20_EXPRESSION_LANGUAGE_[EL]→

=) Expresion language is a substitutes for the Expressions in the scripting.

<%= 12 %> ⟹ ${12}

★ What we can write in EL → We can write any contant (string, integer, boolean etc) and attributes in the EL.

=) The attribute can be ~~to~~ belonged to any scope; i.e. page, request, session, application.

${ }
↑ constant

${ }
↑ attribute name

<implicit-objects>.setAttribute("marks", 987);

${marks}
↳ will print 987

${'omji'}
${"omji"}
} → This will both treated as string.

${3.14f} → We can't use suffix with contants in EL.

<% int marks = 789; %>
<%=marks %> ⟶ this will print 789, but

~~$t~~ ${marks} ⟶ mark will be treated as an attribute saved in any attribute scope, and if no such attribute, then nothing will be displayed.

x x x

* Attribute Scopes in ⟶

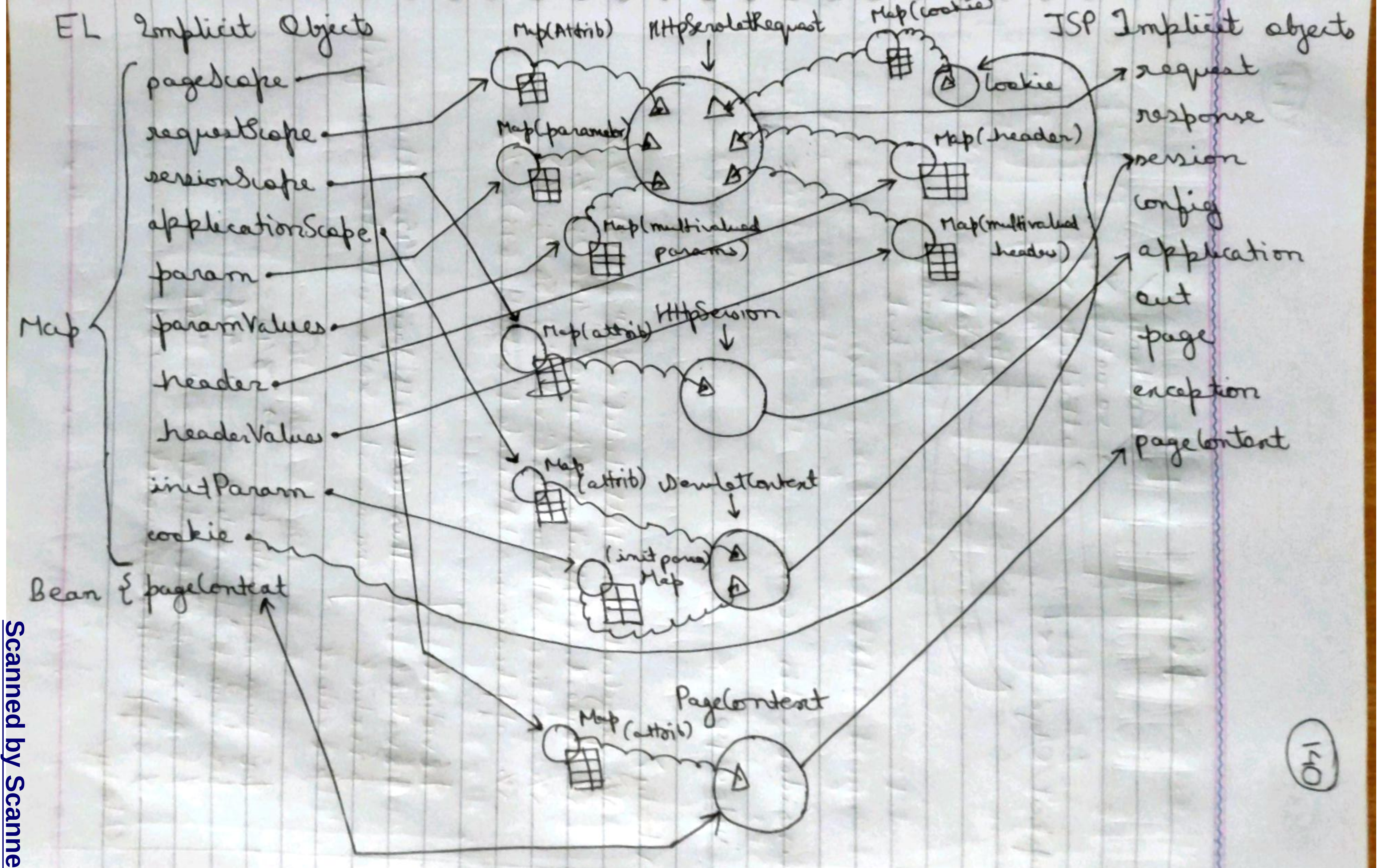
Servlet	JSP	API
Request Scope	Request Scope	HttpServletRequest
Session Scope	Session Scope	HttpSession
Application Scope	Application Scope	ServletContext
	Page Scope	PageContext

