

Assembler-Programmierung

- if-Anweisung

```
doit() {  
    Befehl_1;  
    if(Bedingung) {  
        Befehl_2;  
        Befehl_3;  
    }  
    Befehl_4;  
}
```

----- { If beendet

```
doit: Befehl_1  
      Bedingung_negiert    # slt ...  
      Bedingung_negiert erfüllt -> überspringe Befehle # Label  
                                # !(Bedingung) erfüllt  
                                # ≠ Bedingung nicht erfüllt  
  
      Befehl_2  
      Befehl_3  
Label:Befehl_4  
      jr $ra
```

----- { If beendet

- while-Schleife

```
doit2() {  
    Befehl_1;  
    while(Bedingung) {  
        Befehl_2;  
    }  
    Befehl_3;  
}
```

----- { while beendet

```
doit2:Befehl_1  
while:Bedingung_negiert    # slt ...  
      Bedingung_negiert erfüllt -> überspringe Befehle # Label  
                                # !(Bedingung) erfüllt  
                                # ≠ Bedingung nicht erfüllt  
  
      Befehl_2  
      j while  
Label:Befehl_3  
      jr $ra
```

----- { while beendet

- for-Schleife

```
doit3() {  
    Befehl_1;  
    for(Vorbedingung;  
        Schleifen-Bedingung;  
        Wiederholung) {  
        Befehl_2;  
    }  
    Befehl_3;  
}
```

----- { for beendet

```
doit2:Befehl_1  
      Vorbedingung        # z.B. addi $t0, $0, 0 (int i=0)  
for  :Bedingung_negiert    # slt ...  
      Bedingung_negiert erfüllt -> überspringe Befehle # Label  
                                # !(Bedingung) erfüllt  
                                # ≠ Bedingung nicht erfüllt  
  
      Befehl_2  
      Wiederholung        # z.B. addi $t0, $t0, 1 (i++)  
      j for  
Label:Befehl_3  
      jr $ra
```

----- { for beendet