Timothy Astacio

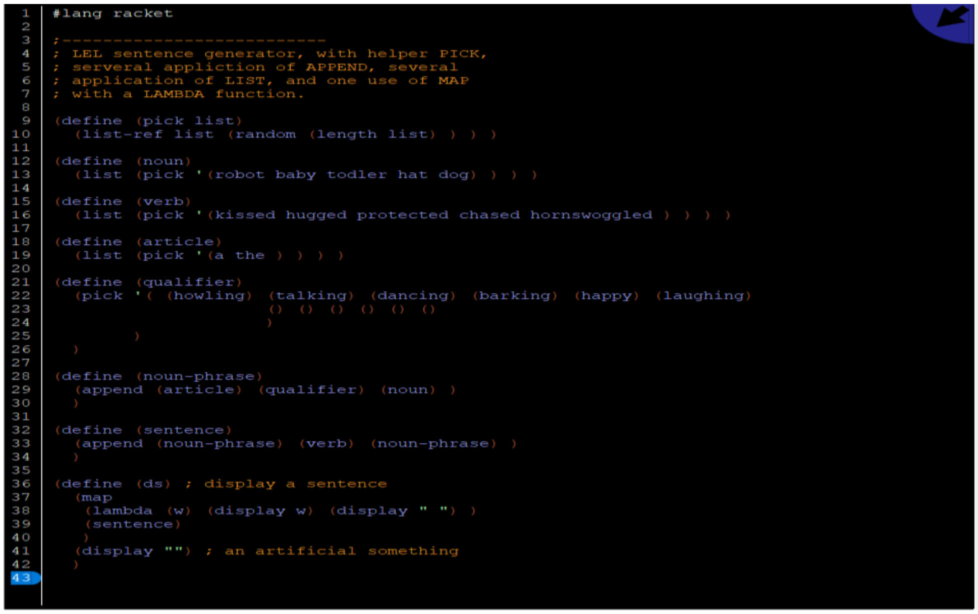
CSC344 Programming Languages

02/06/2023

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

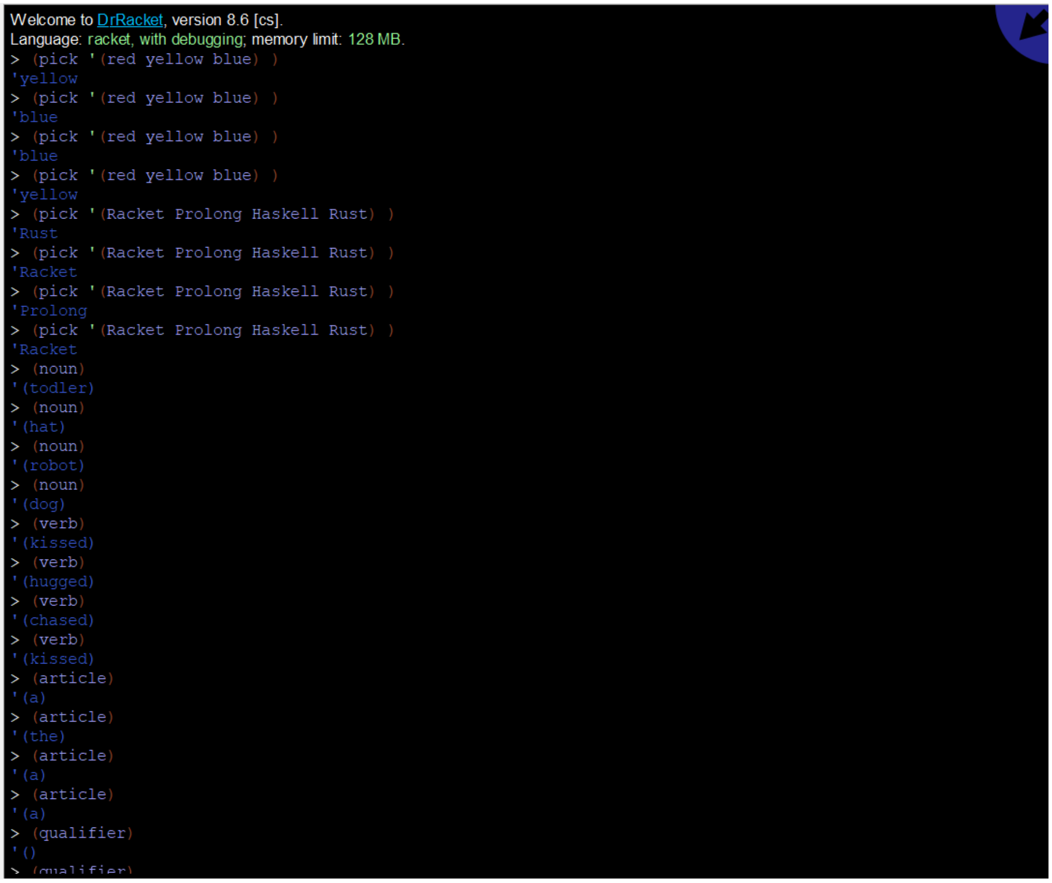
Abstract: The goal of this first program is to familiarize oneself with Racket/Dr Racket. In order to achieve this, we ran a LEL (little English language) sentence generator. This program generates random words and sentences which the user can define. The program was written to give a full understanding of how the racket syntax and environment works. The code and demo for the LEL sentence generator is placed below.

