- Intrinsic functions

Intrinsic (built in) functions are (mostly) mathematical functions that you can use in your calculations

- commonly used intrinsic functions (see appendix B in the book for a complete list)

Transcendental functions

SQRT(x) → $\sqrt{x}$

EXP (x) → e^x

LOG(x) → $\ln(x)$

LOG10(x) → $\log_{10}(x)$

COS(x), SIN(x), TAN(x) → Trig functions

ACOS(x), ASIN(x), ATAN(x) → inverse trig functions (principle angle)

ATAN2(y,x) → $\tan^{-1}\left(\frac{y}{x}\right)$ = 4 quadrant inverse tangent

SINH(x), COSH(x), TANH(x) → hyperbolic trig functions

- cos, sin, tan assume the input angle is in radians
- inverse trig functions return an angle in radians
- not defined for integer values

  SIN (1) will generate a compiler error
- The output type is the same as the input type

  SIN (1.0) = single precision sine

  SIN (1.0d0) = double precision sine
- If you need roots other than $\sqrt{\ }$, you need to use fractional exponents, but be careful

$y = x^{1/3}$ → y=x**(1/3) is wrong (why?)

Several correct ways to do this

```
y = X** (DBLE(1)/3)
  = X** (1/DBLE(3))
  = X** (DBLE(1)/DBLE(3))
  = X** (1.0d0/3)
  = X** (1/3.0d0)
  = X** (1.0d0/3.0d0)
```

mixed mode with exponents.

- if you need an exponent of 3, use an integer 3

```
y = (number)** (integer 3)
  = (number) * (number) * (number)
```

but

```
y = (number) **(double 3)
  = e^(double 3 * ln(number))
```

more expensive, slightly less accurate

- so if you need

$$y = x^4$$

just use

```
y = X**4
```

- a negative number to a double power is undefined (unless you are using complex variables)

```
y = (-1.2d0)** 4  ⟶ OK
y = (-1.2d0)** (1.0d0/3) = e^((1.0d0/3) · ln(-1.2d0))
                                  undefined
```

- other helpful functions

INT(x) → converts x to integer by dropping fractional part

DBLE(i) → promotes i to double

NINT(x) → rounds x to nearest integer

NINT(1.3) = 1

NINT(1.9) = 2

ABS(x) → absolute value

FLOOR(x) → round towards $-\infty$

FLOOR(3.7) = 3

FLOOR(-3.7) = -4

CEILING(x) → rounds towards ∞

CEILING(3.7) = 4

CEILING(-3.7) = -3

MOD(x,y) → remainder when x is divided by y

MAX(a,b,c,...) → maximum of a,b,c,...

MIN(a,b,c,...) → minimum of a,b,c,...

output type is the same as the input type

NOTE: Angles are always assumed to be in radians unless explicitly given in degrees