FORMATING AND VALIDATION

Number Format

- NumberFormat helps you to format and parse numbers for any locale. Your code can be completely independent of the locale conventions for decimal points, thousands-separators, or even the particular decimal digits used, or whether the number format is even decimal.

NumberFormat

- NumberFormat object in java.text package.
- New instance:
  - NumberFormat cnf= NumberFormat.get…Instance(Locale);

Method get a installs of number format

Set Locale to using format of this local. EX format of currency

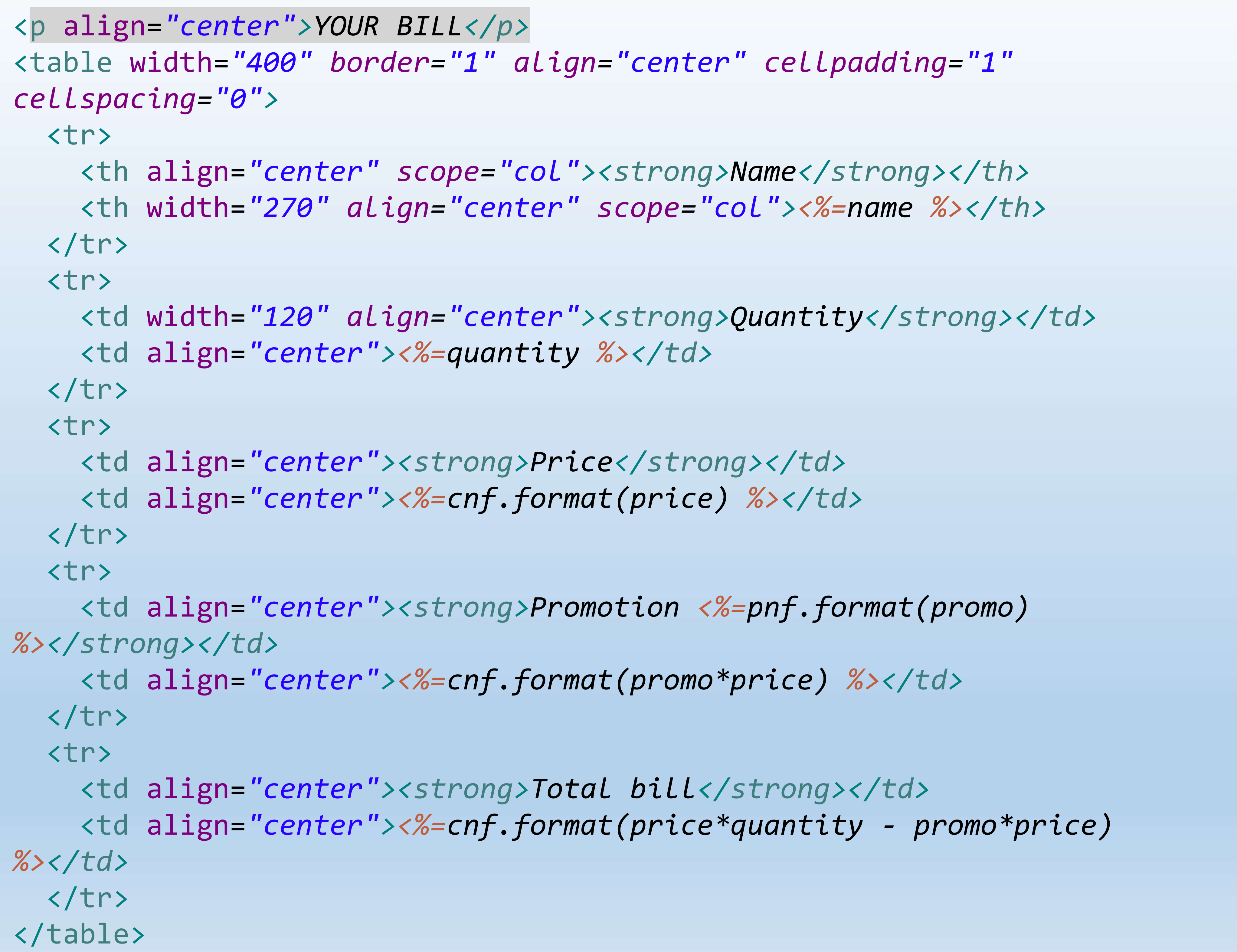
- Set properties for NumberFormat instance
  - pnf.propertiesMethod(parameters);
  - Ex: nf.setMaximumIntegerDigits(6);