CODE FOR ASSIGNMENT:

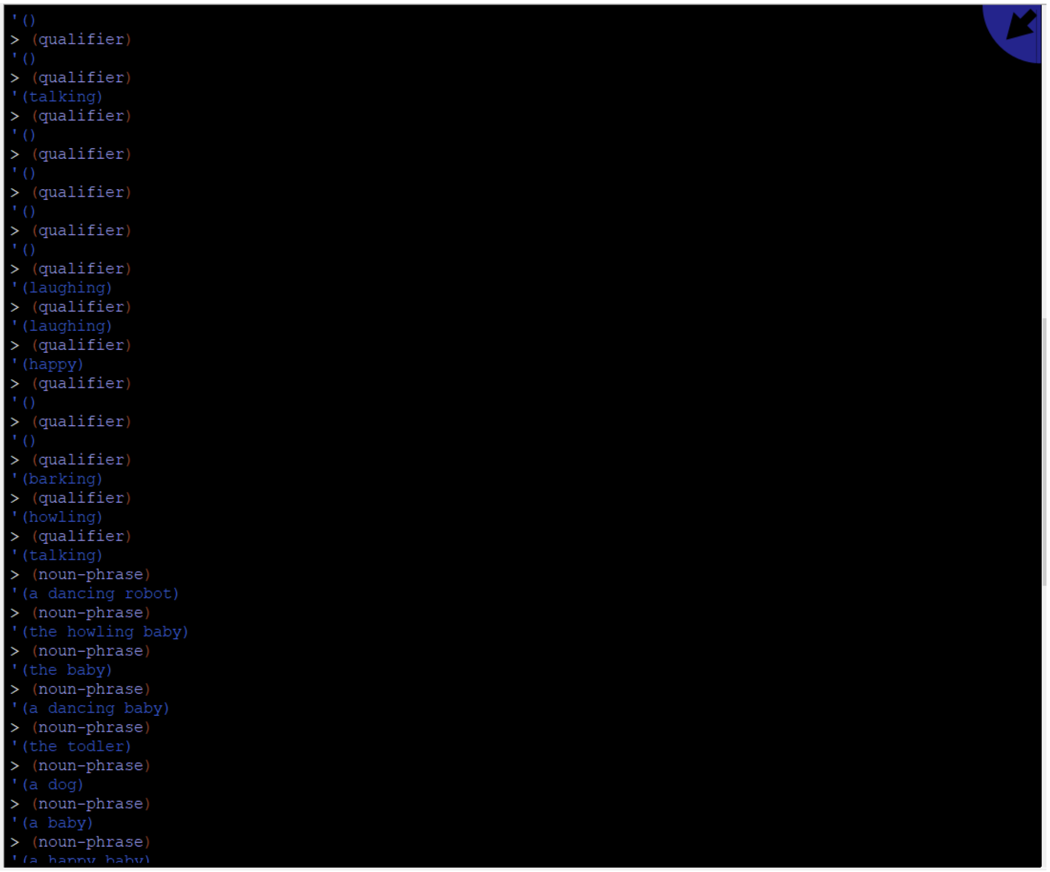


```
#lang racket

;---------------------------
; LEL sentence generator, with helper PICK,
; serveral appliction of APPEND, several
; application 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
  )
```