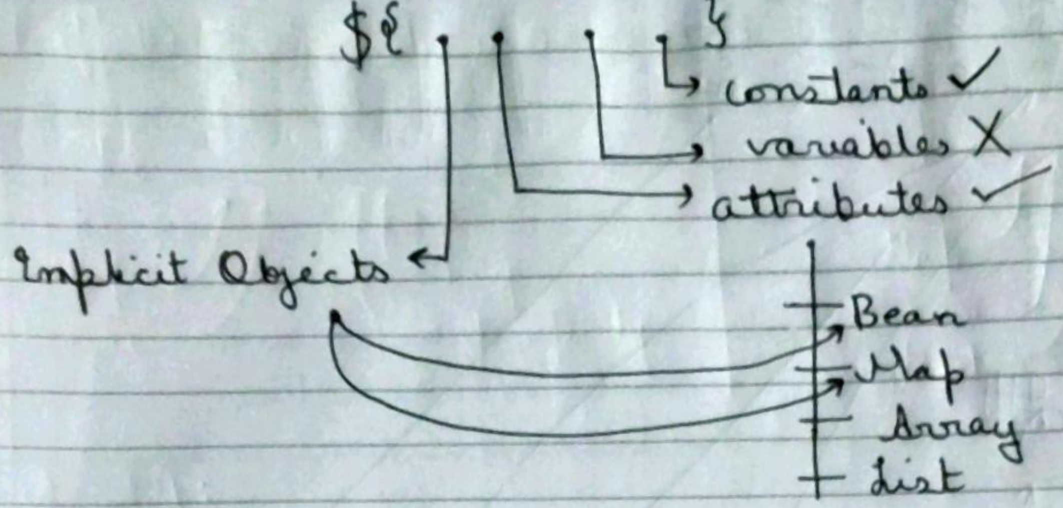
x x x

* Different Implicit Objects ⟶ There are different implicit objects in the EL and in the JSP.

=> There are 11 implicit objects in EL and 8 ~~empt~~ implicit objects in JSP.

(PTO)


EL Implicit Objects
Map
pageScope
requestScope
sessionScope
applicationScope
param
paramValues
header
headerValues
initParam
cookie
Bean
pageContext
Map(Attrib)
HttpServletRequest
Map(cookie)
cookie
Map(parameter)
Map(header)
Map(multivalued params)
Map(multivalued header)
Map(attrib)
HttpSession
Map (attrib)
ServletContext
(init param) Map
PageContext
Map (attrib)
JSP Implicit objects
request
response
session
config
application
out
page
exception
pageContent



★ Consider this case ⟶

index.jsp

```
<body>
    <% pageContext.setAttribute("value", 345);  %>
    <% request.setAttributes("value", 456 );  %>
    <% session.setAttributes ("value", 567);  %>
    <% application.setAttribute ("value", 678);  %>

     ${value}    }
     ${value}    }  This will all print 345, the
     ${value}    }  narrowest Scope will be seen first
     ${value}    }

     ${pageScope.value}  ⟶  345
     ${requestScope.value}  ⟶  456
     ${sessionScope.value}  ⟶  567
     ${applicationScope.value}  ⟶  678
</body>
```

★ Consider these two cases →

Servlet

```
User user = new User();
user.setName("raja");
user.setAge(7);
request.setAttribute("user",User);
```

↓ forward

Model class (User)

```
class User {
    private String name;
    private Integer age;

    { -- " --
      -- " --
```

getter & setter acc to the convention

JSP

```
${user}  → User@[redacted]79af7

${user.name}
```

Bean stored as attribute

property of Bean

Servlet

```
Map map = new HashMap();
map.put("aaa", 111)
    :
request.setAttribute("records", map);
```

