

HTML5 Friendly Markup

HTML5 Friendly Markup

- HTML5 offers new elements and attributes that can make it unnecessary to write your own components. It also provides many new capabilities for existing components.
- JavaServer Faces technology supports HTML5 not by introducing new UI components that imitate HTML5 ones, but by allowing you to use HTML5 markup directly.

HTML5 Friendly Markup

- JavaServer Faces technology support for HTML5 falls into two categories:
 - Pass-through elements
 - Pass-through attributes
- You can mix and match JavaServer Faces and HTML5 components and elements as you see fit.

Using Pass-through Elements

- Pass-through elements allow you to use HTML5 tags and attributes but to treat them as equivalent to JavaServer Faces components, associated with a server-side UIComponent instance.
- To make a non-JavaServer Faces element a pass-through element, specify at least one of its attributes using the `http://xmlns.jcp.org/jsf` namespace. See next example

Using Pass-through Elements

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml" [
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:jsf="http://xmlns.jcp.org/jsf"
      xmlns:f="http://xmlns.jcp.org/jsf/core">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <input type="range" jsf:id="range" name="range" value="#{test.value}">
  </h:body>
</html>
```

Using Pass-through Attributes

- If you specify a pass-through attribute in a JavaServer Faces UIComponent, the attribute name and value are passed straight through to the browser without being interpreted.
- There are several ways to specify pass-through attributes.

Using Pass-through Attributes

- Use the JavaServer Faces *namespace* for *pass-through attributes* to prefix the attribute names within a JavaServer Faces component.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:f="http://xmlns.jcp.org/jsf/core"
      xmlns:p="http://xmlns.jcp.org/jsf/passthrough">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <h:inputText id="test" p:type="range" value="#{test.value}"
                  p:min="0" p:max="100" p:title="Slidet this">
    </h:inputText>
  </h:body>
</html>
```

Using Pass-through Attributes

- To pass a single attribute, nest the `f:passThroughAttribute` tag within a component tag.

```
<h:body>
  <h:inputText value="#{test.value}">
    <f:passThroughAttribute name="type" value="range">
    </f:passThroughAttribute>
  </h:inputText>
</h:body>
```


Using Pass-through Attributes

- To pass a group of attributes, nest the `f:passThroughAttributes` tag within a component tag, specifying an EL value that must evaluate to a `Map<String, Object>`. For example:

```
<h:inputText value="#{bean.nights}">
  <f:passThroughAttributes value="#{bean.nameValuePairs}" />
</h:inputText>
```

```
private Map<String, Object> nameValuePairs;
...
public Bean() {
    this.nameValuePairs = new HashMap<>();
    this.nameValuePairs.put("type", "number");
    this.nameValuePairs.put("min", "1");
    this.nameValuePairs.put("max", "30");
    this.nameValuePairs.put("required", "required");
    this.nameValuePairs.put("title",
        "Enter a number between 1 and 4 inclusive.");
}
```