Recap

- FP basics
    - Bindings, scoping, values
    - Immutability vs mutation
    - Defining functions
        - Curried form vs tupled form
    - Anonymous functions (lambdas)
    - Higher-order functions
- Discriminated unions and pattern matching
- Additional matching patterns for **lists**, **arrays**, **records**
    - `fsharp match myList with | [] -> ... | head :: tail -> ... | head1 :: head2 :: tail -> ... | [el1; el2; el3] -> ... | _ -> ...`
    - `fsharp match myArray with | [||] -> ... | [| el1; el2; el3 |] -> ... | _ -> ...`
    - `fsharp match myRecord with | { FirstName = fname } -> ... | { FirstName = fname; Age = age } -> ... | _ -> ...`
- Sequences (lazy vs. eager)
    - Aggregate operations (`map`, `fold`, `iter`, etc.)

Imperative F

- Reference cells
    - `fsharp let x = ref 1 x := 2 incr x // increments an int ref`
- Mutable bindings
    - `fsharp let mutable x = 1 x <- 2`
- Mutable record fields
    - `type Person =    {       mutable FirstName: string       mutable LastName: string       Age: int    }let john =    {       FirstName="John"       LastName="Smith"       Age=30    }john.FirstName <- "Johnny"`
- `for` loops
    - `for i = 1 to 99 do    printfn "%d" i`
- `while` loops
    - `while i < 100 do    printfn "%d" i    i <- i+1`
- Exceptions
    - Defining
        - `exception FooBar of string * int`
        - as a class derived from `System.Exception`
    - Throwing exceptions
        - `raise`
        - `failwith` and `failwithf` -> `System.Exception`
        - `invalidArg` -> `System.InvalidArgumentException`
- `null` values (doc)
    - Can't arise from F# code, unless a class type is annotated to allow `null` values via the `AllowNullLiteral` attribute.

    - Use `null` with .NET APIs if needed
    - Can pattern match against `null` to check for nullness.

Make your code more functional

- Use **recursion** instead of loops
    - Usually this involves adding an accumulator argument
- Make your recursive functions **tail recursive**
    - `fsharp // NOT tail recursive if n < 2 then n else n * fact (n-1)`
- Avoid shared references in records