CS3220 Web and Internet Programming
Expression Language (EL)

Chengyu Sun
California State University, Los Angeles

What is EL?

- Expression Language (EL)
  - Syntax: ${ expression }
  - Since the JSP 2.0 Specification
  - A concise way to access data in JSP

Elements of a Programming Language

- Comments
- Literals
- Variables and Types
- Operators
- Expressions: anything that evaluates to a single value
- Statements
- Functions
- Classes
- Packages

Comments

- **<%--** JSP (Hidden) Comments **--%>**
- **<!--** HTML Comments **-->**

Expression

- Literals
- Variables
- Operators

EL Literals

- `true`, `false`
- `23`, `0x10`, …
- `7.5`, `1.1e13`, …
- `"double-quoted"`, `'single-quoted'`
- `null`

- No char type

EL Variables

- You cannot declare new variables using EL (after all, it's called *"expression"* language).
- However, you can access implicit objects, scoped variables (i.e. the objects saved in application, session, and request scopes), and their properties

What Are Properties?

```
Class User {

    private String firstName;
    private String lastName;

    public User() {}

    public String getFirstName()
    {  return firstName; }

    public String getLastName()
    {  return lastName; }

    public String getName()
    {  return firstName + " " + lastName; }
}
```

Fields

Constructor

Methods

Properties Are Defined by Getters and/or Setters

```
Class User {

    private String firstName;
    private String lastName;

    public User() {}

    public String getFirstName()
    {  return firstName; }

    public String getLastName()
    {  return lastName; }

    public String getName()
    {  return firstName + " " + lastName; }
}
```

Properties

`firstName`

`lastName`

`name`

About Properties

- Property naming conventions
  - 1st letter should be in lower case
  - 1st letter should be capitalized in getter (accessor) and/or setter (mutator)
- Property types
  - *read-only* property: only getter
  - *write-only* property: only setter
  - *read-write* property: both

Property Example

- What properties does `Foobar` have?
  - Read-only: ??
  - Write-only: ??
  - Read-write: ??

```
public class Foobar {

    private int a, b, c, d;
    private boolean e;

    public Foobar() { a = b = c = d = 0; }

    public int getA() { return a; }
    public void setA( int a ) { this.a = a; }

    public int getB() { return b; }

    public void setC( int c ) { this.c = c; }

    public int getAb() { return a+b; }

    public boolean isE() { return e; }
    public void setE( boolean e ) { this.e = e; }
}
```

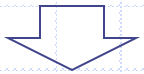
Common Problems with Property …

```
public class Foobar {

    private int a, b, c, d;

    public Foobar() { a = b = c = d = 0; }

    public int getA() { return a; }
    public void setA( String s ) { this.a = Integer.parseInt(s); }

    public int getB( int x ) { return b+x; }

    public void setC( int c, int x ) { this.c = c+x; }

    public void setD( String s ) { this.d = Integer.parseInt(s); }

}
```

How many properties does Foobar have??

… Common Problems with Property

- A getter must have no argument
- A setter must have exactly one argument
- The *type* of a property must be consistent in both the getter and the setter



It's easier and safer to let Eclipse generate getters and setters instead of writing them yourself

Access Properties Using EL

```
${obj_name.property_name}
```

- Class `A`
  - `int` property `id`
  - `String` property `name`
  - `String[]` property `weekdays`
  - `List<Double>` property `numbers`
- Class `B`
  - `A` property `a0`
  - `List<A>` property `listA`

About Accessing Properties with EL

- A property may be an object and have its owner properties, e.g.

```
${b.a0.id}
```

- We can use index to access objects in a List or an array

```
${a.weekdays[1]}
```

Implicit Objects in EL

- pageContext
  - servletContext
  - session
  - request
  - response
- param, paramValues
- header,headerValues
- cookie
- initParam

- pageScope
- requestScope
- sessionScope
- applicationScope

Example: RequestInfo

- Display the request method and client address
- Display objects in request and session scope
- Display the value of a request parameter
- Display the value of a cookie

About Using Implicit Objects in EL

- Find the Java API for the object and look for its properties, e.g.

```
${pageContext.request.remoteAddr}
```

- Access elements in a collection using both `[]` and `.` syntax, e.g.

`${param.a}` and `${param["a"]}`

EL Operators

- Arithmetic
  - +, -, *, /, %
  - div, mod
- Logical
  - &&, ||, !
  - and, or, not
- Relational
  - ==, !=, <, >, <=, >=
  - eq, ne, lt, gt, le, ge
- Conditional
  - ? :
- empty
  - check whether a value is null or empty
- Other
  - [], ., ()

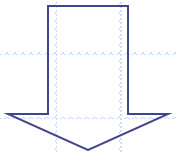
EL Evaluation and Auto Type Conversion

${2+4/2}	
${2+3/2}	
${"2"+3/2}	
${"2"+3 div 2}	
${"a" + 3 div 2}	
${null == 'test'}	
${null eq 'null'}	

${empty ""}	
${empty null}	
${empty "null"}	
${"abc" lt 'b'}	
${"cs3220" > "cs2013"}	

Limitation of EL

- Only expressions, no statements, especially *no control-flow statements*



JSTL