- Using NumberFormat instance
  - <%=nf.format(number) %>
  - String data= nf.format(number) ;

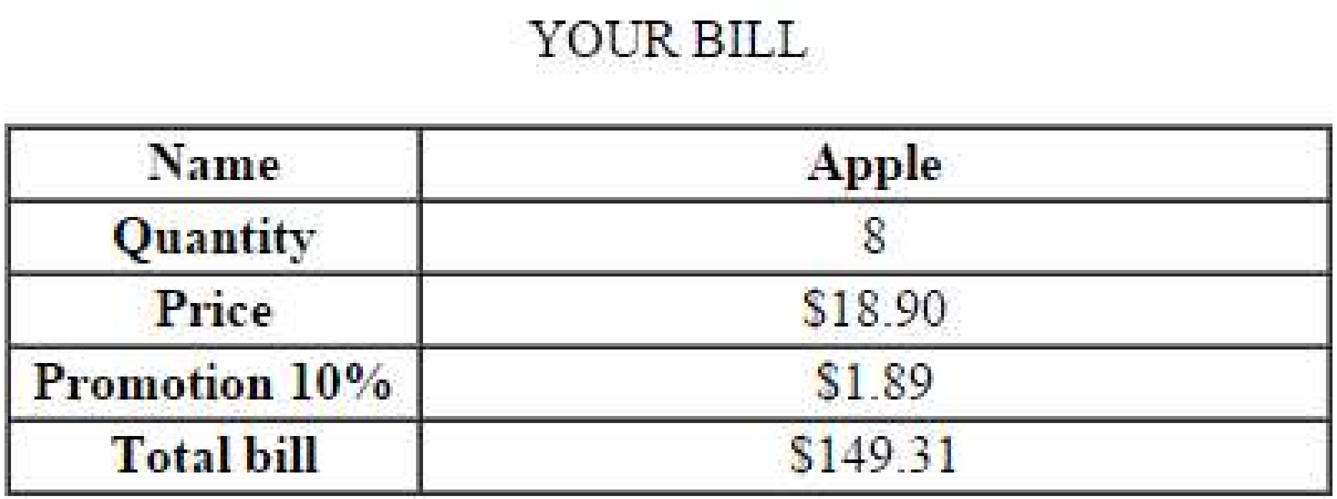
- Set minimum or maximum number of fraction digits used
  - NumberFormat nf= NumberFormat.getInstance();
  - nf.setMaximumFractionDigits(2);
  - nf.setMinimumFractionDigits(2);
  - nf.format(1234.567) → 1234.57
- Currency format
  - NumberFormat nf= NumberFormat.getCurrencyInstance();
  - nf.format(4.567) → $4.57
- Percent format
  - NumberFormat nf= NumberFormat.getPercentInstance();
  - nf.format(0.6) → 60%

```
<%
NumberFormat cnf= NumberFormat.getCurrencyInstance();
NumberFormat pnf= NumberFormat.getPercentInstance();

int quantity= 8;
double price=18.9;
double promo=0.1;
String name="Apple";
%>
```



```
<p align="center">YOUR BILL</p>
<table width="400" border="1" align="center" cellpadding="1"
cellspacing="0">
  <tr>
    <th align="center" scope="col"><strong>Name</strong></th>
    <th width="270" align="center" scope="col"><%=name %></th>
  </tr>
  <tr>
    <td width="120" align="center"><strong>Quantity</strong></td>
    <td align="center"><%=quantity %></td>
  </tr>
  <tr>
    <td align="center"><strong>Price</strong></td>
    <td align="center"><%=cnf.format(price) %></td>
  </tr>
  <tr>
    <td align="center"><strong>Promotion <%=pnf.format(promo)
%></strong></td>
    <td align="center"><%=cnf.format(promo*price) %></td>
  </tr>
  <tr>
    <td align="center"><strong>Total bill</strong></td>
    <td align="center"><%=cnf.format(price*quantity - promo*price)
%></td>
  </tr>
</table>
```

