# Aplicaciones RIA con ZK

- Sesión 3: El lenguaje zscript



©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# El lenguaje zscript

- El lenguaje zscript
- Expresiones EL
- Atributos ZK
- Espacio de identificadores
- Procesamiento y carga de una página ZUL


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# El lenguaje zscript

- Distinto de JavaScript; se ejecuta en el servidor
- Es “Java interpretado”: un lenguaje de scripting interpretado por la librería BeanShell
- ¿Cómo ejecutar los scripts?
  - Dentro de etiquetas <zscript> situadas en cualquier parte de la página
  - En los manejadores de eventos
  - En la instrucción ?init de inicialización de la página


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Etiquetas zscript

- Se pueden incluir en cualquier parte de la página
- Se evalúan cuando el cargador de ZK (ZK loader) llega a esa instrucción al procesar la página
- Puede haber más de una, en distintos lugares de la página; cada declaración genera una invocación distinta de BeanShell

```
<zscript>
   Date now = new Date();
   abc ="def";
</zscript>
${now}
${abc}
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Código zscript en manejadores de eventos

```
<button label="Haz click"
   onClick='alert("Click en botón.")'/>
<label value="Mi etiqueta"
   onClick='alert("Click en etiqueta.")'
   onRightClick='alert("Botón derecho")'/>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Código zscript en la inicialización

```
<?init zscript="/my/init.zs"?>
```

/my/init.zs:

```
import java.util.Date;

miCadena = “Hola mundo”;
Date ahora = new Date();
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Importando paquetes en zscript

```
<zscript>
import java.sql.*;

void addItem() {
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   String url = "jdbc:odbc:Fred";
   Connection conn = DriverManager.getConnection(
                              url,"myLogin", "myPassword");
   ...
   conn.close();
}
</zscript>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Accediendo a componentes por su id

```
<zscript><![CDATA[
   import dao.PersonaDao;
   import model.Persona;
   void addItem() {
      PersonaDao personaDao = new PersonaDao();
      Persona persona = new Persona();
      persona.setNombre(nombre.getValue());
      persona.setApellidos(apellidos.getValue());
      personaDao.add(persona);
   }
]]></zscript>
```

- Escribimos una página en la que hayan dos campos de texto (textbox) con esos nombres.


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Ámbito de las variables

- Es muy importante tener en cuenta el ámbito de las variables a la hora de acceder a ellas
- El ámbito de una variable es al espacio de identificadores en el que se declara
- Hay un espacio de identificadores por cada ventana

```
<window id="A">
    <zscript>var1 = "abc";</zscript>
    <window id="B">
        <zscript>var2 = "def";</zscript>
    </window>
</window>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Variables locales

- Es posible definir variables locales que no se guardan en el espacio de identificadores
- Al declarar la variable “var1” no es visible en la ventana:

```
<window>
   <zscript>
   {
      int var1 = 12;
      var2 ="abc";
   }
   </zscript>
   var1: ${var1}
   var2: ${var2}
</window>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Otros lenguajes de script

- JavaScript, por medio del intérprete Rhino
- Ruby, por medio del intérprete JRuby
- Groovy, por medio del intérprete Groovy

```
<?page zscriptLanguage="Groovy"?>

<zscript>
   def name = "Hello World!";
</zscript>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Un ejemplo final

```
<window title="Ejemplo ZK" border="normal" width="200px">
   ¿Qué gestor de disposición es el que utilizas más?
   <hbox>
      <checkbox id="l1" label="Border" onCheck="doChecked()" />
      <checkbox id="l2" label="Box" onCheck="doChecked()" />
      <checkbox id="l3" label="Table" onCheck="doChecked()" />
      <checkbox id="l4" label="Portal" onCheck="doChecked()" />
      <checkbox id="l5" label="Column" onCheck="doChecked()" />
   </hbox>
   <hbox>
      Has seleccionado :
      <label id="layout" />
   </hbox>
   <zscript>
      void doChecked() {
         layout.value = (l1.isChecked() ? l1.label+' ' : &quot;&quot;)
                      + (l2.isChecked() ? l2.label+' ' : &quot;&quot;)
                      + (l3.isChecked() ? l3.label+' ' : &quot;&quot;)
                      + (l4.isChecked() ? l4.label+' ' : &quot;&quot;)
                      + (l5.isChecked() ? l5.label+' ' : &quot;&quot;);
      }
   </zscript>
</window>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Expresiones EL

- *${expresión}*
- Se utilizan para dar valor a atributos de los componentes
- Las expresiones más utilizadas son las que obtienen propiedades de objetos, accediendo a sus métodos *getters*:

*${objeto.propiedad}*


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Ejemplo

- Creamos un objeto persona y lo inicializamos en zscript y lo mostramos en la página utilizando expresiones EL


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Es posible acceder a propiedades de otros componentes

```
<window>
   <textbox id="source" value="ABC"/>
   <label value="${source.value}"/>
</window>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# No es correcto combinar zscript y EL

