



PL/SQL funtion 만들기

function 사용 예제 -1

생성후 다른곳에서 사용

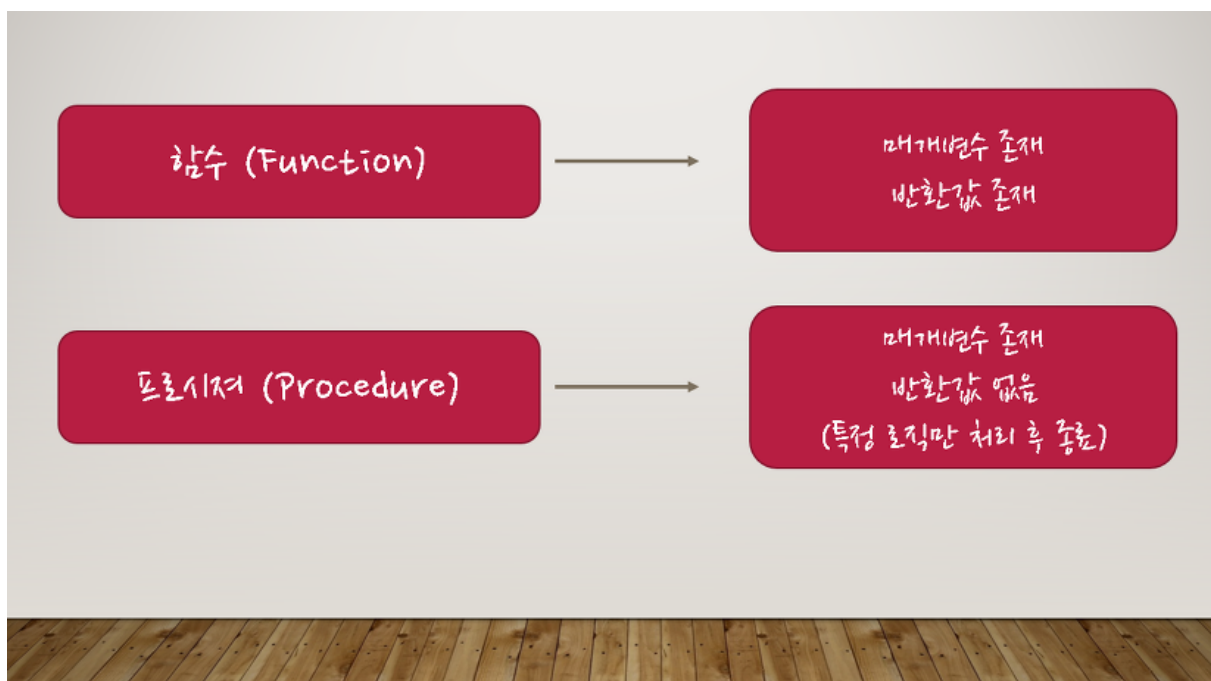
오버로딩 가능 및 함수에서 함수부르기 가능

전역변수를 저장하는 Vo처럼 사용가능

로컬시간 가져오는 함수

Function을 통한 package 도 선언이 있고 본문이 있다.

function 선언문 return 값을 명시한 예제



function 사용 예제 -1

```

create or replace function get_sal  --function을 만들다 (id넣으면
(p_emp_id number)
return number                    -- return 숫자값 반환지정
is
v_sal number;
begin
    select salary into v_sal
    from employees
    where employee_id=p_emp_id;

    return v_sal;                -- return 반환값 지정 가능

end;
-----
-----사용예제
declare
v_sal number;
begin
v_sal:=get_sal(100);
dbms_output.put_line (v_sal);
end;
-----사용예제
begin
dbms_output.put_line (get_sal(100));
end;

-----사용예제
variable b_salary number;
execute :b_salary:=get_sal(100)
print b_salary
-----사용예제
select get_sal(100) from dual;
--테이블 형태로 반환

-----사용예제
create or replace function get_sal
(p_emp_id number)

