It gives global instructions to jsp page compiler to generate the code in JES class like importing packages ….etc.

Standard Syntax:

============

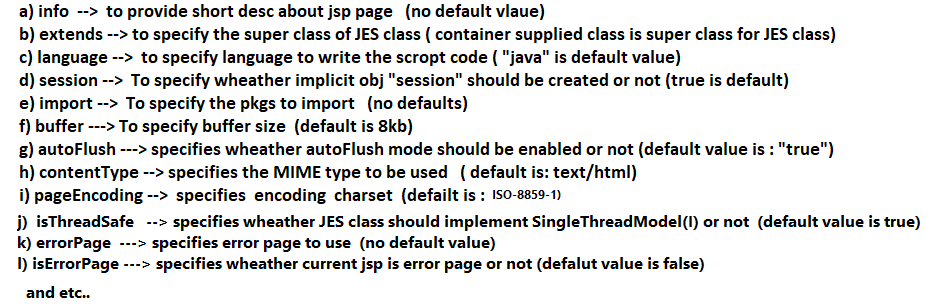


Xml syntax::

=========



The attributes are

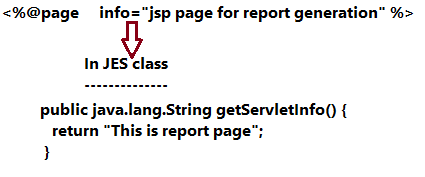


**1.Info:-**

**=====**

* It does not have default value.
* It is useful to specify short description about jsp page.
* Helps other programmer to know the purpose of jsp page when its source code is opened.
* It is optional attribute to place.

**Example:**



The getServletinfo() of Jakarta.servlet.Servlet(I) will be implemented in JES class having the message kept in “info” attribute of <%@page%> directive tag.

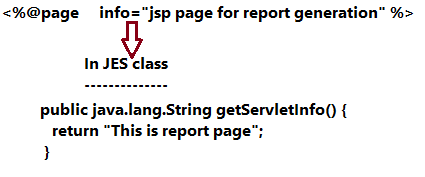
If “info” attribute is not taken in <%page%> tag then getServletinfo() method in JES class will not be overridden.

2. language

=========

* Default value is “java” and “java” is only allowed value.
* Allows to specify the programming that can be used to write script code in scripting tags.
* As of now “java” is only one language that is allowed to pass .. So “java” is default value and the only one value that is allowed to pass.

Example:

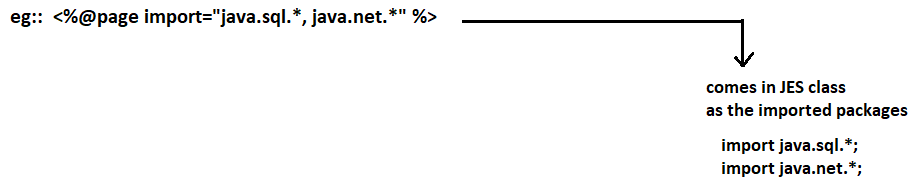


Note:- No code comes in the JES class with respect to this attribute.. but only “java” is allowed to pass.

3.import

======

* This attribute does not has default value.
* It allows to specify the multiple packages to import to JES class.
* We can specify multiple packages as comma separated list of values.



This attribute can be repeated either in single page directive tag or in multiple page directive tags of a jsp page either with same values or with different values.

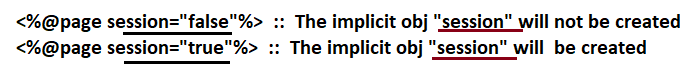
By default every JES class get the following 3 packages to import

1. Jakarta.servlet.\*;
2. Jakarta.servlet.http.\*
3. Jakarta.servlet.jsp.\*

4.session:

=======

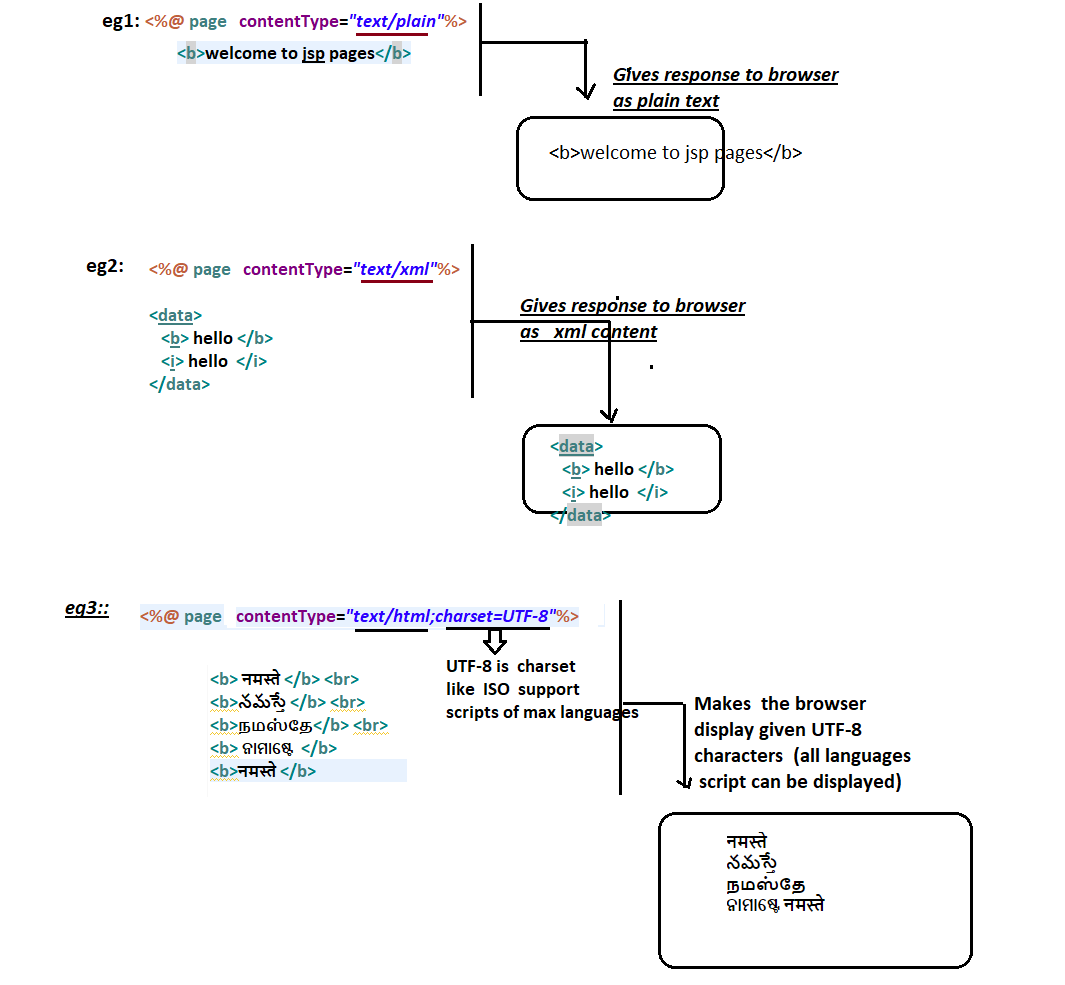
* Its default value is true.
* Specifies wheather implicit obj “session” should be created or not because we do not need Session object if session tracking is not enabled.



5.ContentType:

==========

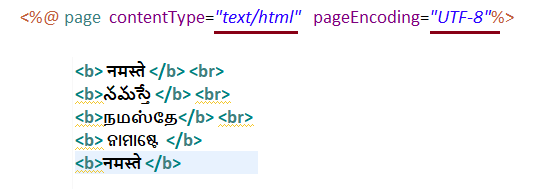
* This attribute has text/html;charset=ISO-8859-1 as default value.
* Specifies the given **MIME** type as response content type to display the generated output in different formats.