```
<window>
   <zscript>
   public void doAlert(String texto){
      alert(texto);
   }
   </zscript>
   <textbox id="aviso"/>
   <button label="¡Di texto!" onClick="doAlert(${aviso.value})" />
</window>
```

**Incorrecto**

```
<window>
   <zscript>
   public void doAlert(String texto){
      alert(texto);
   }
   </zscript>
   <textbox id="aviso"/>
   <button label="¡Di texto!" onClick="doAlert(aviso.getValue())" />
</window>
```

**Correcto**


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Atributos ZK

- if y unless

```
<window border="normal">
   <zscript>
      newBtn = true;
   </zscript>
   <button label="Nuevo" if="${newBtn}">
      <attribute name="onClick">
         alert("I am a new Button!");
      </attribute>
   </button>
   <button label="Viejo" unless="${newBtn}"/>
</window>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Atributos ZK

- each y foreach

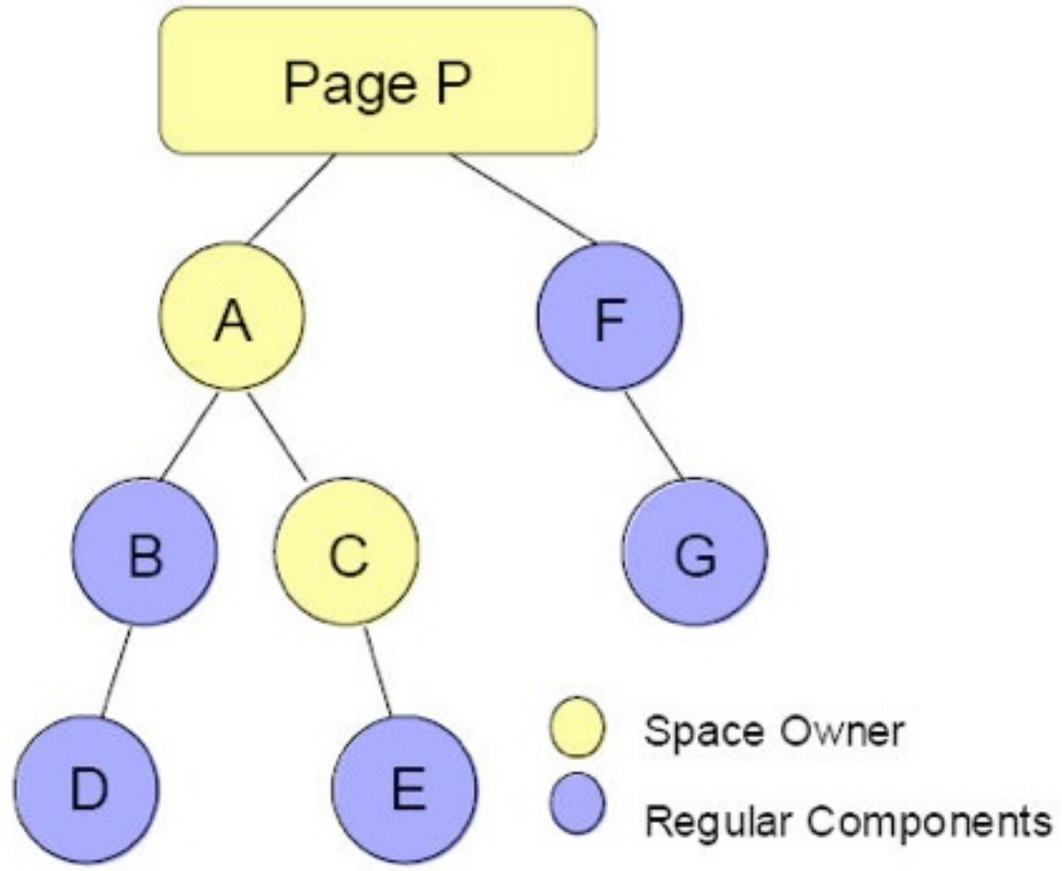
```
<window border="normal">
   <zscript>
      dias = new String[] {"Lunes", "Martes","Miércoles"};
   </zscript>
   <listbox width="100px">
      <listitem label="${each}" forEach="${dias}" />
   </listbox>
</window>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Espacio de identificadores

- ID Space: ámbito asociado a páginas y ventanas en el que residen los identificadores (variables de zscript e identificadores de componentes)

```
<?page id="P"?>
<zk>
   <window id="A">
      <hbox id="B">
         <button id="D" />
      </hbox>
      <window id="C">
         <button id="E" />
      </window>
   </window>
   <hbox id="F">
      <button id="G" />
   </hbox>
</zk>
```





©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Accediendo a un componente por su identificador

- Con el método getFellow(String) sobre cualquier componente
- Con el método Path.getComponent(String)

