

## Funkcionális programozás 3. gyakorlat

### Emlékeztető: Rendezett n-es/Tuple

- Összetett adattípus, komponensekből áll
- Bármilyen típusú értéket képes tárolni
- Párokra illeszthető minta: (x, y)
- <http://lambda.inf.elte.hu/Patterns.xml#mint%C3%A1k-p%C3%A1rok-ra-prelude.fst>

Példák:

```
fst :: (a, b) -> a
fst (a, b) = a
```

```
snd :: (a, b) -> b
snd (a, b) = b
```

1. Definiáljuk a `swap` függvényt, amely megcseréli egy pár két elemét!

```
swap (1, True) == (True, 1)
swap ('a', 2.5) == (2.5, 'a')
```

2. Definiáljuk a `triplicate` függvényt, amely egy tetszőleges paraméterként adott `x` esetén egy rendezett hármast ad vissza, amelynek minden eleme `x`!

```
triplicate 0 == (0,0,0)
triplicate True == (True, True, True)
```

3. Definiáljuk az `addPair` függvényt, amely elemenként összead két számokból álló rendezett párt! Oldjuk meg a feladatot mintaillesztéssel és a `fst`, `snd` függvények segítségével is!

```
addPair (0,0) (1,2) == (1,2)
addPair (1,2) (-1,-2) == (0,0)
addPair (2,2) (4,5) == (6,7)
```

(4.) Definiáljuk a `divAndMod` függvényt, amely megadja két egész számnak az egymással való egész osztás eredményét és az osztási maradékot egy rendezett párként!

```
divAndMod 10 5 == (2,0)
divAndMod 9 5 == (1,4)
```

### Listák

<http://lambda.inf.elte.hu/Lists.xml>

Példa konkrét listákra:

```
exampleList1 :: Num a => [a]
exampleList1 = [1,2,3,4,5]

exampleList2 :: [Bool]
exampleList2 = [True, False, False, True]

exampleList3 :: [Char] -- == String
exampleList3 = ['a', 'l', 'm', 'a']

exampleList4 :: [(Integer, Bool)]
exampleList4 = [(2, True), (-6, False)]
```

### Pontpont kifejezések

```
exampleList5 = [1..10]
exampleList6 = [-10,-9..10]
exampleList7 = ['a'..'z']
```

### Függvények számokat tartalmazó listákon

```
exampleSum = sum [1..10]
exampleProd = product [1..10]
```

### 5. Gyakorló feladatok

- Soroljuk fel 0-tól 30-ig a számokat hármassával!
- Soroljuk fel 10-től visszafelé -10-ig a számokat!
- Számoljuk ki a 20 és 30 között lévő páros számok összegét!

### Listagenerátorok/halmazkifejezések (list comprehension)

<http://lambda.inf.elte.hu/Comprehensions.xml>

Példa: páros számok négyzete az [1..10] intervallumban

```
listComp :: [Integer]
listComp = [n^2 | n <- [1..10], even n]
```

### 6. Gyakorló feladatok

- Állítsuk elő a 2 hatványait növekvő sorrendben 1-től  $2^{10}$ -ig!
- Soroljuk fel az első 10 négyzetszám kétszeresét!
- Soroljuk fel 1 és 100 között azokat a számokat, amelyek oszthatóak 3-mal, de nem oszthatóak 5-tel!
- Soroljuk fel a 60 osztóit!

- e) Állítsunk elő 1-50 között a számokat és 3-mal való osztási maradékukat egy rendezett párokat tartalmazó listában.
- f) Állítsuk elő azt a listát, amely sorrendben tartalmazza az összes (óra, perc) párt!

**7. Definiáljuk az `isPrime` függvényt, amely eldönti egy tetszőleges számról, hogy prím-e!**