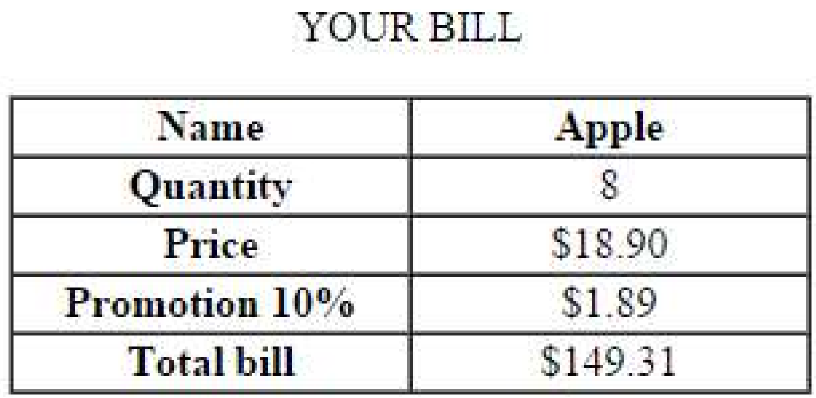


YOUR BILL

Name	Apple
Quantity	8
Price	$18.90
Promotion 10%	$1.89
Total bill	$149.31

Format with localtion

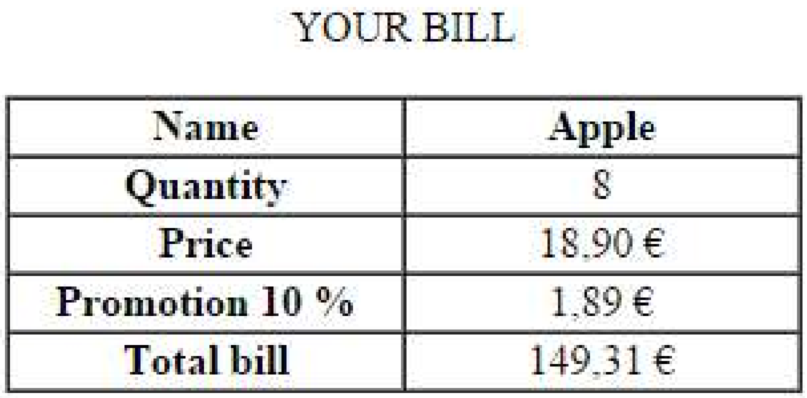
- Formats use different conventions in different parts of the world!
- Your applications must adjust to this!



YOUR BILL

Name	Apple
Quantity	8
Price	$18.90
Promotion 10%	$1.89
Total bill	$149.31

US localtion



YOUR BILL

Name	Apple
Quantity	8
Price	18,90 €
Promotion 10 %	1,89 €
Total bill	149,31 €

France localtion

- Every computer has default locale. You can it from computer.
  - Locale df=Locale.getDefault();
- Get locale from list locale of class **Locate**
  - Locale.(locale name);
  - Locale fr=Locale.FRANCE;
- Use Locale
  - Locale **df**=Locale.getDefault();
  - NumberFormat nf= NumberFormat.getInstance(**df**);
  - NumberFormat pnf= NumberFormat.getPercentInstance(**df**);

Date format

- Formats of dates also locale specific
- Formats of dates with a pattern.

Pattern	Output
dd.MM.yy	30.06.09
yyyy.MM.dd G 'at' hh:mm:ss z	2009.06.30 AD at 08:29:36 PDT
EEE, MMM d, ''yy	Tue, Jun 30, '09
h:mm a	8:29 PM
H:mm	8:29
H:mm:ss:SSS	8:28:36:249
K:mm a,z	8:29 AM,PDT
yyyy.MMMMM.dd GGG hh:mm aaa	2009.June.30 AD 08:29 AM

