

Introduction to Functional Programming in *OCaml*

Roberto Di Cosmo, Yann Régis-Gianas, Ralf Treinen

Week 1 - Sequence 0: Basic Data Types



Overview of Week 1

0. Basic Data Types
1. More Data Types
2. Expressions
3. Identifiers
4. Functions
5. Recursion

Type Inference

- ▶ Types of identifiers are inferred, not declared.
- ▶ A distinguishing feature of OCaml.
- ▶ Reconciles the flexibility of untyped languages with the safety of typed languages.
- ▶ A very rich type system.
- ▶ Polymorphic types provide additional flexibility.

Syntax Errors and Type Errors I

```
3*(4+1)-7;;
```

```
# - : int = 8
```

```
17 +;;
```

```
# Characters 5-7:
```

```
17 +;;  
   ^^
```

```
Error: Syntax error
```

Syntax Errors and Type Errors II

```
42 + "hello";;
```

```
# Characters 6-13:
```

```
42 + "hello";;  
      ^^^^^^^
```

```
Error: This expression has type string but an expression was  
      expected of type  
      int
```

A Rich Type System

- ▶ Basic types `int`, `bool`, `float`, `string`, `char` ...
- ▶ Structured types: see Weeks 2 and 3
- ▶ Functional types: see Week 4
- ▶ Reference types: see Week 5
- ▶ Objects and classes: not in this Mooc

Integers

- ▶ Type: `int`
- ▶ Values: $-2^{62} \dots 2^{62} - 1$ (on 64-bit architectures, much less in the exercise environment!)
- ▶ Arithmetic operators: `+`, `-`, `*`, `/`
- ▶ Calculations performed modulo 2^{63}

Integer Pitfalls

- ▶ `/` is integer division : $7/2=3$
- ▶ `mod` is integer remainder : $7 \bmod 2 = 1$

Examples Integers I

```
2+3*5;;
```

```
# - : int = 17
```

```
5/2;;
```

```
# - : int = 2
```

```
-5 mod 3;;
```

```
# - : int = -2
```

Booleans

- ▶ Type: `bool`
- ▶ Values: `true` and `false`
- ▶ Boolean operators: `&&`, `||`, `not`
- ▶ Comparison operators: `<`, `>`, `=`, `<=`, `=>`

Boolean Pitfalls

- ▶ Negation is `not`; `!` is wrong
- ▶ Conjunction is `&&`; `&` is wrong
- ▶ Conjunction is `&&`; the keyword `and` has a different meaning
- ▶ You can only compare values of the same type

Examples Booleans I

```
true && true;;
```

```
# - : bool = true
```

```
false || true;;
```

```
# - : bool = true
```

```
true && not (false || true);;
```

```
# - : bool = false
```

```
1 < 7;;
```

```
# - : bool = true
```

Examples Booleans II

```
5.0 > "hello";;
```

```
# Characters 7-14:
```

```
5.0 > "hello";;  
      ^^^^^^^
```

```
Error: This expression has type string but an expression was  
       expected of type  
       float
```

```
(7.56 <= 8e32) && (6 > -3);;
```

```
# - : bool = true
```

To Know More

The OCaml Manual:

- ▶ The core library
 - ▶ Module Pervasives:
 - ▶ Boolean Operations
 - ▶ Integer Arithmetic