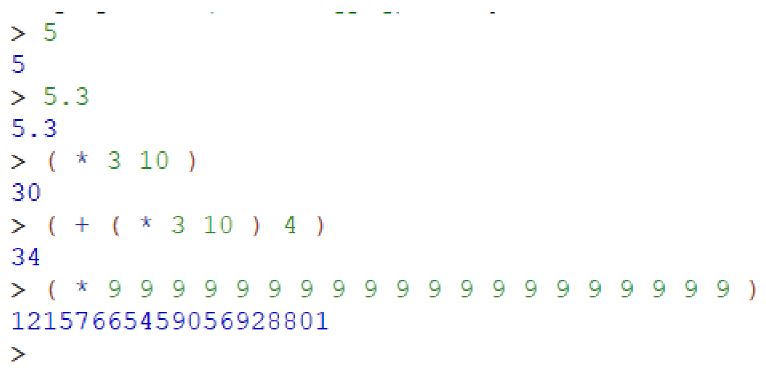
# First Racket Programming assignment solution

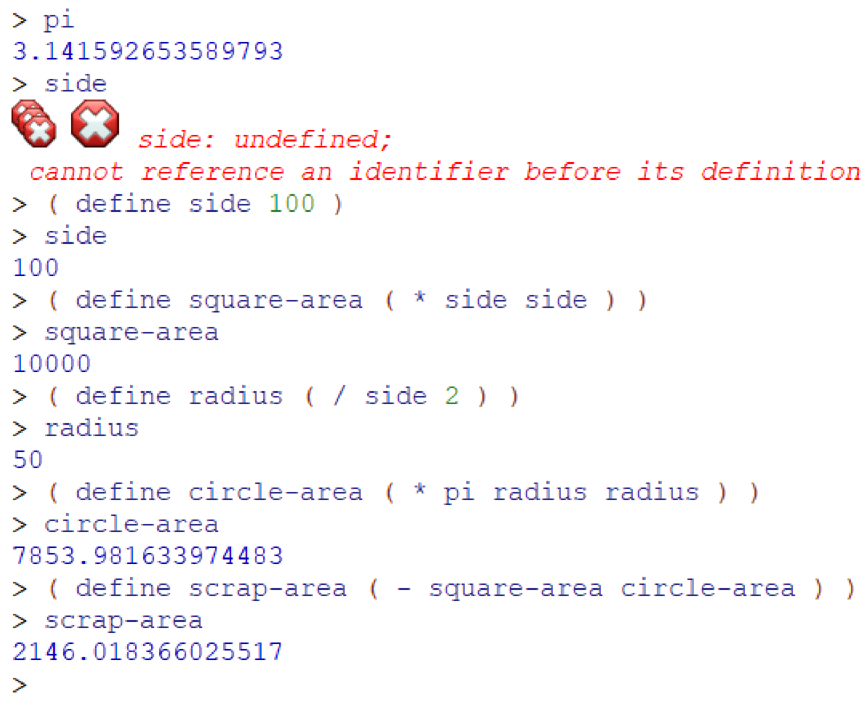
Learning Abstract: In this assignment I learned how to do some basic numeric computations in racket. I learned how to define and call variables. I also learned how to include aka require a Racket library and I used that library to give visual representations of the problem sets I worked on. For this I used the interactions pan of the DrRacket PDE.

## Interaction: Simple Numeric Processing



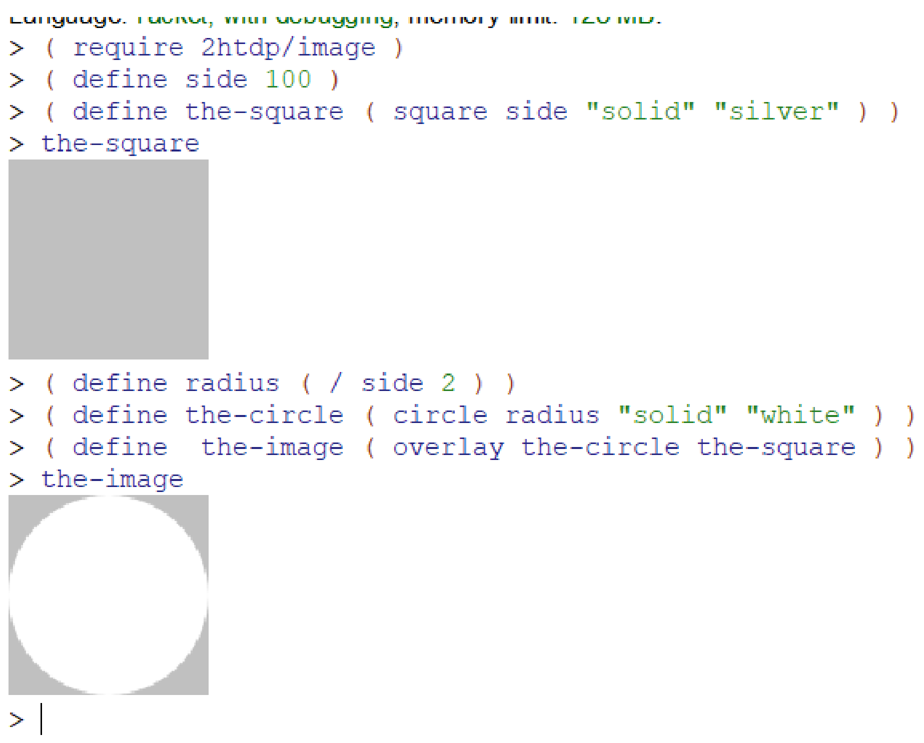
```
> 5
5
> 5.3
5.3
> ( * 3 10 )
30
> ( + ( * 3 10 ) 4 )
34
> ( * 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 )
12157665459056928801
>
```

