M(10,5)	M(10,6)	M(10,7)	M(10,8)	M(10,9)	M(10,10)

2. *envoyer la réponse vers un fichier Excel :*

```java
import java.io.IOException;
import java.io.PrintWriter;

import javax.sql.rowset.serial.SerialException;

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class servlet1 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */


    /**
     * @throws IOException
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        // TODO Auto-generated method stub
        //response.getWriter().append("Served at: ").append(request.getContextPath());

        response.setContentType("application/vnd.ms-excel");
        response.setHeader("Content-Disposition","attachment;filename=MaServlet.xls");
        PrintWriter out = response.getWriter();
        out.println("<table border='2'>");
        for(var i=1; i<11; i++) {
            out.println("<tr>");
            for(var j=1; j<11 ; j++) {

                out.println("<td>"+ "M("+i+","+j+")"+"</td>");
            }
            out.println("</tr>");
        }
        out.println("</table>");

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws  IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
```

- *Affichage :*

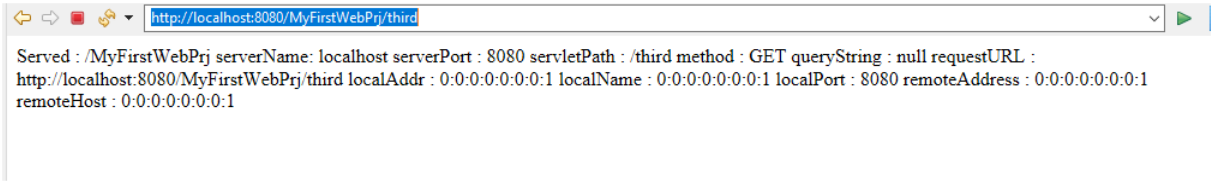
	A	B	C	D	E	F	G	H	I	J	K
1	M(1,1)	M(1,2)	M(1,3)	M(1,4)	M(1,5)	M(1,6)	M(1,7)	M(1,8)	M(1,9)	M(1,10)	
2	M(2,1)	M(2,2)	M(2,3)	M(2,4)	M(2,5)	M(2,6)	M(2,7)	M(2,8)	M(2,9)	M(2,10)	
3	M(3,1)	M(3,2)	M(3,3)	M(3,4)	M(3,5)	M(3,6)	M(3,7)	M(3,8)	M(3,9)	M(3,10)	
4	M(4,1)	M(4,2)	M(4,3)	M(4,4)	M(4,5)	M(4,6)	M(4,7)	M(4,8)	M(4,9)	M(4,10)	
5	M(5,1)	M(5,2)	M(5,3)	M(5,4)	M(5,5)	M(5,6)	M(5,7)	M(5,8)	M(5,9)	M(5,10)	
6	M(6,1)	M(6,2)	M(6,3)	M(6,4)	M(6,5)	M(6,6)	M(6,7)	M(6,8)	M(6,9)	M(6,10)	
7	M(7,1)	M(7,2)	M(7,3)	M(7,4)	M(7,5)	M(7,6)	M(7,7)	M(7,8)	M(7,9)	M(7,10)	
8	M(8,1)	M(8,2)	M(8,3)	M(8,4)	M(8,5)	M(8,6)	M(8,7)	M(8,8)	M(8,9)	M(8,10)	
9	M(9,1)	M(9,2)	M(9,3)	M(9,4)	M(9,5)	M(9,6)	M(9,7)	M(9,8)	M(9,9)	M(9,10)	
10	M(10,1)	M(10,2)	M(10,3)	M(10,4)	M(10,5)	M(10,6)	M(10,7)	M(10,8)	M(10,9)	M(10,10)	
11											
12											
13											

Exercice 3 :

1. *Récupération des informations sur l'url :*

```
import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
public class servlet2 extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        //response.getWriter().append("Served at: ").append(request.getContextPath());

        PrintWriter out= response.getWriter();

        out.println("Served : " + request.getContextPath());
        out.println("serverName: " + request.getServerName());
        out.println(" serverPort : " + request.getServerPort());
        out.println(" servletPath : " + request.getServletPath());
        out.println(" method : " + request.getMethod());
        out.println(" queryString  : " + request.getQueryString());
        out.println(" requestURL : " + request.getRequestURL());
        out.println(" localAddr : " + request.getLocalAddr());
        out.println(" localName : " + request.getLocalName());
        out.println(" localPort : " + request.getLocalPort());
        out.println(" remoteAddress : " + request.getRemoteAddr());
        out.println(" remoteHost : " + request.getRemoteHost());

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
```