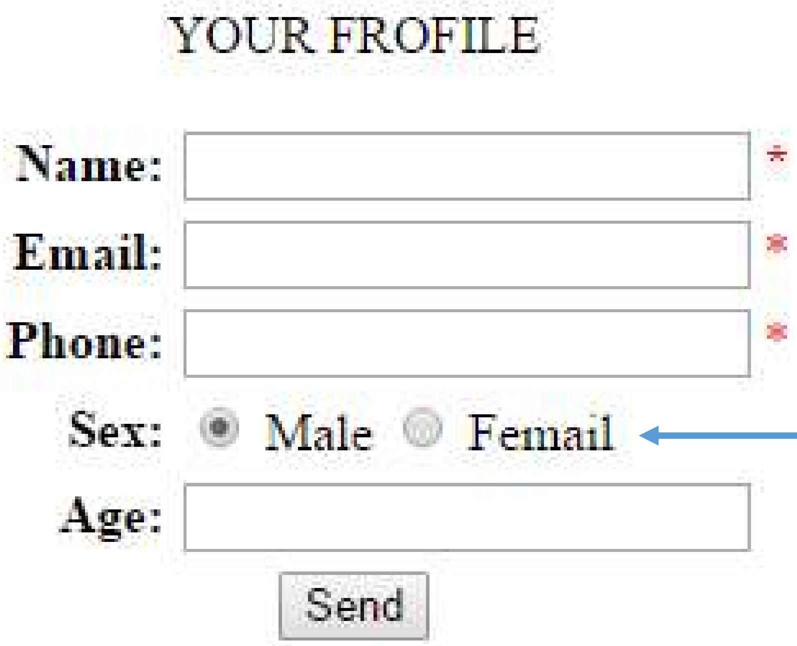
- To format Dates you use class SimpleDateFormat.
- New instance:
  - SimpleDateFormat dateFormat=new SimpleDateFormat(date pattern);
- Using SimpleDateFormat:
  - dateFormat.format(date)
- EX:
  - `SimpleDateFormat dateFormat=new SimpleDateFormat("dd/MM/yyyy");`
  - `String today=dateFormat.format(new Date());`

VALIDATION

- Detecting user error
  - Input error
  - Data type
  - Checking data with DB.
- Help users input
  - Show hint when user input.
  - Checking some error before submit form.
  - Providing information or how to fix error
  - Using input type and required in input field to prevent error

Form control hint

- Using character “*” with red color with required field.
- Set default value with radio button group and combo box.
- Show some example with input field use pattern.



