

In deze opdracht ga je zelf functies maken die je kunt gebruiken voor de eindopdracht (het Markov-algoritme).

Theorie

Lees in de syllabus paragraaf 3.12 over het zelf maken van functies en probeer de voorbeelden uit in Racket.

Opdracht 1 – een functie

Maak een functie die een willekeurig item uit een lijst geeft

Voorbeeld:

```
(get-element '(a b c d e))
```

Heeft als mogelijk resultaat:

```
'c
```

Opdracht 2 – hergebruik van functies

Maak een functie die een lijst maakt die is samengesteld uit drie items uit een gegeven lijst. Elementen mogen meerdere keren voorkomen.

Voorbeeld:

```
(create-list '(a b c d e))
```

Heeft als mogelijk resultaat:

```
'(d c c)
```

Opdracht 3 – hergebruik van functies

Maak een functie die een item uit een lijst met lijsten kiest. Randvoorwaarde: het eerste symbool van elke sub-lijst doet niet mee. In het onderstaande voorbeeld mogen dus geen hoofdletters gekozen worden.

Voorbeeld:

```
(get-element '((A b c) (D e f) (K l m) (P q r)))
```

Heeft als mogelijk resultaat:

```
'm
```

maar niet A, D, K of P

Opdracht 4 –

Maak een functie die een lijst maakt die bestaat uit elementen die willekeurig uit een lijst met lijsten gekozen worden, waarbij ook weer de eerste items uit elke sub-lijst niet meedoen.

Voorbeeld:

```
(create-list '((a b c) (b c d) (c d) (d e)))
```

Heeft als mogelijk resultaat:

```
`(d c c)
```