Damian Randall 2/6/2023

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

Abstract:

In this assignment we were asked to mindfully type a pre-generated Racket language program in order to establish familiarity and serve as an introduction to our time with Racket. We were asked to simply type the program and demo the program while conscientious of our actions

The code:

```
#lang racket
;-----------------------------------------------
; LEL sentence generator, with helper PICK,
; serveral 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 todler 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))
'blue
> (pick ' (red yellow blue))
'yellow
> (pick ' (red yellow blue))
'red
> (pick ' (red yellow blue))
'blue
```

```
> ( pick '( Racket Prolog Haskell Rust ) )
'Rust
> ( pick '( Racket Prolog Haskell Rust ) )
'Haskell
> ( pick '( Racket Prolog Haskell Rust ) )
'Haskell
> ( pick '( Racket Prolog Haskell Rust ) )
'Prolog
> ( noun )
'(robot)
> ( noun )
'(robot)
> ( noun )
'(hat)
> ( noun )
'(hat)
> ( verb )
'(chased)
> ( verb )
'(kissed)
> ( verb )
'(hornswoggled)
> ( verb )
'(protected)
> ( article )
'(a)
> ( article )
'(a)
> ( article )
'(a)
```

> (article)

'(the)

> (qualifier)

'()

> (qualifier)

'(happy)

> (qualifier)

'(barking)

> (qualifier)

'()

> (qualifier)

'()

> (qualifier)

'(dancing)

> (qualifier)

'()

> (qualifier)

'(laughing)

> (qualifier)

'()

> (qualifier)

'(talking)

> (qualifier)

'()

> (qualifier)

'(barking)

> (qualifier)

'(talking)

> (qualifier)

'(laughing)

> (qualifier)

'()

> (qualifier)

'(howling)

> (noun-phrase)

'(a dog)

> (noun-phrase)

'(the todler)

> (noun-phrase)

'(a dog)

> (noun-phrase)

'(a baby)

> (noun-phrase)

'(the talking dog)

> (noun-phrase)

'(the talking baby)

> (noun-phrase)

'(a robot)

> (noun-phrase)

'(a robot)

> (sentence)

'(a hat protected the dancing dog)

> (sentence)

'(the happy hat hornswoggled the robot)

> (sentence)

'(the baby hugged a dancing todler)

> (sentence)

'(the hat hugged the todler)

> (ds)

the hat kissed the hat

> (ds)

a happy todler kissed a barking todler

> (ds)

the dog hornswoggled a talking dog

> (ds)

a todler hornswoggled a dancing todler

> (ds)

a dog protected a barking dog

> (ds)

the dancing baby kissed a baby

> (ds)

the dancing dog kissed the talking todler

> (ds)

the dancing hat hugged a hat

> (ds)

the talking todler protected a barking todler

> (ds)

a hat hornswoggled a happy hat

> (ds)

the robot kissed a howling dog

> (ds)

the hat protected the happy dog

>