# Webes és mobil programozás F#-ban 1. gyakorlat

## Berezvai Dániel jegyzete <http://elte.3ice.hu/>

Kezdés: 14:00

Vége: 15:30

Tanár: Sütőné Kovács Andrea

<http://people.inf.elte.hu/andieri>

andieri@caesar.elte.hu

## F#

Nem Haskell. A Haskell lusta, az F# alapból mohó nyelv.

Az F# egyik tervezője magyar. (Gránicz Ádám)

.NET alapú, funkcionális, imperatív nyelv.

Írjuk meg a GUI-t imperatívan, a működést pedig funkcionálisan.

## F# honlapok

Mire fordul: Statikus HTML + JavaScript (Mindenhol működik)

<http://cloudsharper.com/> (tananyag, elérhető angolul)

<http://fpish.net/> közösségi oldal (Mint a Stack Overflow)

## Félév menete, jegyszerzés

Őszi szünetig $F\#$ alapozás.

Őszi szünet előtt az utolsó órán Adám vendégelőadó.

Utána két hét kimarad:

* Október 23 szünet
* Őszi szünet

Beadandó feladatot december közepéig kell elkészíteni és bemutatni.

Utána példákat nézünk:

* Bevásárlókosár
* Adatbázis kapcsolat
* Login form

Minden JavaScript API F# alatt használható.

## CloudSharper program ma nem működik, helyette Visual Studio lesz ideiglenesen

Pont jó, le tudom tölteni jövő hétre. (Még mindig nincs WiFi a második emeleten…)

## Szám, változó

> let intNumber=12315;;

val intNumber : int = 12315

A két pontosvessző csak az interaktív ablakban kell.

### További változók

> let floatNumber = 12.3545

let charLiteral = 'F'

let stringLiteral = "Welcome int F# World!";;

val floatNumber : float = 12.3545

val charLiteral : char = 'F'

val stringLiteral : string = "Welcome to the F# World!"

## Operátorok

Minden szokásos.

Modulo:

> intNumber % 6;;

val it : int = 3

Az eredmény név nélküli érték ($it$)

## Negatív érték zárójellel

> - (215 \* 4);;

val it : int = -860

## Float float-tal szorzása rendben megy

> floatNumber \* 25.56;;

val it : float = 315.78102

## Függvény

> let mulInt x y = x\*y;;

val mulInt : int -> int -> int

### Rossz típus

> mulInt 52.45 6.54;;

 mulInt 52.45 6.54;;

 -------^^^^^

stdin(19,8): error FS0001: This expression was expected to have type int but here has type float

### Új függvény

> let mulFloat x y : float = x \* y;;

val mulFloat : float -> float -> float

### Most már működik

> mulFloat 52.45 6.54;;

val it : float = 343.023

## Konkatenáció

> let concatenateString s1 s2 : string = s1 + s2;;

val concatenateString : string -> string -> string

### Használata

> let string = concatenateString "Welcome to the " "F# world!";;

val string : string = "Welcome to the F# world!"

## N-edik karakter függvény

> let nthChar n (s:string) = s.[n];;

val nthChar : int -> string -> char

> nthChar 0 "hi";;

val it : char = 'h'

> nthChar 1 "hi";;

val it : char = 'i'

> nthChar 5 string;;

val it : char = 'm'

### Substring

> let substring n m (s:string) = s.[n..m];;

val substring : int -> int -> string -> string

> substring 0 6 string;;

val it : string = "Welcome"

Karaktertömböt 0-tól indexeljük.

## Igaz hamis boolean értékek

> true;;

val it : bool = true

> true && false;;

val it : bool = false

> true || false;;

val it : bool = true

## Párok (tuple)

> (7, "valami", 32.6);;

val it : int \* string \* float = (7, "valami", 32.6)

### Tovább

> let t1 = ("Example", 4);;

val t1 : string \* int = ("Example", 4)

> let t2 = (56, 12.6, 'c');;

val t2 : int \* float \* char = (56, 12.6, 'c')

> (t1, t2);;

val it : (string \* int) \* (int \* float \* char) =

 (("Example", 4), (56, 12.6, 'c'))

### Első

> fst t1;;

val it : string = "Example"

> snd t1;;

val it : int = 4

## Feladat: Mintaillesztés

> let t = ("str",1,'c');;

val t : string \* int \* char = ("str", 1, 'c')

> let sndOfTriplet (a,b,c) = b;;

val sndOfTriplet : 'a \* 'b \* 'c -> 'b

> sndOfTriplet t;;

val it : int = 1

### Bonyolultabban

> let sndOfTriplet ((a,b,c):(string \* int \* char)) : int = b;;

val sndOfTriplet : string \* int \* char -> int

## Lista

### Üres

> [];;

val it : 'a list = []

### Elemek felsorolása

Utolsó elem mögé nem kötelező a pontosvessző.

