

## 함수<sup>Function</sup>들

처리한 결과를 알려줘야 하는 프로시저는 함수로 만들어서 활용합니다.

이제 함수를 살펴보자. 함수<sup>Function</sup>는 프로시저와 거의 같다. 차이점은 값을 반환<sup>return</sup>해 주는 문장이 하나 더 들어 있다는 것뿐이다.

텔파이 안에는 수많은 함수들이 미리 만들어져 있다. 하지만, 함수를 직접 만들어야 하는 상황도 많다. 예를 들어 아래의 예문을 보면  $Tg(A)$  라는 함수를 사용하여 문제를 풀도록 되어 있다. 하지만 텔파이 안에는  $Tg(A)$  라는 함수가 없다.

직각삼각형이 있다. 빗변이 아닌 변 하나의 길이와 이 변과 빗변 사이의 각도를 받아서, 빗변이 아닌 다른 한 변의 길이를 계산해보자.

버튼을 더블 클릭하여 코드 편집기를 연다.

```
procedure TfrmCatheti.btnRunClick(Sender: TObject);
var
    a, b, alfa: Real;
begin
    alfa:= StrToFloat(edtAlfa.Text); //alfa 각도를 읽는다.
    a:= StrToFloat(edtA.Text); //인접한 변 a의 길이를 읽는다.
    b:= a*Tg(alfa); //빗변이 아닌 또 다른 변 b의 길이를 계산한다.
    lblB.Caption:= FloatToStr(b); //b를 출력한다.
end;
```

우리는 우리의 함수를 구현<sup>implementation</sup> 구역 안에 적어 넣으면 된다. 함수를 선언할 때에는 이름, 파라미터의 개수와 타입, 반환 값의 타입을 명시해야 한다.

### 함수<sup>Function</sup> 선언 형식

```
function <함수 이름> (<파라미터 목록>): <반환return 값의 타입>;
    <지역local 선언declaration들>
begin
    <구문들>
end;
```

함수의 파라미터 규칙 또한 프로시저의 파라미터 규칙과 같다. 값<sup>value</sup> 파라미터와 변수<sup>variable</sup> 파라미터를 모두 사용할 수 있다.

함수는 값을 반환해야 하기 때문에 반드시 할당 연산자가 있게 된다. 반환할 값은 함수의 이름에 할당한다. 함수 이름 대신 *Result*라는 키워드를 사용해도 된다.

```
function Tg(x: Real): Real;
var
    y: Real;
begin
    y:= x/180*pi;
    Tg:= sin(y)/cos(y); //함수 이름(Tg) 대신 Result를 써도 된다.
end;
```

위 예문에 있는 새 변수 *y*는 함수 안에서 선언되었다. 이와 같이 함수 안에서 선언된 변수는 ‘지역<sup>local</sup> 변수’라고 한다. 지역변수는 선언된 지역(여기서는 함수) 안에서만 사용될 수 있다.

## 실습

### Exercise 1.

이 표현식의 값을 계산해보자.  $x = \frac{\sqrt{6}+6}{2} + \frac{\sqrt{13}+13}{2} + \frac{\sqrt{21}+21}{2}$

이 함수를 사용한다.  $y = \sqrt{x} + x$

### Exercise 2.

이 표현식의 값을 계산해보자.  $x = \frac{15+\sqrt{8}}{8+\sqrt{15}} + \frac{6+\sqrt{12}}{12+\sqrt{6}} + \frac{7+\sqrt{21}}{21+\sqrt{7}}$

이 함수를 사용한다.  $y = a + \sqrt{b}$

### Exercise 3.

삼각형의 세 꼭지점의 좌표를 받아서, 삼각형의 둘레를 계산해보자.

### Exercise 4.

사용자가 3개의 값 a, b, c를 입력하면, 아래 표현식의 값을 계산해보자.

$$t = \frac{\max(a, b, c) - \min(a, b, c)}{2 + \max(a, b, c) \cdot \min(a, b, c)}$$

함수  $\text{Min}(a, b, c)$  과  $\text{Max}(a, b, c)$  는 3개의 숫자 값 중 최소값과 최대값을 반환한다.