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

---

Abstract

This assignment is to familiarize myself with the introduction of the Racket programming language as well as the DRRacket software

---

LEL Sentence Generator Code

```
#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

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

'Rust
> (noun)
'(dog)
> (noun)
'(hat)
> (noun)
'(hat)
> (noun)
'(toddler)
> (verb)
'(chased)
> (verb)
'(hugged)
> (verb)
'(protected)
> (verb)
'(hugged)
> (article)
'(a)
> (article)
'(the)
> (article)
'(a)
> (article)
'(a)
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'(happy)
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'(laughing)
> (qualifier)
'(talking)
> (qualifier)

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

the happy toddler hugged the dancing baby
> (ds)
a happy robot protected a hat
>