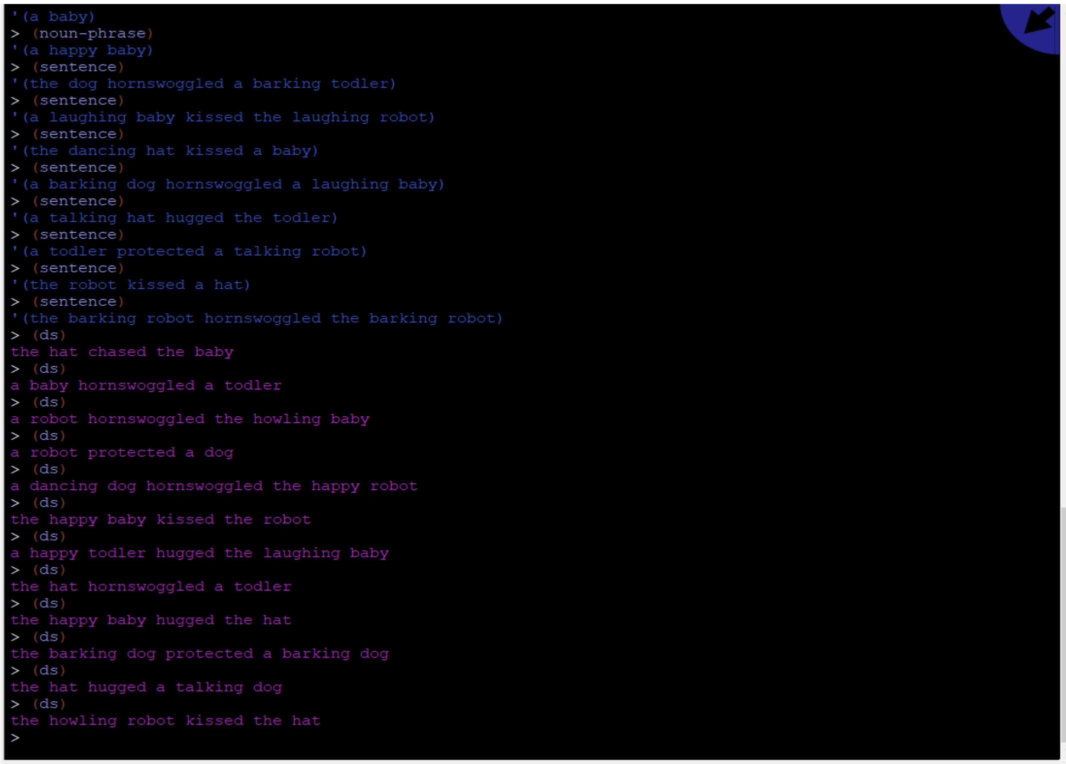
DEMO FOR THE LEL SENTENCE GENERATOR



```
Welcome to DrRacket, version 8.6 [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) )
'yellow
> (pick '(Racket Prolong Haskell Rust) )
'Rust
> (pick '(Racket Prolong Haskell Rust) )
'Racket
> (pick '(Racket Prolong Haskell Rust) )
'Prolong
> (pick '(Racket Prolong Haskell Rust) )
'Racket
> (noun)
'(todler)
> (noun)
'(hat)
> (noun)
'(robot)
> (noun)
'(dog)
> (verb)
'(kissed)
> (verb)
'(hugged)
> (verb)
'(chased)
> (verb)
'(kissed)
> (article)
'(a)
> (article)
'(the)
> (article)
'(a)
> (article)
'(a)
> (qualifier)
'()
> (qualifier)
```



```
'()
> (qualifier)
'()
> (qualifier)
'(talking)
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'(laughing)
> (qualifier)
'(laughing)
> (qualifier)
'(happy)
> (qualifier)
'()
> (qualifier)
'()
> (qualifier)
'(barking)
> (qualifier)
'(howling)
> (qualifier)
'(talking)
> (noun-phrase)
'(a dancing robot)
> (noun-phrase)
'(the howling baby)
> (noun-phrase)
'(the baby)
> (noun-phrase)
'(a dancing baby)
> (noun-phrase)
'(the todler)
> (noun-phrase)
'(a dog)
> (noun-phrase)
'(a baby)
> (noun-phrase)
'(a happy baby)
```



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