PSP

Just as there are html tags, there are psp tags. There are several tags that define the type of command contained within a block.

We have PSP Header Tags. These are found within the **<%@ %>** block.
Eg A Typical header of a master_details.psp

<%@ page language="PL/SQL" %>
<%@ plsql procedure="app_Position" %>

These header tags are ALWAYS found at the top of a PSP document.
They describe the language used, and the name of the procedure that this document will create through the **loadpsp** program.

Other Tags include the insertion tag **<%= %>**, the declaration of variables tag **<%! %>** and the ordinary tag that simply tells the compiler to perform some DATABASE Related function, like a select statement etc **<% %>**.

When a procedure has been called from another HLTML page, they may pass variables from forms or just from hyperlinks. These variables are set in a master_details.psp document like:

A HTML HYPERLINK

<a href="http://dragon.uow.edu.au:777
**/pls/csci8/PL_SQL_PROCEDURE_NAME?VARIABLE_NAME=<%= DATABASE_FIELD %>"> this is a link **

As we can see above, there is a normal html hyper-link to an address, that just happens to be a database server address. After the basic address we have the name of the procedure that we want to call. A procedure is named through it's header tags (see above). After we see the name of the procedure, in this case we also see some extra HTML (the '?VARIABLE_NAME=' which is what a link looks like If called from a submit form. This is saying that it is sending a variable across to the server. The contents of the variable are the results of an insertion from a database variable.

Obviously this can also done through an actual form. Variables will be sent over the same way as in the HTML Hyperlink case, except the HTML browser sets up the format of the variables in the link after the submit button is clicked. A form might look like this:

HTML FORMS

<FORM METHOD=POST
ACTION="http://dragon.uow.edu.au:7777/plsql/print_positions">

<INPUT NAME="app_no" TYPE=text >

<INPUT TYPE=submit id="execute">
</FORM>

The variables that the details.psp will have to work with are referred to by the same name that they are passed through the POST by, in this case it's name is 'app_no'.

To capture and use these variables on the details.psp we must include the parameter header tags. In the case above with the form, we would have:

<%@ plsql parameter="app_no" type="number" %>

Types can be **number**, **varchar**, special type **OWA_UTIL.IDENT_ARR**, ... etc

Creating oracle pl/sql variables is done through the <%! %> tag. We can either by explicitly declaring types:

<%! USERNAME VARCHAR(30); %>

or we can set the type as the same as used in a particular table. Eg

<%! USERNAME APPLICANT.NAME%TYPE; %>

INSERTIONS

If we want the VALUE of a SQL variable to appear on a web page (in whatever form) we need the insertions tag **<%= %>.**
This may be done simply like this:

<%= USERNAME %>

SQL FUNCTIONS

To perform all the other sql functions, like select, insert to table, update, cursors etc, we use the basic psp tag **<% %>**. Inside these tags we can place **any number of statements** to perform our database queries.

-- a select query --

```
<%
      select fname, lname
      into first_name, last_name
      from APPLICANT
      where APPLICANT.A#=app_no;

%>
```

-- a cursor --

```
<% FOR app_record IN (
                        select APPDATE, EMPLOYER, PTITLE,
                        from APPLICANT, POSITION, APPLIES
                        where APPLICANT.A# = app_no AND
                        APPDATE >= from_date
                        AND APPLICANT.A# = APPLIES.A#
                        AND APPLIES.P# = POSITION.P# )  LOOP %>

      /*    ALL CONTENTS WITH BETWEEN THE LOOP STATEMENT, AND THE END LOOP
            STATEMENT, are looped until there are no more records. If there
            Is only ONE record, there is only ONE loop.     */


// we can now print out these records through insertion statements

<br><%= app_record.APPDATE %>
<br><%= app_record.PTITLE %>

<% END LOOP; %>
```

QUICK NOTE ON INSERTING ITEMS IN PLSQL --- where in sql, if inserting a text field we must enclose it with single quotes, in the case of plsql, we do not need to do that if the value is a variable. Eg

```
<% insert into APPLICANT values ( last_name, first_name phone_num, 'NSW');
```

here we show inserting values from variables last_name, first_name and phone_num, which are varchar, varchar, number repectively, and then we insert what would be another varchar, but it is hard-coded. In this case only we have single quotes around the value (because it isn't a variable).

-- LISTS --

If we use a checkbox in our html master_details page, we have to do some extra work to use those values, as in a check box, we can check the boxes or one **or more items**. This means that we have a list of values all under the **same name**. To capture and use these values in the list we must use the plsql special type **OWA_UTIL.IDENT_ARR.**

As well as this, in the master_details page, it is required to set up the form so that when the variables are sent across to the sql server, we can discover which of those values in the checkbox list is the last in the list! We can do this through a little HTML trick called a hidden field. We do it by setting the hidden field value to a known value, and we have the name of the hidden field THE SAME as the name of each checkbox.

EXAMPLE

-- master details page --

```
<FORM METHOD=POST
      ACTION="http://dragon.uow.edu.au:7777/pslql/pls/csci8/new_results">

      <input type="checkbox" name="positions" value="1"> position 1 <br>
      <input type="checkbox" name="positions" value="2"> position 2 <br>
      <input type="checkbox" name="positions" value="3"> position 3 <br>

      <input type="hidden" name="positions" value="end">

</FORM>
```

now when we iterate through the list of items in the positions list, when we see the value "end" we know that we are at the end of the list and to disregard this last item.

-- server details page --

```
<%@ page language="PL/SQL" %>
<%@ plsql procedure="show_checkboxes" %>

<%@ plsql parameter="positions" type="OWA_UTIL.IDENT_ARR" %>

……
……

<% LOOP

      IF positions(column) ='end' then EXIT;

      else %>

            You clicked this --->  <% positions(column) %> <br>

      ENDIF;

      <% Column := column + 1;

END LOOP %>
```