

Kontrollanweisungen

Schleifen

```
for k from kmin to kmax do  
    Befehle  
end do
```

```
while logischer Ausdruck do  
    Befehle  
end do
```

Verzweigungen

if logischer Ausdruck then

 Befehle

elif logischer Ausdruck then

 Befehle

else

 Befehle

end if

Beispiel

> # loops

Beispiel

- > # loops
- > squares := {}:

Beispiel

```
> # loops
> squares:={}:
> for k from 1 to 9 do
> squares:=squares union {k^2}
> od:
```

Beispiel

```
> # loops
> squares:={}:
> for k from 1 to 9 do
> squares:=squares union {k^2}
> od:
> squares;

{1, 4, 9, 16, 25, 36, 49, 64, 81}
```

Beispiel

```
> # loops
> squares:={}:
> for k from 1 to 9 do
> squares:=squares union {k^2}
> od:
> squares;
           {1, 4, 9, 16, 25, 36, 49, 64, 81}
> fibonacci:=[1,1]: n:=2:
```

Beispiel

```
> # loops
> squares:={}:
> for k from 1 to 9 do
> squares:=squares union {k^2}
> od:
> squares;
           {1, 4, 9, 16, 25, 36, 49, 64, 81}
> fibonacci:=[1,1]: n:=2:
> while fibonacci[2]<100 do
> fibonacci:=[fibonacci[2],fibonacci[1]+fibonacci[2]]
> od:
```


Beispiel

```
> # loops
> squares:={}:
> for k from 1 to 9 do
> squares:=squares union {k^2}
> od:
> squares;
           {1, 4, 9, 16, 25, 36, 49, 64, 81}
> fibonacci:=[1,1]: n:=2:
> while fibonacci[2]<100 do
> fibonacci:=[fibonacci[2],fibonacci[1]+fibonacci[2]]
> od:
> fibonacci;
           [89, 144]
```

Beispiel

> # branching

Beispiel

```
> # branching  
> s:=rand(-1..1): s:=s();  
s := -1
```

Beispiel

```
> # branching
> s:=rand(-1..1): s:=s();
                                s := -1
> if s>0 then
> print(positive)
> elif s<0 then
> print(negative)
> else
> print(zero)
> fi;
```

negative

Beispiel

```
> # branching
> s:=rand(-1..1): s:=s();
                                s := -1
> if s>0 then
> print(positive)
> elif s<0 then
> print(negative)
> else
> print(zero)
> fi;
```

negative