YOUR FROFILE

Name: ________ *

Email: ________ *

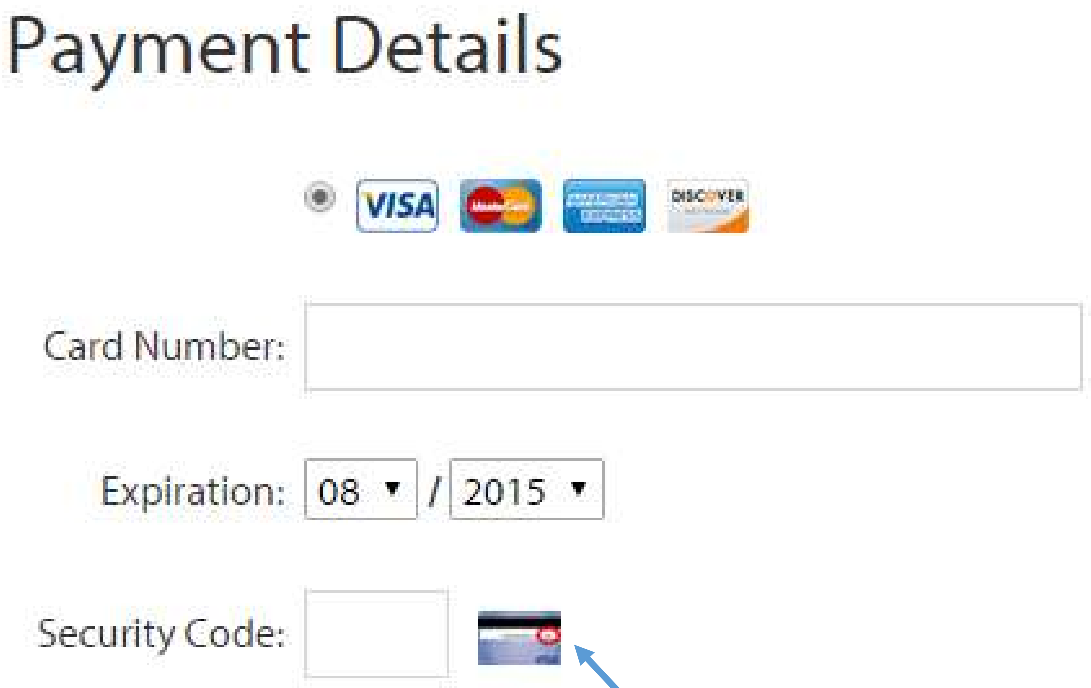
Phone: ________ *

Sex: ● Male ○ Femail

Age: ________

Hint required input

Set default value



Payment Details

● VISA MasterCard AMERICAN EXPRESS DISCOVER

Card Number:	
Expiration:	08 ▼ / 2015 ▼
Security Code:	

Hint about security code

How to validate

- Checking input required
  - Text input:
    - Using method trim() of String object to remove whitespace.

```
String id= request.getParameter("id").trim();
if("".equals(id)){
    //do something
}
```

  - Checkbox or List

```
String []pd=request.getParameterValues("product");
if(pd==null||pd.length==0){
    //do something
}
```

- Validating number:
  - Using boundary class to parse String to integer or double.

```
String ms=“”;
String  age= request.getParameter(“age");
try{
    int n_age=Integer.parseInt(age);
}catch (NumberFormatException e){
    ms=“your age not accept";
}
```

Prevent numberic error

- Using input with type number.
- User only input number to field.
  - Ex:

```
   <input type="number" name=“age"/>
```

- Using list and java loop.

```
<tr>
    <td align="right"><strong>quantity:</strong></td>
    <td align="left">
    <select name="quantity">
    <% for(int i=1;i<31;i++){ %>
        <option value="1"><%=i %></option>
    <%} %>
    </select>
    </td>
  </tr>
```

Regular Expressions

- A **regular expression (RE)** is a kind of pattern that can be applied to text.
- A regular expression either matches the text (or part of the text), or it fails to match

.	Matches any character
\\d	Matches any digit 0-9
\\D	Matches any non-digit
\\w	Matches "word" character a-z, A-Z, 0-9
\\W	Matches any non-"word" character
\\s	Matches any "space" character (, tab, return)
\\S	Matches any non-"space" character

- Quantifiers give <u>number</u> of times a character must appear

*	Any number of times (including 0)
+	At least once
{*number*}	Exactly *number* times

- Examples:
  - Credit card number: **`\\d{16}`**
  - Phone number: **`\\d{3}-\\d{3}-\\d{4}`**
  - Email address: **`\\w+@\\w+(\.\\w+)*`**

Using RE

- Java syntax:
  - Create Pattern object from regular expression
  - Create Matcher object using matcher method of Pattern and the actual input to match with
  - Use matches method of the Matcher object to determine whether match exists

