Since Oracle 11g the Developer can use reserved words etc as identifiers for local variables, by using double quotes around the variable, for example …

```
declare

   "begin" varchar2(1000);

begin

   "begin" := 'Hello';
   dbms_output.put_line('The value of begin is '||"begin");

end;
```

Note that the variable name now becomes 'case sensitive', that is the declared name must be used exactly as declared in the same case…

```
declare

   "begin" varchar2(1000);

begin

 dbms_output.put_line('The value of begin is '||"Begin");
                                                *
ERROR at line 9:
PLS-00201: identifier 'Begin' must be declared
```

Double quotes will allow variables to have spaces and symbols …

```
declare

   "begin the begin $" varchar2(1000);

begin

   "begin the begin $" := 'Hello';
   dbms_output.put_line('The value of Begin is '||"begin the begin $");

end;
```

PL/SQL Structure - Quoted Identifier Variables

The following are examples of what will now compile in PLSQL (but are not necessarily recommended) …

```
declare

   "www.address" varchar2(1000);
   "The end" varchar2(1000);
   "declare" varchar2(1000);
   "start,end" varchar2(1000);
   "form-start" varchar2(1000);

begin
```

There are a greater number of local variable datatypes available in PLSQL than in SQL, this part of the section examines some datatypes which are different from the SQL datatypes and examines their usefulness.

Firstly, examining the varchar2 datatype, its maximum length is 32767 bytes, in addition, it has synonyms of *string* and *varchar* (not recommended).

All of these declarations are legal …

```
l_var    varchar2(1);
l_var1   varchar(32767);
l_var2   string(4001);
```