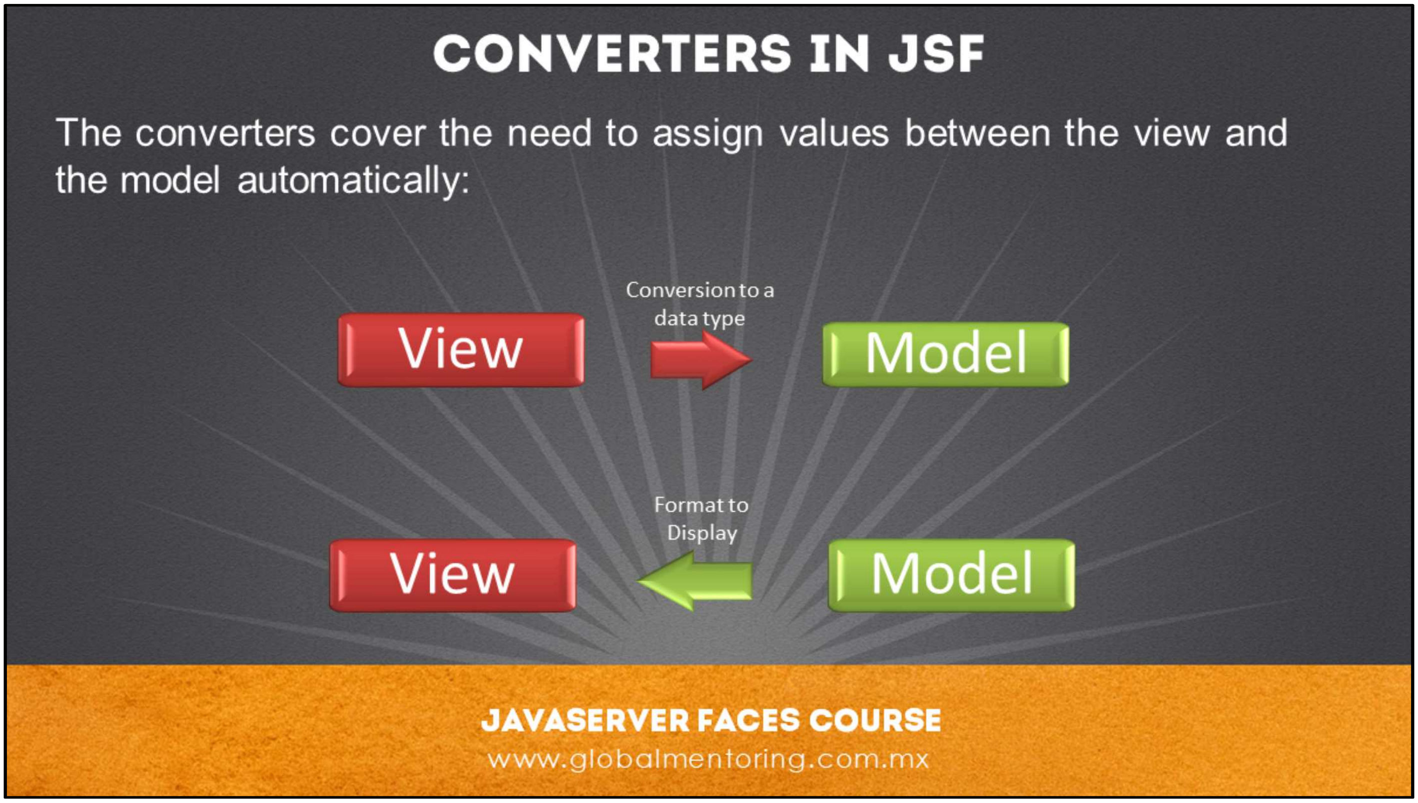
Hello, Ubaldo Acosta greets you again. I hope you're ready to start with this lesson ..

We are going to study the topic of Converters in JSF.

Are you ready? Come on!



CONVERTERS IN JSF

The converters cover the need to assign values between the view and the model automatically:

View

Conversion to a data type

Model

View

Format to Display

Model

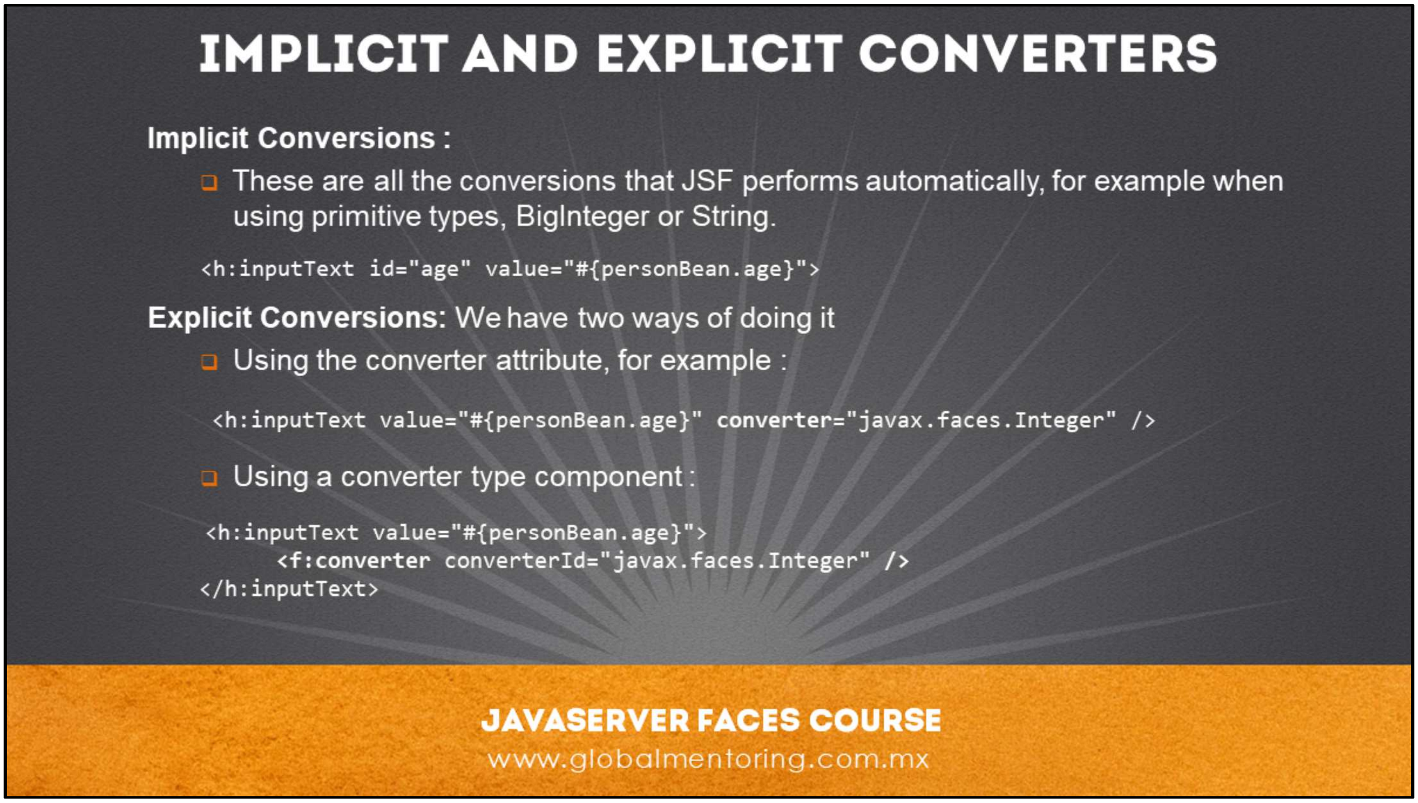
JAVASERVER FACES COURSE

www.globalmentoring.com.mx

The converters cover the need to assign values between the view and the model automatically, handling different types than the default (String). For example, converting a date chain and assigning it directly to the property of our model using the same type.

In the same way, it covers the need to display the information of the model with a specific format, for example a format of a date, a decimal value with certain positions, etc.

Next we will see the types of converters managed in JSF.



IMPLICIT AND EXPLICIT CONVERTERS

Implicit Conversions :

- These are all the conversions that JSF performs automatically, for example when using primitive types, BigInteger or String.

```
<h:inputText id="age" value="#{personBean.age}">
```

Explicit Conversions: We have two ways of doing it

- Using the converter attribute, for example :

```
<h:inputText value="#{personBean.age}" converter="javax.faces.Integer" />
```

- Using a converter type component :

```
<h:inputText value="#{personBean.age}">
      <f:converter converterId="javax.faces.Integer" />
</h:inputText>
```

JAVASERVER FACES COURSE
www.globalmentoring.com.mx

In JSF we have different ways of doing conversions. We can make implicit and explicit conversions.

In addition, JSF has standard converters, which we can use immediately for certain types of data. Some examples are:

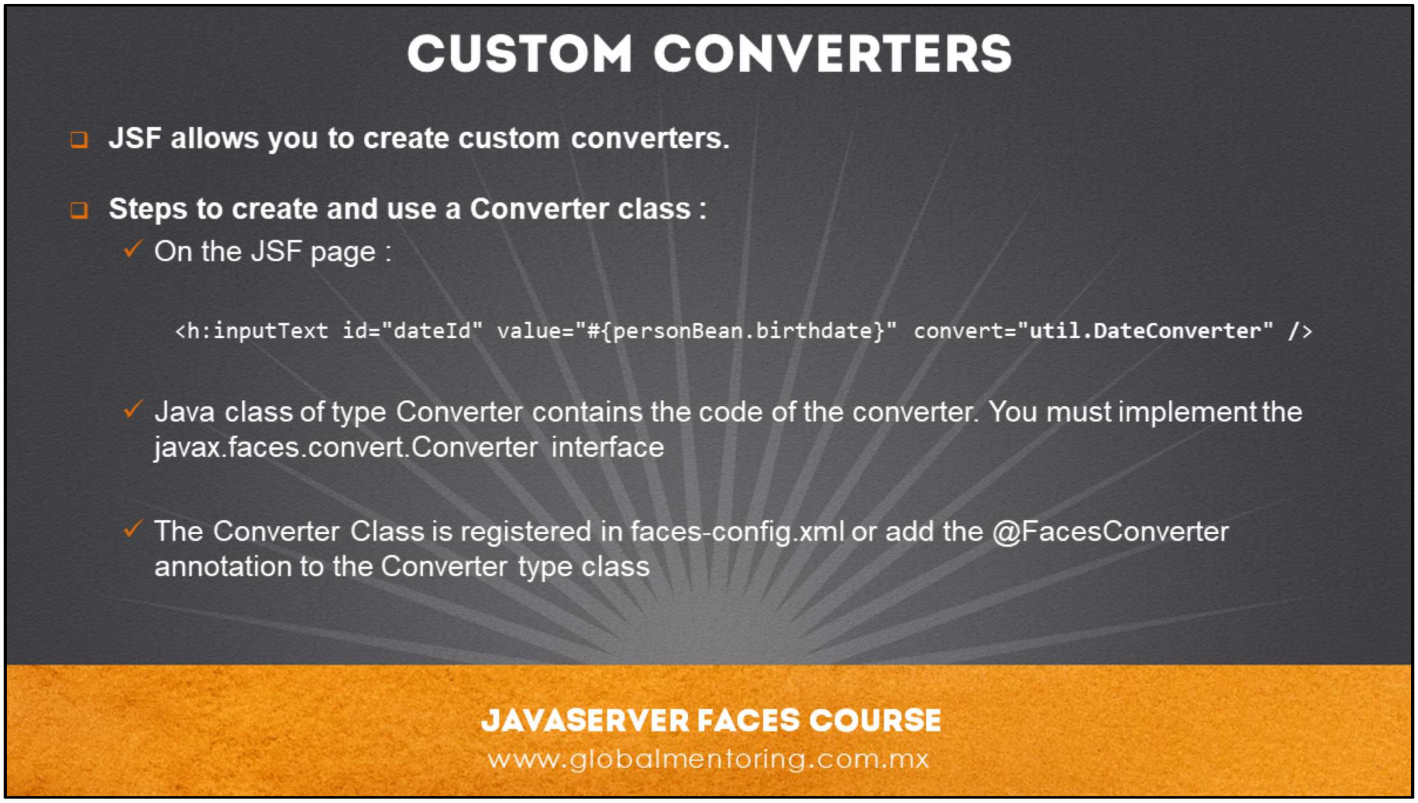
Number Converter: Convert the value of a component to some subclass of java.lang.Number.

Ex. <F: convertNumber type = "currency" />:

Dates Converter: Convert to value of a component to the java.util.Date class.

Ex. <F: converterDateTime pattern = "dd/MM/yyyy" />

To consult a list of formats for string, consult the java.text.SimpleDateFormat class of the Java API.



CUSTOM CONVERTERS

- JSF allows you to create custom converters.
- Steps to create and use a Converter class :
  - ✓ On the JSF page :

```
<h:inputText id="dateId" value="#{personBean.birthdate}" convert="util.DateConverter" />
```

  - ✓ Java class of type Converter contains the code of the converter. You must implement the javax.faces.convert.Converter interface
  - ✓ The Converter Class is registered in faces-config.xml or add the @FacesConverter annotation to the Converter type class

JAVASERVER FACES COURSE
www.globalmentoring.com.mx

JSF allows you to create custom converters. The elements are:

a) The JSF tag, for example:

```
<h:inputText id=“dateId" value="#{personBean.birthDate}"
convert="util.DateConverter" />
```

b) Java class: Contains the code that implements the converter, must implement the interface javax.faces.convert.Converter, we must also overwrite the method getAsObject (...) or getAsSring (...) depending on the type of conversion required.

c) The converter must be registered in the faces-config.xml file or the annotation can be used @ javax.faces.convert.FacesConverter

We can throw an exception with the following code:

```
FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_ERROR,
“Conversion Error", “Invalid Date");
throw new ConverterException(message);
```

ONLINE COURSE

JAVASERVER FACES (JSF)

By: Eng. Ubaldo Acosta

Global
Mentoring

JAVASERVER FACES COURSE

www.globalmentoring.com.mx