```
import org.zkoss.zk.ui.Path;
import org.zkoss.zul.Textbox;
...
Textbox nombre = (Textbox) Path.getComponent("/win1/nombre");
Textbox apellidos = (Textbox) Path.getComponent("/win1/apellidos");
persona.setNombre(nombre.getValue());
persona.setApellidos(apellidos.getValue());
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

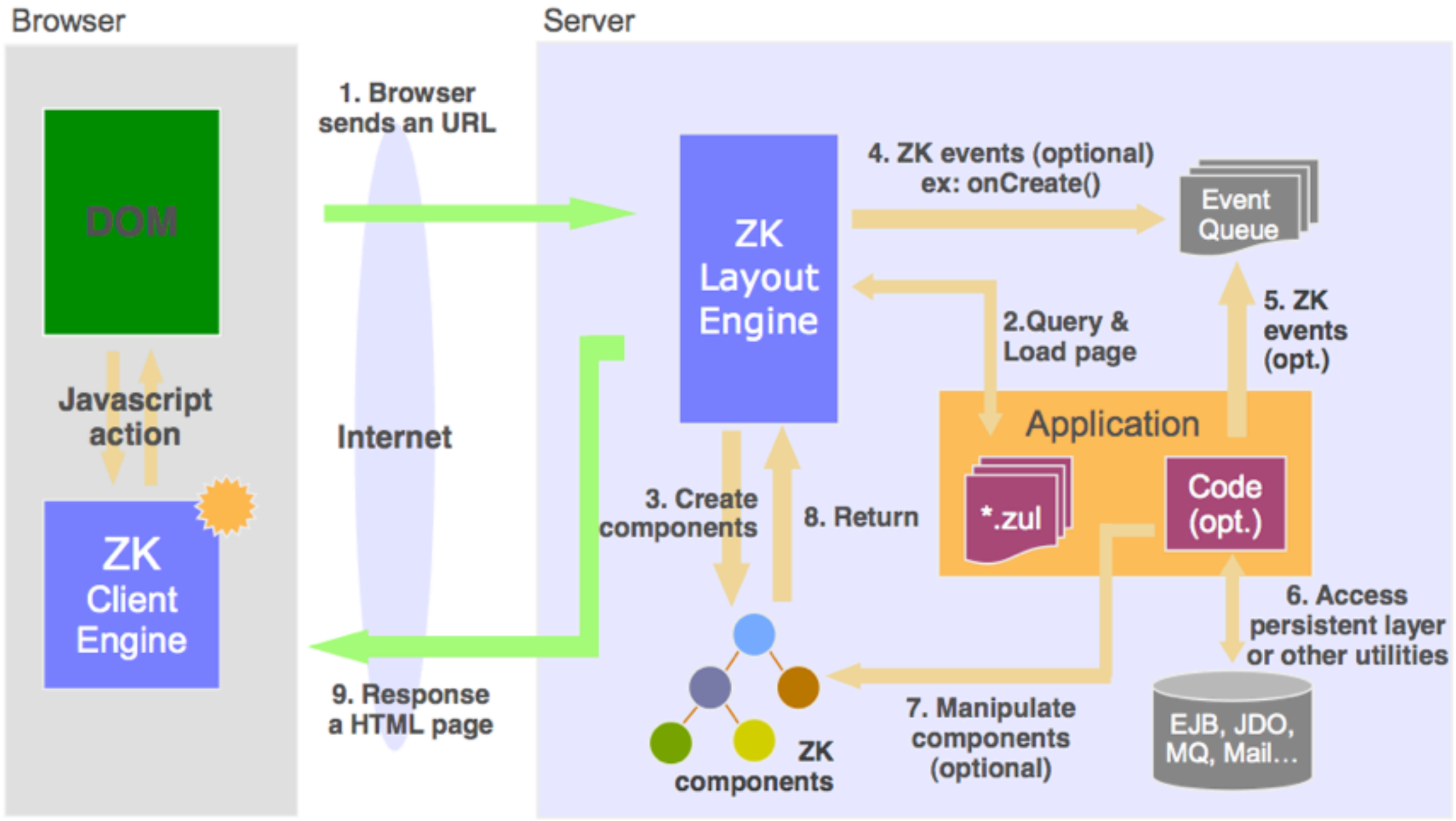
# Ámbito de identificadores de componentes

- Las identificadores de componentes son visibles en el espacio de identificadores en el que están definidos

```
<window id="w1">
   <window id="w2">
      <checkbox id="c"/>
      <button label="click me" onClick='c.setLabel("w2")'/>
   </window>
   <checkbox id="c"/>
   <button label="click me" onClick='c.setLabel("w1")' />
</window>
```


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Ciclo de vida de una página ZUL (1)





©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante

# Ciclo de vida de una página ZUL (2)

1. Fase de inicialización de la página
2. Fase de creación de componentes
3. Fase de procesado de eventos
4. Fase de renderizado


©2010 Depto. Ciencia de la Computación e IA
Universidad de Alicante