Programming in Maple

September 6, 2023 11:08 PM

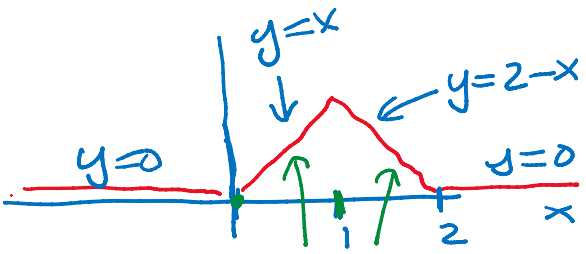
If statements.

```
if cond then
    S1;
    S2;
[else
    e1;
    e2;]
fi;
```

E.g $y = |x|$


```
if x < 0 then
    y := -x;
else
    y := x;
fi;
```

```
if x<0 or x>2 then
    y:=0;
else if x<1 then
        y:=x;
    else
        y:=2-x;
    fi;
fi;
```




```
if x<0 or x>2 then
    y:=0;
elif x<1 then
    y:=x;
else y:=2-x;
fi;
```

Relational operators $< \quad > \quad = \quad \leq \quad \geq \quad \neq$

Maple `< > = <= >= <>`

Boolean operators $\wedge \quad \vee \quad \neg$

Maple `and or not`.

```
if x>0 and x<=1 then y:=x;
elif x>1 and x<2 then y:=2-x;
else y:=0;
fi;
```

TL;

While loops.

```
while cond do
    S1;
    S2;
    ⋮
od;
```

C C++ Java

```
while ( cond ) {
    S1;
    S2;
    ⋮
}
```

For loops.

default is m=1 ; default is n=∞ ; default s=1.

```
for i [from m] [to n] [by s] do
    S1;
    S2;
    ⋮
od;
```

break;
exits the loop.

Calculate $S = \sum_{i=1}^{n} i^2$

```
S := 0;
for i from 1 to n do
    S := S + i^2;
od;
S;
```

S+=i^2; also works

L[i]

Maple list L[i]

```
L := [3, 6, 9, 4];
```

#L or |L|

```
S := 0;
for i to nops(L) do
```

```
S:=0;
for x in L do
```

```
for i to nops(L) do
    x := L[i];
    S += x;
od;
```

```
for x in L do
    S += x;
od;
```

Maple procedures.

inputs (parameters) 0 or more.

```
f := proc( a1, a2, .... )
    [local  l1, l2, ... ;]
    [global g1, g2, ... ;]
    S1;
    S2;
    ⋮
    Sn;   ← value returned.
end;
```

return x;

Example1. $f(x,y) = x^2 - y$.

```
f := proc(x, y)
    x^2-y;
    end;
```

restrict inputs?

$f(1,1) \longrightarrow 0$

$f(z,z) \longrightarrow z^2 - z$

Maple types

```
f := proc( x :: integer,
           y :: integer )
    x^2-y;
end;
```

integer	-3
rational	2/3
numeric	3.1