JSP

```
${records.aaa}
```

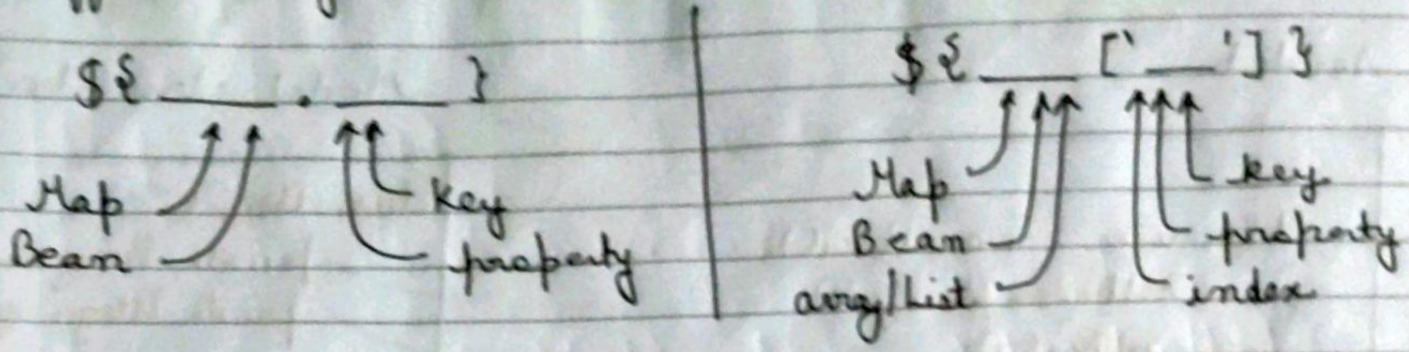
map stored in attribute

key in map

```
${records}
```

This will return map (key-value) pair

★ Different way to use EL ⟶



Servlet:

```
request.setAttribute("#3r", 12);
```

↳ here the key is not follwing the java identifier rules

JSP:

```
${#3r}  X
```

~~$~~

Servlet:

```
int[] x = {12,17,19};
request.setAttribute("recs", x);
```

JSP:

```
${recs[0]}
${recs[1]}
```

Servlet:

```
Arraylist x = new Arraylist();
x.add("ram");
x.add("om");
request.setAttribute("recs", x);
```

JSP:

```
${recs[0]}
${recs[1]}
```

★ Implicit objects param and paramValues ⟶

=) The implicit object param resembles the map that have name of the input field as the key and value passed in the input field as the value of the map.

=) The implicit object paramValues resembles the map that have name of the multiple type input field as key and the array of the values of input field as the value of the map.

index.jsp

```
<body>
    <form action="next.jsp">
            <input type="email" name="email">
            <— " —— " — = "text" name="company">
            <— " —— " —= "number" name="salary">

            <input type="checkbox" name="food" value="Samo">
            —— " ——— " ——— " ——— " —— " ——= "jabali">
            —— " ——— " ——— " ——— " —— " ——= "poha">

          <select name="city" multiple>
                  <option> Jabalpur </option>
                  <— " —> Jaipur <— " —>
                  <— " —> Nagpur <— " —>
                  <— " —> Raipur <— " —>
          </select>
</body>
```

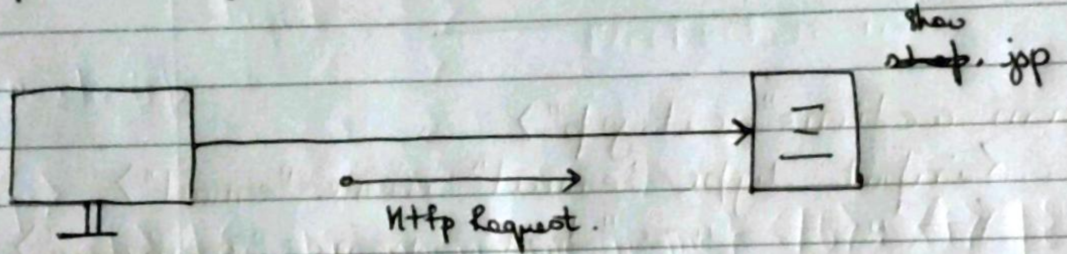
next.jsp

```
${param.email}
${param.company}
${param.salary}

${paramValues.food[0]} , ${paramValues.food[1]}

${paramValues.city[0]}, ${paramValues.city[1]}
```

★ Implicit object header and headerValues →



Http Request

```
GET   act.do ?age =12       Http/1.1

Host: abc.com
user-agent: chrome
accept : HTML, image , audio . . .
accept-language : en-US
acapt-encoding : gzip , deflate, jake
keep - alive : True
```

} Request Header.

:) And we can get request Headers as →

show.jsp

```
${header}
${ headerValues}

${header.host}

${header.user-agent}
```

X because it's # against the rules of identifier

```
${header['user-agent']
```

★ Implicit object initParam →

web.xml

```
<context-param>
     <param-name> count </param-name>
     <param-value>  34 </param-value>
</context-param>
```

index.jsp

```
${ initParam.count}
```

★ Implicit object cookie →

index.jsp

```
<%
     Cookie c1 = new Cookie("username", "ganesh");
     Cookie c2 = new Cookie("age", "23");

     response.addCookie(c1);
     response.addCookie(c2);
%>

<a href="next.jsp">Next Page </a>
```

next.jsp

```
<p>
    <b> ${cookie.username.name} </b> : <i> ${cookie.name.value} </i>
    <b> ${cookie.age.name} </b> : <i> ${cookie.age.value} </i>
```

★ Implicit object pageContext → This implicit object refers to the class (bean) PageContext which contains object of all the other implicit objects.

(PTO)

index.jsp

```
${pageContext}

${pageContext.request}
${pageContext.response}
${pageContext.session}
${pageContext.servletConfig}
${pageContext.servletContext}
${pageContext.out}
${pageContext.page}
```