Racket Assignment #1: Getting Acquainted with Racket/DrRacket + LEL Sentence Generation

Abstract

My first experience working with the Racket programming language will be demonstrated in this project. Particularly, it clarifies how LEL sentence generation functions in Racket.

Code for the LEL Sentence Generator

```
#lang racket
;-------------------------
; LEL sentence generator, with helper PICK
; several applications of APPEND, several
; applications of LIST, and one use of MAP
; with a LAMBDA function.

( define ( pick list )
   ( list-ref list ( random ( length list ) ) )
)

( define ( noun )
    ( list ( pick ' ( robot baby toddler hat dog ) ) )
)

( define ( verb )
   ( list ( pick ' ( kissed hugged protected chased hornswoggled ) ) )
)

( define ( article )
   ( list ( pick ' ( a the ) ) )
)

( define ( qualifier )
   ( pick ' ( ( howling ) ( talking ) ( dancing )
```

```
              ( barking ) ( happy ) ( laughing )
              () () () () () ()
            )
    )
)

( define ( noun-phrase )
   ( append ( article ) ( qualifier ) ( noun ) )
)

( define ( sentence )
   ( append ( noun-phrase ) ( verb ) ( noun-phrase ) )
)

( define ( ds ) ; display a sentence
   ( map
     ( lambda ( w ) ( display w ) ( display " " ) )
     ( sentence )
   )
   ( display "") ; an artificial something
)
```

Demo for the LEL Sentence Generator

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( pick '(red yellow blue ) )
'blue
> ( pick '(red yellow blue ) )
'blue
> ( pick '(red yellow blue ) )
'yellow
> ( pick '(red yellow blue ) )
'yellow
> ( pick '( Racket Prolog Haskell Rust ) )
'Prolog
> ( pick '( Racket Prolog Haskell Rust ) )
'Haskell
> ( pick '( Racket Prolog Haskell Rust ) )
'Prolog
> ( pick '( Racket Prolog Haskell Rust ) )
'Racket
```

```
> ( noun )
'(baby)
> ( noun )
'(toddler)
> ( noun )
'(hat)
> ( noun )
'(baby)
> ( verb )
'(kissed)
> ( verb )
'(kissed)
> ( verb )
'(hugged)
> ( verb )
'(hornswoggled)
> ( article )
'(the)
> ( article )
'(the)
> ( article )
'(a)
> ( article )
'(the)
> ( qualifier )
'(dancing)
> ( qualifier )
'()
> ( qualifier )
'()
> ( qualifier )
'()
> ( qualifier )
'(talking)
> ( qualifier )
'()
> ( qualifier )
'(barking)
> ( qualifier )
'(talking)
> ( qualifier )
'()
> ( qualifier )
'()
```

```
> ( qualifier )
'(dancing)
> ( qualifier )
'()
> ( qualifier )
'(barking)
> ( qualifier )
'(happy)
> ( qualifier )
'()
> ( qualifier )
'()
> ( noun-phrase )
'(the laughing dog)
> ( noun-phrase )
'(the robot)
> ( noun-phrase )
'(a baby)
> ( noun-phrase )
'(a toddler)
> ( noun-phrase )
'(the happy toddler)
> ( noun-phrase )
'(the toddler)
> ( noun-phrase )
'(a dancing hat)
> ( noun-phrase )
'(a robot)
> ( sentence )
'(a howling hat hornswoggled a robot)
> ( sentence )
'(a baby chased a dancing robot)
> ( sentence )
'(the talking dog chased the howling hat)
> ( sentence )
'(the dancing baby hugged the toddler)
> ( sentence )
'(the talking hat protected the robot)
> ( sentence )
'(a robot chased a dog)
> ( sentence )
'(the baby protected the talking hat)
> ( sentence )
'(a toddler kissed a laughing hat)
```

```
> ( ds )
the dog hornswoggled a happy robot
> ( ds )
the dancing toddler hornswoggled the barking baby
> ( ds )
the talking dog protected a dog
> ( ds )
the talking hat hugged a dancing baby
> ( ds )
a barking baby chased a laughing toddler
> ( ds )
a talking toddler kissed the talking robot
> ( ds )
a robot chased a dancing baby
> ( ds )
a dog chased a hat
> ( ds )
a toddler hornswoggled a talking hat
> ( ds )
the happy robot hugged the talking baby
> ( ds )
the baby chased the toddler
> ( ds )
the baby hornswoggled a happy toddler
>
```