```

```

return number
is
v_sal number;
begin
    select salary into v_sal
    from employees
    where employee_id=p_emp_id;

    return v_sal;                                --함수이기 때문에 retrun 여러번 사
exception
when no_data_found then
    return -1;                                --함수이기 때문에 retrun 여러번 사

end;

```

생성후 다른곳에서 사용

```

create or replace function get_sal_tax
(p_sal number)
return number
is

begin

    if p_sal<5000 then
        return p_sal* (10/100);
    else
        return p_sal* (15/100);
    end if;

end;
-----

select employee_id, first_name, salary, get_sal_tax(salary)
from employees;

```

```
select employee_id, first_name, salary, get_sal_tax(p_sal=>sa
from employees;
```

```
select employee_id, first_name, salary, get_sal_tax(salary)
from employees
where get_sal_tax(salary)>2000
order by get_sal_tax(salary);
```

오버로딩 가능 및 함수에서 함수부르기 가능

```
----- 오버로딩!!!!-----
create or replace package area
is

function square_area( p_side number )
return number;

function rectangle_area( p_l number,p_w number )
return number;

--we dont have begin in package specification
end;

----- 함수에 함수 부르기 가능 -----
create or replace package body area
is
    function square_area( p_side number )
    return number
    is
    begin
    return p_side*p_side;
    end;

    function rectangle_area( p_l number,p_w number )
    return number
```

```

        is
        begin
        return p_l*p_w;
        end;

begin
DBMS_OUTPUT.PUT_LINE('welcome ');
end;

select area.square_area(4) from dual;
select area.rectangle_area(4,10) from dual;
begin
DBMS_OUTPUT.PUT_LINE(area.square_area(4));
end;

```

전역변수를 저장하는 Vo처럼 사용가능

```

--we can create package specification without body
--this used when we want to define global variable

create or replace package global_Measurement
is

    c_mile_to_km    constant number:=1.6093;    //마일을 km 변환
    c_kilo_to_mile  constant number:=0.6214;    // km 마일로 변환

end;

-- 해당 상수 값 확인
execute dbms_output.put_line('60 mile:='||60* global_Measurement.c_mile_to_km);
execute dbms_output.put_line('100 KM:='||100* global_Measurement.c_kilo_to_mile);

--마일 값을 입력하면 km로 변환해주는 함수 만듦
create or replace function get_mile_to_km
( p_value number )
return number
is
begin

```

```

        return p_value* global_Measurement.c_mile_to_km;

end;

--마일인자값 입
select get_mile_to_km(100) from dual;

```

로컬시간 가져오는 함수

```

declare
    function get_sysdate
    return date
    is
    begin
        return sysdate;
    end;

begin

    dbms_output.put_line(get_sysdate);

end;

-- 23/02/20

```

Function을 통한 package 도 선언이 있고 본문이 있다.

```

create or replace package p_test      //패기지중 선언문
is
c_var1 constant number:=10;
c_var2 varchar2(100):='welcome';

```

```

procedure print111;  //함수를 프로시저에 등록한다.
end;
-----
create or replace package body p_test      //패키지중 본문
is
c_var3 varchar2(100):='hi there';
procedure print111  //함수를 프로시저에 등록한다.
is
c_var4 varchar2(100):='hi';
begin
dbms_output.put_line('this variable came from package spec.
dbms_output.put_line('this variable came from package spec.
dbms_output.put_line('this variable came from package body.
  dbms_output.put_line('this variable came from print Proc. '
end;

end;

execute p_test.print111;

```

function 선언문 return 값을 명시를 통한 예제

-- 오버라이딩 예제

```

create or replace package override
is
-- 선언에서 함수 형식을 선언
function to_char( p1 number, p2 date )
return varchar2; --return 값 지정

procedure print;

end;

create or replace package body override
is

```

```

function to_char( p1 number, p2 date )
return varchar2  --반환값
is
begin
return p1||p2;  -- p1+p2 합해서 출력
end;

procedure print
is
begin
dbms_output.put_line(to_char(1,'23/01/01' )); -- 의도한 9
dbms_output.put_line(standard.to_char(10)); -- 반환값이 10
end;

end;

execute override.print

```