

OCaml: Pattern matching, Datentypen

Notiztitel

08.07.2003

```
# let vokal = function  
  'a'|'e'|'o'|'u'|'i' -> true  
  _ -> false;;  
val vokal : char -> bool = <fun>  
# vokal 'a';;  
- : bool = true  
# vokal 'b';;  
- : bool = false
```

```
let impl x y = match (x,y) with  
  (true,false) -> false  
  |(x,y) -> true;;  
val impl : bool -> bool -> bool = <fun>  
# impl true true;;  
- : bool = true  
# impl true false;;  
- : bool = false
```

```
# let equiv x y = match (x,y) with
(x,x) -> true
|(x,y) -> false;;
Characters 38-39:
(x,x) -> true
^
```

This variable is bound several times in this matching

```
# let equiv x y = match (x,y) with
(true,true) -> true
|(false,false) -> true
|(x,y) -> false;;
val equiv : bool -> bool -> bool = <fun>
```

Typen in Pascal

Operatoren:

int: +, -, *, /, mod,

float: +, -, *, /, ., **

bool: and, or, xor

list: @ (Zusatzoperator)

string: ^ (Zusatzoperator)