

Learning to Program with F#  
Exercises  
Department of Computer Science  
University of Copenhagen

Jon Sparring, Martin Elsmann, Torben Mogensen, Christina Lioma

October 21, 2022

## 0.1 Weekday

### 0.1.1 Teacher's guide

**Emne** Sumtyper og lister

**Sværhedsgrad** Let

### 0.1.2 Introduction

In the following, you are to work with the discriminated union weekday:

```
type weekday =  
  Monday | Tuesday | Wednesday | Thursday | Friday | Saturday | Sunday
```

which represents the days of the week.

### 0.1.3 Exercise(s)

**0.1.3.1:** Make a function `dayToNumber : weekday -> int` which given a weekday returns an integer, such that Monday is 1, Tuesday is 2, etc.

**0.1.3.2:** Make a function `nextDay : weekday -> weekday` which given a day returns the next day, i.e., Tuesday is the next day of Monday, and Monday is the next day of Sunday.

**0.1.3.3:** Make a function `numberToDay : n : int -> weekday option` which given an integer in the range `1...7` returns one of the weekdays Monday...Sunday as an option type. An integer not in the range, i.e. `< 1` or `> 7` should return `None`.

Examples:

The call `numberToDay 1` should return `Some Monday` and The call `numberToDay 42` should return `None`.