**6.pageEncoding:**

**=============**

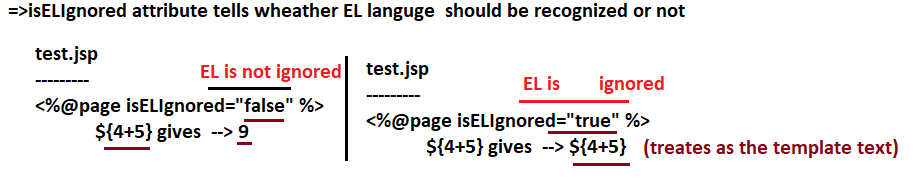
Specify only MIME type in “contentType” and charset in the “pageEncoding”. (This is best practice).



7.**isELIgnored:**

**===========**

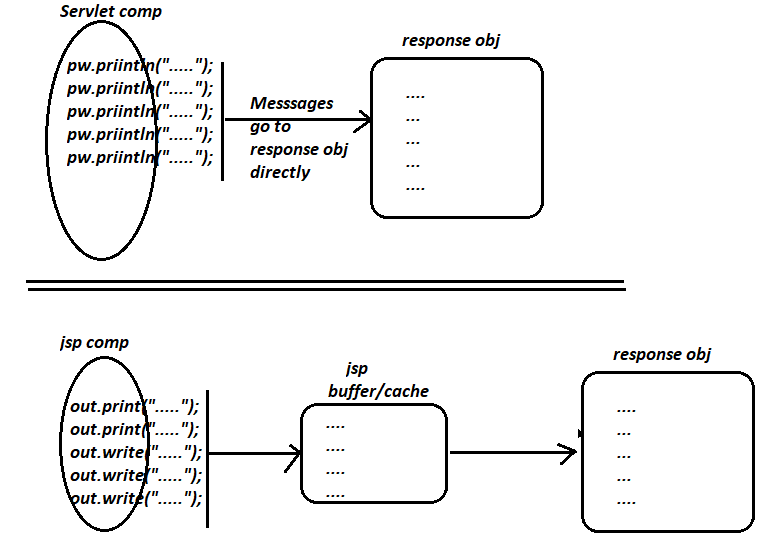
We should avoid or minimize java code in jsp pages and prefere tags based programming in jsp pages. But doing arithmetic and logical operations without java code is impossible. So EL(Expression language) is introduced as alternate which will be converted to java code by jsp page compiler.



8.Buffer:

=======

* Buffer/cache is a temporary memory that holds data for temporary period.
* Servlet comp generated messages(output)goes to response obj directly where as the jsp comp generated messages(output) goes to response object through buffer.



The messages or outputs given by jsp first goes jsp buffer .. if the buffer filled up or flush() is called or jsp page execution is called then the buffer content/messages go to response obj.

The process of moving buffer content its real destination is called “**flushing”**. If it is happening automatically then it is called auto flushing.

On jsp buffer the Autoflushing takes place in the following situations:

1. When buffer is filed up.
2. Though buffer is not filled up if the jsp page execution is completed.

=> The default size of jsp buffer is 8kb and it can controlled using “buffer” attribute.

=> By default autoflush is enabled on jsp buffer and it can be enabled or disabled using “autoFlush” attribute.

=> buffer=”non” disables the buffer i.e jsp page output will be written to response obj directly.

9.autoFlush:

=========

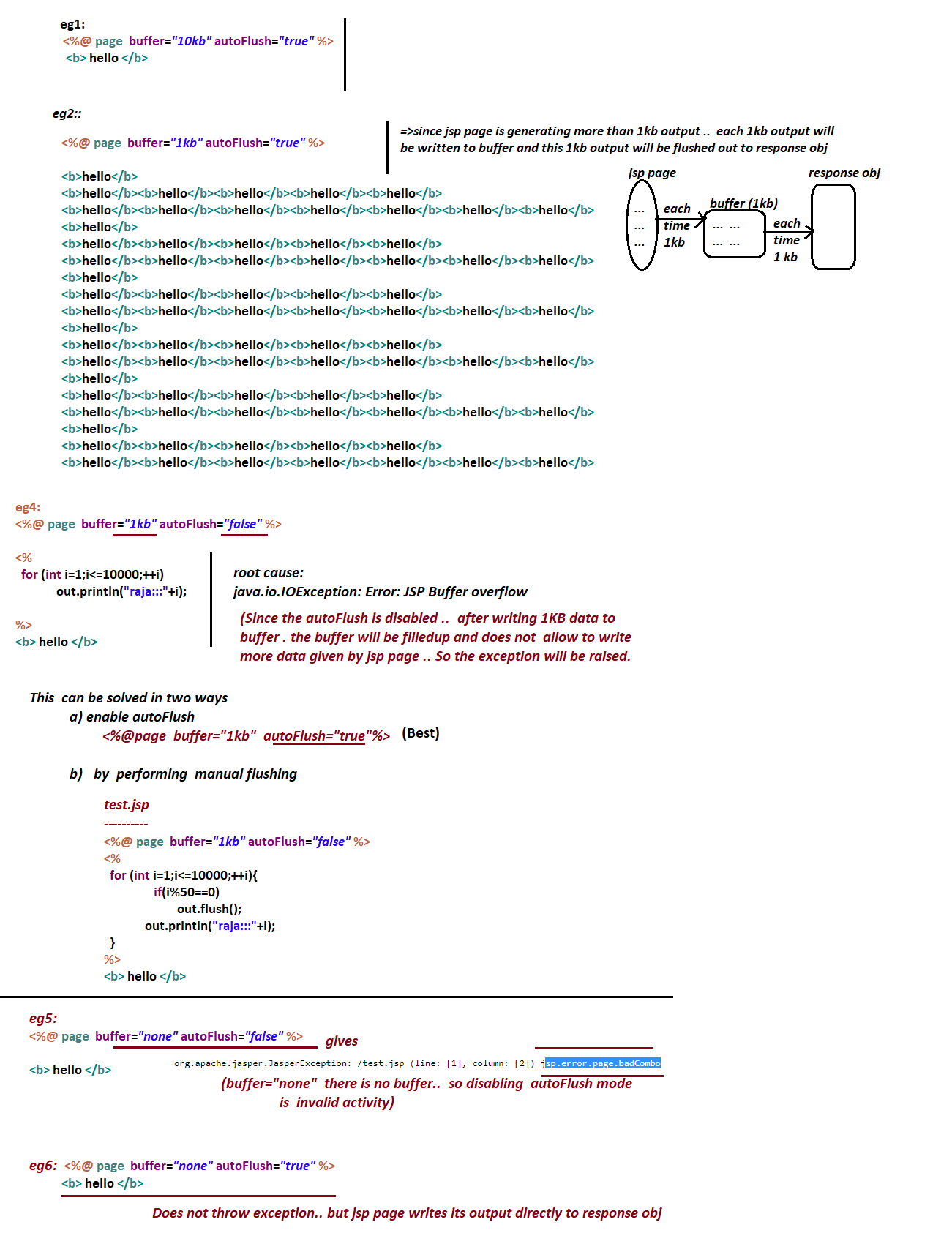
Specifies wheather autoFlush on buffer should be enabled or not.

True -🡪 enables the autoFlushing.

False 🡪 disables the autoFlushing.

Examples related to both buffer and autoflush

================================



**10.errorPage:**

=========

* Allows us to specify error page(html or jsp page) which will be executed only when exceptions are raised in the current jsp page.

In test.jsp(main jsp page)

<[%@page errorPage=”error.jsp”%](mailto:%25@page%20errorPage=”error.jsp”%25)>

* This error page cfg is useful to display exception related messages to enduser as non-technical guiding messages to enduser.

11.isErrorPage

===========

* Specifies wheather current jsp page should taken as error page or not.

