

Wiederholung: “Schnellkurs” Teil 1

```
val x=4*7+3
```

```
val y=x*(x-29)
```

- ▶ **Wörter:** Bezeichner, Operatoren, Konstanten, Schlüsselwörter
- ▶ **ML Programme:** Folge von Deklarationen
- ▶ **Deklaration:** `val <Bezeichner> = <Ausdruck>`
- ▶ **Ausdruck:** Bezeichner, Konstanten, Operatoren und Klammern



Doppelrollen möglich: `val x = (5=6)`

- ▶ **Typen:** `int bool real int * real int -> int`

Wiederholung: "Schnellkurs" Teil 1

- ▶ **Prozedurdeklaration:** `fun quadrat(x:int) = x*x`



Eine Funktion kann durch
verschiedene Prozeduren berechnet werden

- ▶ **Konditional:** `if ~7>7 then 5>7 else 7>5`



Konsequenz und Alternative
müssen gleichen Typ haben

- ▶ **Lokale Deklarationen:**

```
fun hoch8(x:int) =  
  let  
    val a=x*x  
    val b=a*a  
  in  
    b*b  
  end
```

Wiederholung: “Schnellkurs” Teil 1

▶ Hilfsprozeduren:

```
fun q (y:int) = y*y  
fun hoch8(x:int) = q (q (q x))
```

▶ **Tupel:** zusammengesetzte Werte

- ▶ Bsp: (1, 2, true, 3, 2, 1) Länge 6, 4 Komponente
Länge 6, 4 Komponente
- ▶ 2-stelliges Tupel: Paar
- ▶ 3-stelliges Tupel: Tripel (1,2,(3,4))



0-stelliges Tupel: leeres Tupel

() : unit (spezieller Typ)

- ▶ Projektionen aus Tupeln: #

```
val x = (7, (5, (), 2), 1, 4, (2,3,4,5))
```

```
#2 x
```

```
val it = (5, (), 2): int * unit * int
```