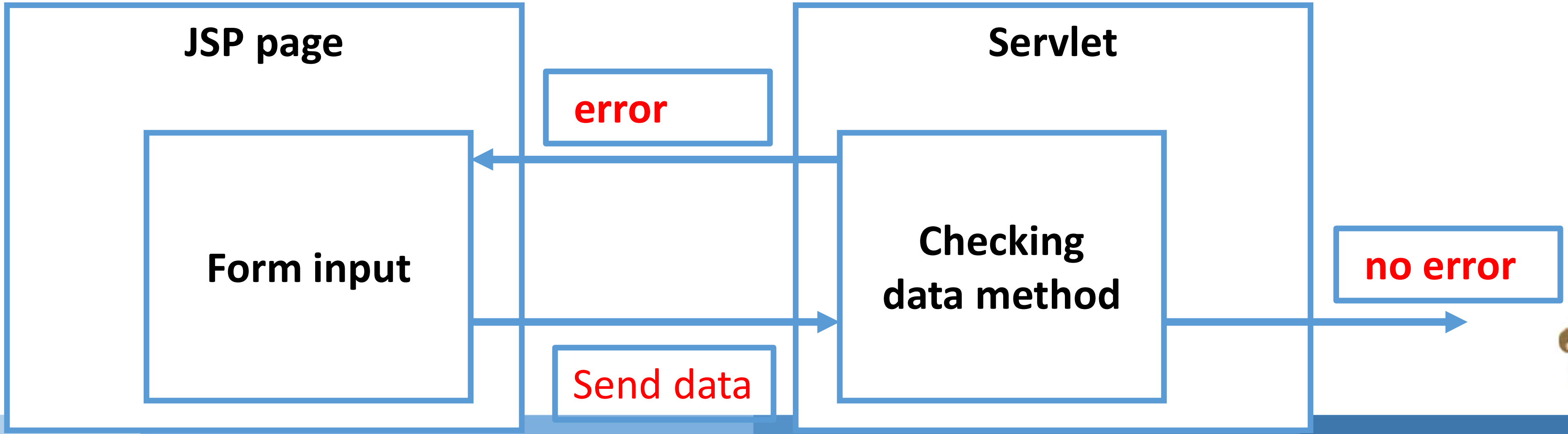
```
Pattern patternObject =
     Pattern.compile("regular expression");
Matcher matcherObject =
     patternObject.matcher(string to match with);
if (!matcherObject.matches()) {
      code to handle failed match
      }
```

Using RE

```
Pattern patternObject =
Pattern.compile("\\d{5}");
Matcher matcherObject =
patternObject.matcher("12344");
if (!matcherObject.matches()) {
        //do something
}else{
        //do something
}
```

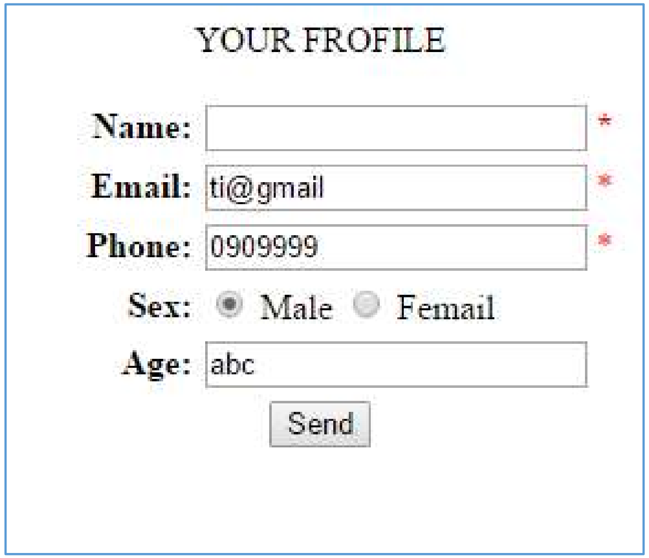
Display error

- When user input wrong data we need rollback the old page and display message error.
- Message error need easy understand.
- To send message to jsp page and keep old value you must use method forward of RequestDispatcher object.