> [1;2;3;4;5];;

val it : int list = [1; 2; 3; 4; 5]

### Utolsó kifejezés újra kérése az $it$ szóval

> it;;

val it : int list = [1; 2; 3; 4; 5]

### Elé fűzés

> 0::it;;

val it : int list = [0; 1; 2; 3; 4; 5]

### Range

> ['a'..'z'];;

val it : char list =

 ['a'; 'b'; 'c'; 'd'; 'e'; 'f'; 'g'; 'h'; 'i'; 'j'; 'k'; 'l'; 'm'; 'n'; 'o';

 'p'; 'q'; 'r'; 's'; 't'; 'u'; 'v'; 'w'; 'x'; 'y'; 'z']

### Out of memory error demonstrálása

[1..100000000];;

Kösz, lefagyott az IDE-m…

## Generálás

> [for x in 1..60 -> x\*x];;

val it : int list =

 [1; 4; 9; 16; 25; 36; 49; 64; 81; 100; 121; 144; 169; 196; 225; 256; 289;

 324; 361; 400; 441; 484; 529; 576; 625; 676; 729; 784; 841; 900; 961; 1024;

 1089; 1156; 1225; 1296; 1369; 1444; 1521; 1600; 1681; 1764; 1849; 1936;

 2025; 2116; 2209; 2304; 2401; 2500; 2601; 2704; 2809; 2916; 3025; 3136;

 3249; 3364; 3481; 3600]

## Két lista összefűzése

> [1;2]@[3;4];;

val it : int list = [1; 2; 3; 4]

## Feladat mutable vs. immutable bemutatására

1. Készítsünk két listát
2. Fűzzük össze
3. Adjunk még egy elemet az első listához

Mi történik?

Várhatóan nem fog frissülni a harmadik lista az új elemmel. Még ha újra összefűzzük sem.

> let a=[1;2]

let b=[3;4]

let c=a@b;;

val a : int list = [1; 2]

val b : int list = [3; 4]

val c : int list = [1; 2; 3; 4]

> 0::a;;

val it : int list = [0; 1; 2]

> c;;

val it : int list = [1; 2; 3; 4]

### Új változót kell használni.

> let d = 0::a;;

val d : int list = [0; 1; 2]

## Lista

### Hossza

> List.length [1..10];;

val it : int = 10

### Lista első eleme és maradéka

> List.head [1..10];;

val it : int = 1

> List.tail [1..10];;

val it : int list = [2; 3; 4; 5; 6; 7; 8; 9; 10]

### Zip, unzip

> List.zip [1..5] [6..10];;

val it : (int \* int) list = [(1, 6); (2, 7); (3, 8); (4, 9); (5, 10)]

> List.unzip it;;

val it : int list \* int list = ([1; 2; 3; 4; 5], [6; 7; 8; 9; 10])

### Hozzáfűz

> List.append [1..5] [6..10];;

val it : int list = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10]

### Tömbbé alakít, visszaalakít

> List.toArray [1..10];;

val it : int [] = [|1; 2; 3; 4; 5; 6; 7; 8; 9; 10|]

> List.ofArray it;;

val it : int list = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10]

## Feladat: Listakifejezés ahol felváltva vannak igaz, hamis értékek

val it : bool list =

 [true; false; true; false; true; false; true; false; true; false]

## Lépésközös range

Első két szám a lépésköz, harmadik a meddig.

> [0..6..90];;

val it : int list =

 [0; 6; 12; 18; 24; 30; 36; 42; 48; 54; 60; 66; 72; 78; 84; 90]

## Mintaillesztés listára

(Ez a lista hosszát adja vissza.)

> let patternMatch list =

 match list with

 | [] -> 0

 | x::xs -> 1 + List.length xs;;

val patternMatch : 'a list -> int

> patternMatch [1..5];;

val it : int = 5

> patternMatch [];;

val it : int = 0

## Mintaillesztés párra

> let pairMatch (a,b) =

 match a with

 | [] -> (0,b)

 | x::xs -> (List.length a, b);;

val pairMatch : 'a list \* 'b -> int \* 'b

> pairMatch ([1..5],4);;

val it : int \* int = (5, 4)

## Joly

$x∷xs$ helyett egy \_ (underscore)

> let pairMatch (a,b) =

 match a with

 | [] -> (0,b)

 | \_ -> (List.length a, b);;

val pairMatch : 'a list \* 'b -> int \* 'b

## Őrfeltétel

> let pairMatch (a,b)=

 match a with

 | [] when b = false -> ((-1),(-1))

 | x::xs -> (List.length a, 4);;

 match a with

 ----------^

stdin(36,11): warning FS0025: Incomplete pattern matches on this expression.

val pairMatch : 'a list \* bool -> int \* int

## Házi feladat

* $not$ függvény
* Függvény listák listáját kapja, visszatér az első (…)
* Függvény vár egy számot paraméterül. Ha páratlan, akkor 1, ha nem, akkor a bemenetet osztja 2-vel.

Gyakorlat vége.

## HF Megoldás

**let** not a **=**

 **match** a **with**

 **|** **true** **->** **false**

 **|** **\_** **->** **true**

**let** d x **=**

 **match** x**%**2 **with**

 **|** 0 **->** x**/**2

 **|** **\_** **->** 1

### Teszt

> not true;;

val it : bool = false

> not false;;

val it : bool = true

### Teszt 2

> d 6;;

val it : int = 3

> d 2;;

val it : int = 1

> d 0 ;;

val it : int = 0

> d(d 8);;

val it : int = 2

> d(d 16);;

val it : int = 4

> d(d(d 1024));;

val it : int = 128

> d(d(d(d(d 1024))));;

val it : int = 32