- *Affichage :*



2.

a) *Le formulaire en JSP :*

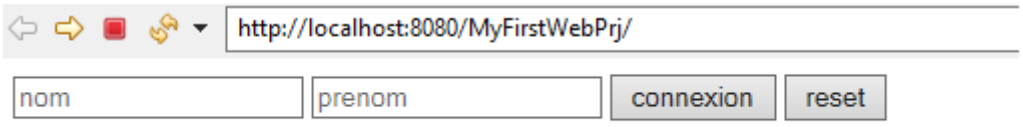
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method='post' action='formulaire' >
<input type='text' name='nom' placeholder='nom'/>
<input type='text' name='prenom' placeholder='prenom'/>
<input type='submit' value='connexion'>
<input type='reset' value='reset'>
</form>
</body>
</html>
```

- *Le servlet pour traiter le formulaire :*

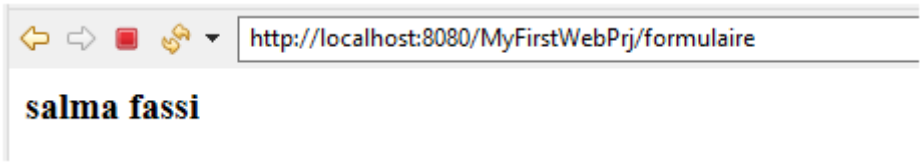
```java
import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class servletform extends HttpServlet {
    protected void doGet(HttpServletRequest req,HttpServletResponse res) throws IOException {

    }
    protected void doPost(HttpServletRequest req,HttpServletResponse res) throws IOException {
        String nom=req.getParameter("nom");
        String prenom=req.getParameter("prenom");
                res.setContentType("text/html");
                PrintWriter out=res.getWriter();
                if(nom.length()!=0 && prenom.length()!=0) {
                    out.print("<h3>"+nom+" "+prenom+"</h3>");
                }
                else {
                    out.print("veuillez remplir");
                }
    }
}
```

- *L'exécution :*

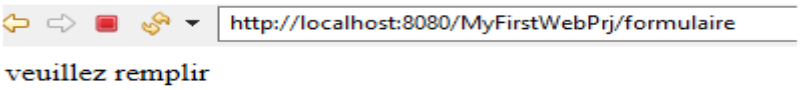
Le formulaire :



Cas de succès :



Cas d'échec :



b) *Le longueur de texte récupéré :*

1. *En utilisant getcontentLength :*

Pour récupérer la longueur de texte, on utilise La fonction getcontentLength() :

```
out.print("la longueur de texte est "+req.getContentLength());
```

Résultat :

salma fassi

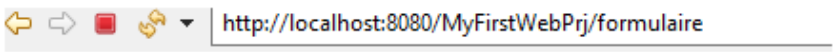
la longueur de texte est 22

2. *En utilisant length() :*

Si on applique la propriété length() sur le champ nom et prénom

```
out.print("la longueur de texte est "+(nom+prenom).length());
```

Résultat :



salma fassi

la longueur de texte est 10

- *Conclusion : la différence entre deux méthodes est gigantesque*

c) *Apres qu'on copie le code source, on obtient plus de détails concernant l'entête et ses contenus.*

3. *Affichage des informations sur client :*

La page servlet « echo.java » :

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class echo extends HttpServlet {
    protected void doGet(HttpServletRequest req,HttpServletResponse res) throws IOException {
        PrintWriter out=res.getWriter();
        out.println("<h1>Servlet Echo</h1><br>");
        out.println("Request method: "+req.getMethod()+"</br>");
        out.println("Request url: "+req.getRequestURL()+"</br>");
        out.println("Request url: "+req.getProtocol()+"</br>");
        Enumeration<String> headerNames=req.getHeaderNames();
        out.println("<h2><center>les entetes provenant du client</center></h2>");
        out.println("<table border=2>");
        out.println("<tr><th ><Header Name/th><th>Header Value</th></tr>");
        while(headerNames.hasMoreElements()) {
            String name=headerNames.nextElement();
            String value=req.getHeader(name);
            out.println("<tr><th>"+name+"</th><th>"+value+"</th></tr>");
        }
        out.println("</table>");
    }
    protected void doPost(HttpServletRequest req,HttpServletResponse res) throws IOException {
        doGet(req,res);
    }
}
```