## Interaction: Solution to the Scrap Problem



```
> pi
3.141592653589793
> side
  side: undefined;
 cannot reference an identifier before its definition
> ( define side 100 )
> side
100
> ( define square-area ( * side side ) )
> square-area
10000
> ( define radius ( / side 2 ) )
> radius
50
> ( define circle-area ( * pi radius radius ) )
> circle-area
7853.981633974483
> ( define scrap-area ( - square-area circle-area ) )
> scrap-area
2146.018366025517
>
```

## Interaction: Illustration of Scrap Problem Situation



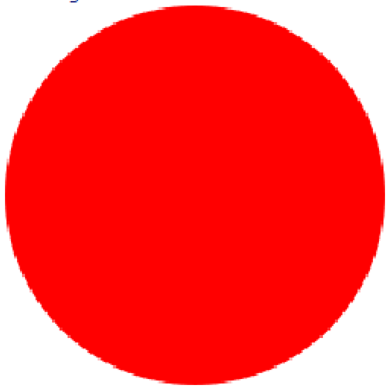
```
Language: racket, with debugging; memory limit: 128 MB.
> ( require 2htdp/image )
> ( define side 100 )
> ( define the-square ( square side "solid" "silver" ) )
> the-square
```

```
> ( define radius ( / side 2 ) )
> ( define the-circle ( circle radius "solid" "white" ) )
> ( define  the-image ( overlay the-circle the-square ) )
> the-image
```

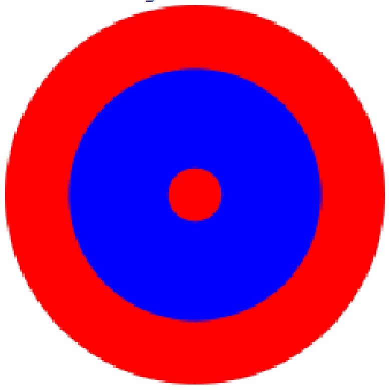
```
> |
```

**Interaction: Illustration of the Target Problem Situation**

```
> (require 2htdp/image)
> (define radius 100)
> (define big-red (circle radius "solid" "red") )
> big-red
```

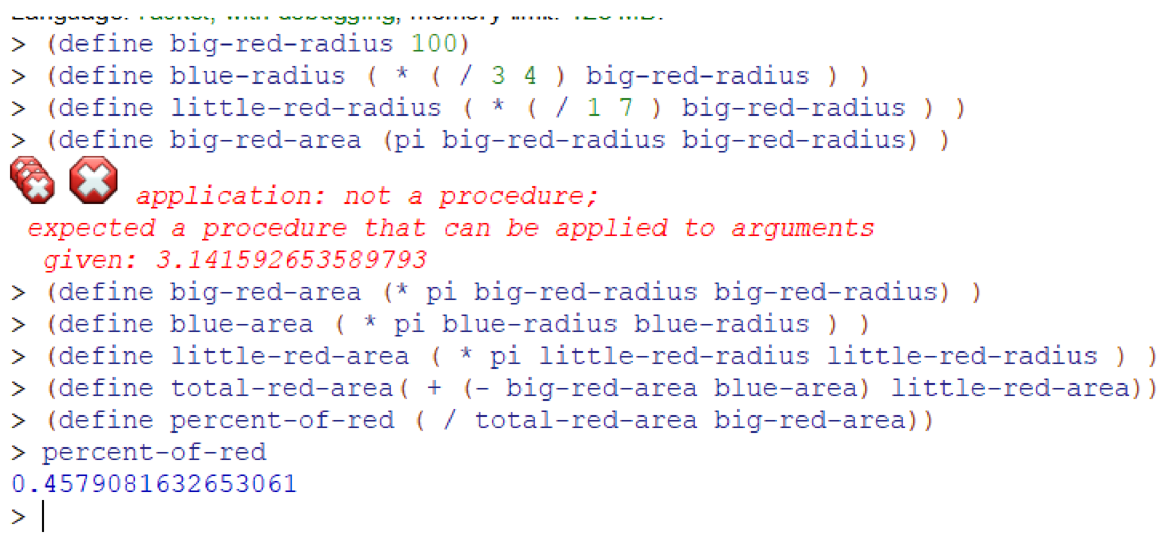


```
> ( define blue-radius ( * ( / 2 3) radius ) )
> ( define blue-buddy ( circle blue-radius "solid" "blue" ) )
> (define little-radius ( * ( / 1 7 ) radius ) )
> ( define little-red ( circle little-radius "solid" "red" ) )
> ( define the-target ( overlay little-red ( overlay blue-buddy big-red ) ) )
> the target
     the: undefined;
 cannot reference an identifier before its definition
> the-target
```



```
>
```

## Interaction: Solution to Target Problem



```
Language: racket, with debugging; memory limit: 128 MB.
> (define big-red-radius 100)
> (define blue-radius ( * ( / 3 4 ) big-red-radius ) )
> (define little-red-radius ( * ( / 1 7 ) big-red-radius ) )
> (define big-red-area (pi big-red-radius big-red-radius) )
   application: not a procedure;
 expected a procedure that can be applied to arguments
  given: 3.141592653589793
> (define big-red-area (* pi big-red-radius big-red-radius) )
> (define blue-area ( * pi blue-radius blue-radius ) )
> (define little-red-area ( * pi little-red-radius little-red-radius ) )
> (define total-red-area( + (- big-red-area blue-area) little-red-area))
> (define percent-of-red ( / total-red-area big-red-area))
> percent-of-red
0.4579081632653061
> 
```