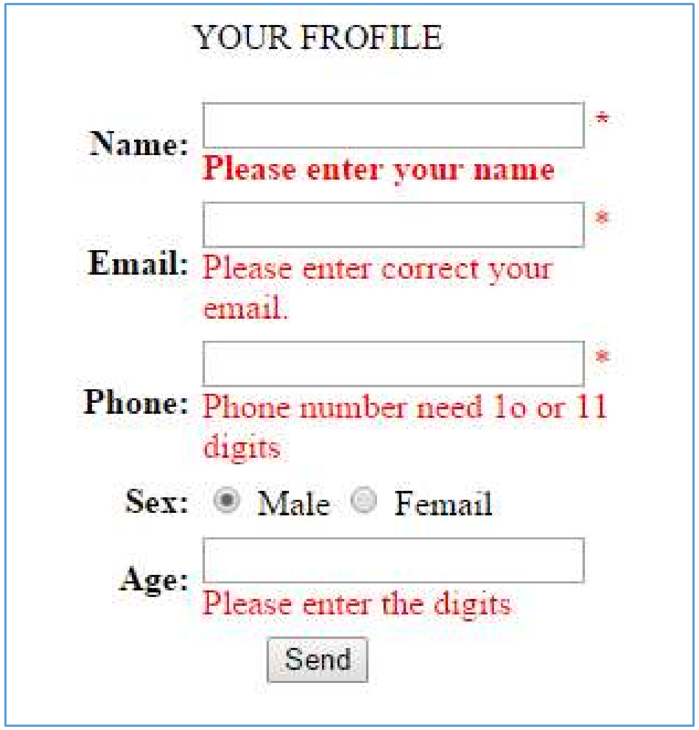
Display error page

- Echo back value user input
  - User only modify error field.
  - Save time for user.
- Display error message next to error field
  - Easy to see error.
  - Easy to understand error.
  - Easy to modify data.



YOUR FROFILE

Name:		*
Email:	ti@gmail	*
Phone:	0909999	*
Sex:	● Male ○ Femail	
Age:	abc	
	Send	



YOUR FROFILE

Name:	 **Please enter your name**	*
Email:	 Please enter correct your email.	*
Phone:	 Phone number need 1o or 11 digits	*
Sex:	● Male ○ Femail	
Age:	 Please enter the digits	
	Send	

Create error message

- Servlet use if condition to create detail message.

```
String url="view.jsp";
String quantity = request.getParameter("quantity").trim();
String quantityError = "";
if (quantity != null && !"".equals(quantity)) {
     try {
         int nQuantity = Integer.parseInt(quantity);
         if (nQuantity < 1 )
               quantityError = "Please enter digit larger than 0";
               url="NewFile.jsp";
    } catch (NumberFormatException e) {
          quantityError = "Please enter the digits";
          url="NewFile.jsp";
    }}
if(!"".equals(quantityError)){
          request.setAttribute("quantityerror", quantityError);
          RequestDispatcher rd= request.getRequestDispatcher(url);
          rd.forward(request, response);
}
```

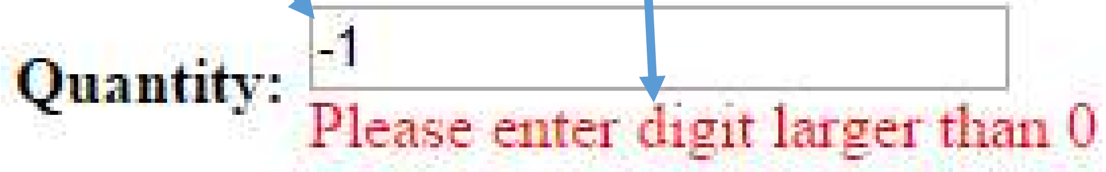
Display error message

- In jsp page, get value send back form servlet.
- Get error message from servlet
- Check error message: if error message equal null set new value to message.
- Display value of control and error message.

```
<% String errorAttributeValue =
(String)request.getAttribute("errorAttributeName");
   if (errorAttributeValue == null)
                        errorAttributeValue = ""; %>
…
<someInputField …> <%= errorAttributeValue>
```

- Display error message and old value:
  - Textfield:

```
<% String quantity=request.getParameter("quantity");
    if(quantity==null)quantity="";
    String quantityError=(String)request.getAttribute("quantityerror");
    if(quantityError==null)quantityError="";
    %>
   <input name="quantity" type="text" id="quantity"
value="<%= quantity%>">
   <span style="color:red"> <%= quantityError%></span>
```