➢ *L'exécution :*

Servlet Echo

Request method: GET
Request url: http://localhost:8080/MyFirstWebPrj/echo
Request url: HTTP/1.1

les entetes provenant du client

	Header Value
accept	image/gif, image/jpeg, image/pjpeg, application/x-ms-application, application/xaml+xml, application/x-ms-xbap, */*
accept-language	en-US,en;q=0.91,fr-FR;q=0.82,fr-BE;q=0.73,fr;q=0.64,es-ES;q=0.55,es;q=0.45,ar-MA;q=0.36,ar-SA;q=0.27,ar;q=0.18,qaa-Latn;q=0.091
cache-control	no-cache
ua-cpu	AMD64
accept-encoding	gzip, deflate
user-agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64; Trident/7.0; rv:11.0) like Gecko
host	localhost:8080
connection	Keep-Alive
cookie	JSESSIONID=9C44939842A0A7D8880A8E8575A75F12

Exercice 4 :

1. *L'affichage de factorielle de nombre entre 0 et 9 :*

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%!
public int fact(int i){
    if(i==0 || i==1){
        return 1;
    }
    else{
         return i*fact(i-1);
    }
}
%>
<%
for(int x=0;x<10;x++)
{
%>
<h3><%= fact(x) %></h3>
<% } %>
</body>
</html>
```

- *L'exécution :*

1

1

2

6

24

120

720

5040

40320

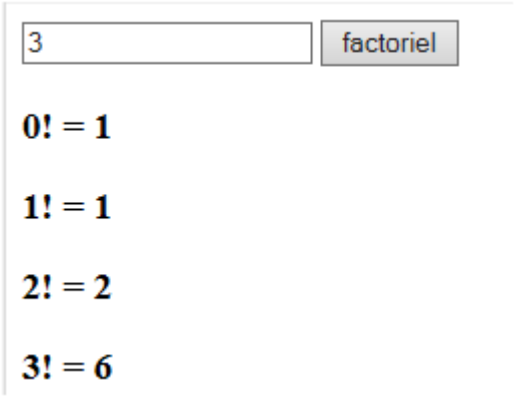
362880

2. *Le factorielle de nombre entre 0 et n :*

Le fichier JSP :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%!
public int fact(int i){
    if(i==0 || i==1){
        return 1;
    }
    else{
         return i*fact(i-1);
    }
}
%>

<form method="get">
<input type="number" name="nbr" value="" placeholder="entrez un integer">
<input type="submit" value="factoriel">
</form>
<%
if(request.getParameter("nbr")!=null){
     int number=Integer.parseInt(request.getParameter("nbr"));

%>
<%
for(int i=0;i<=number;i++){
%>
<h3><%= i %>! = <%= fact(i) %></h3>
<%}
}
%>
</body>
</html>
```

- *L'exécution :*



3. *Le factoriel d'un nombre lu au clavier :*

Le fichier JSP

```
<%!
public int fact(int i){
    if(i==0 || i==1){
        return 1;
    }
    else{
         return i*fact(i-1);
    }
}
%>
<%
if(request.getParameter("nbr")!=null){
     int number=Integer.parseInt(request.getParameter("nbr"));

%>
<form method="get">
<input type="number" name="nbr" value=<%= number%> placeholder="entrez un integer">
<input type="submit" value="factoriel">
</form>
<%}
if(request.getParameter("nbr")!=null){
     int number=Integer.parseInt(request.getParameter("nbr"));

%>

<h3><%= number %>! = <%= fact(number) %></h3>
<%}

%>
```

- *L'exécution :*

3! = 6