Quantity: -1

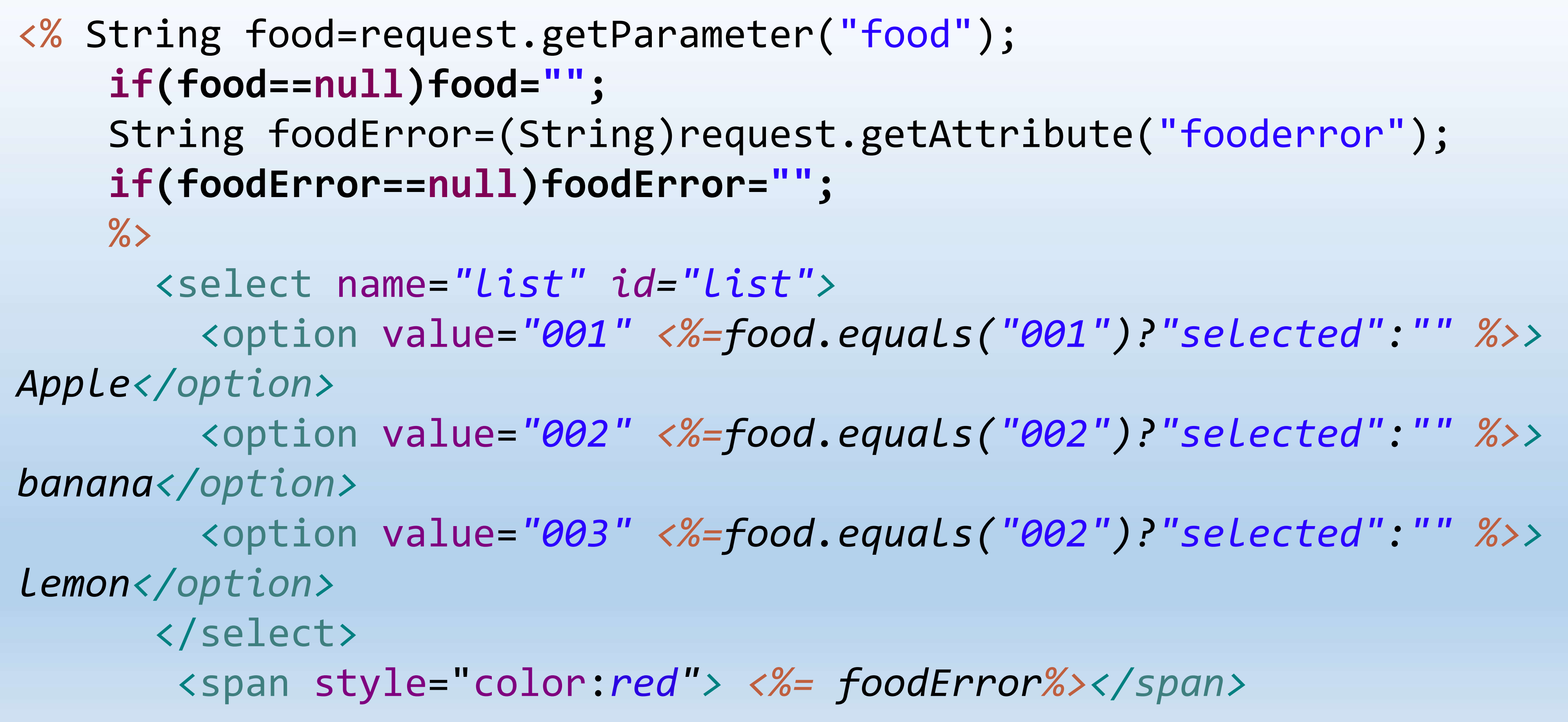
Please enter digit larger than 0

- Radiobutton

```
<% String sex=request.getParameter("sex");
    if(sex==null)sex="1";
    String sexError=(String)request.getAttribute("sexerror");
    if(sexError==null) sexError="";
    %>
       <input name="RadioGroup1" type="radio" id="RadioGroup1_0"
value="1" <%="1".equals(sex)?"checked":""  %> >
        Male
        <input type="radio" name="RadioGroup1" value="0"
id="RadioGroup1_1"  <%="0".equals(sex)?"checked":""  %>>
        Femail <br/>
        <%=sexError %>
```

- Checkbox: like radioButton

- List



```
<% String food=request.getParameter("food");
    if(food==null)food="";
    String foodError=(String)request.getAttribute("fooderror");
    if(foodError==null)foodError="";
    %>
      <select name="list" id="list">
        <option value="001" <%=food.equals("001")?"selected":"" %>>
Apple</option>
        <option value="002" <%=food.equals("002")?"selected":"" %>>
banana</option>
        <option value="003" <%=food.equals("002")?"selected":"" %>>
lemon</option>
      </select>
       <span style="color:red"> <%= foodError%></span>
```

Q & A