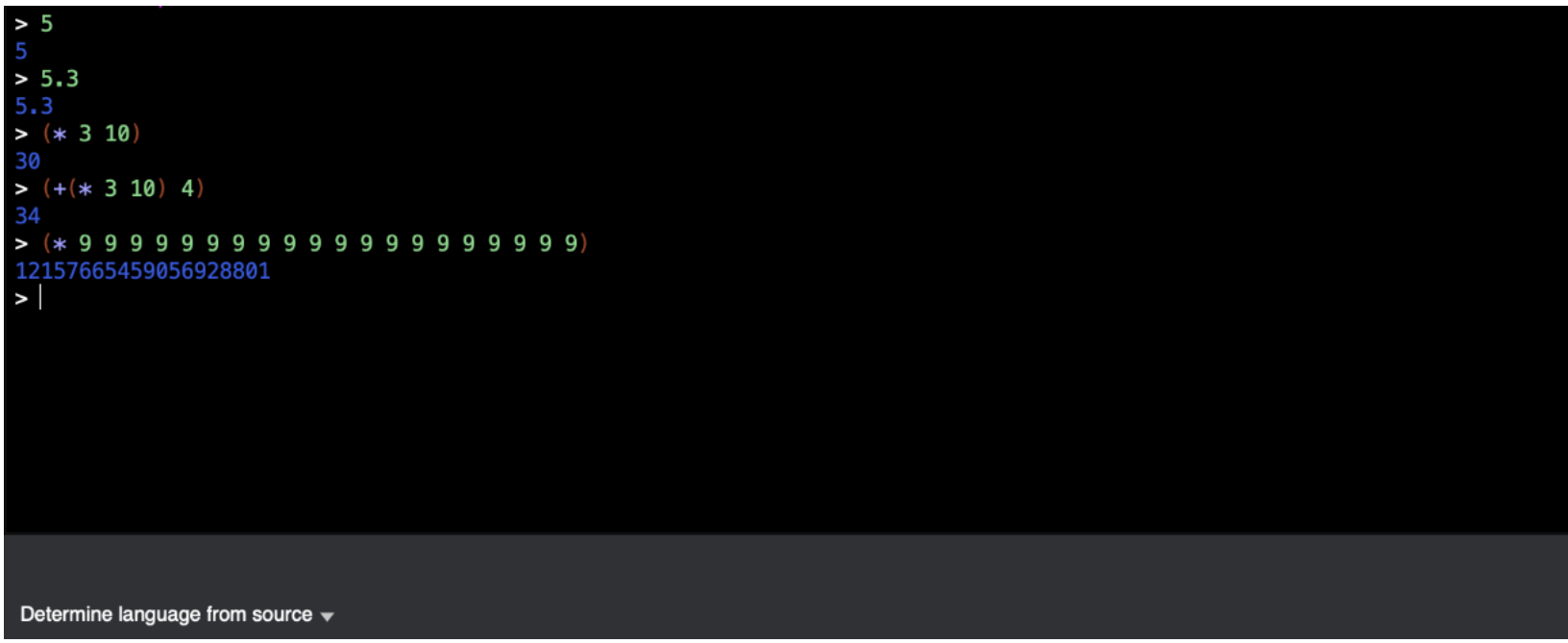
Introduction to Racket/DrRacket

Learning Abstract:

This assignment was completed to have a better understanding of Racket by mimicking tasks done by the professor. A small simple problem was also done to ultimately become familiar with this new programming language.

Tasks:

Interactions: Simple Numeric Processing

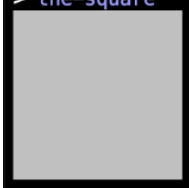


Solution to the Scrap Problem

```
> 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
>
```

Illustration of Scrap Problem Situation

```
> ( require 2htdp/image)
> (define side 100)
> (define the-square (square side "solid" "silver"))
> the-square
```



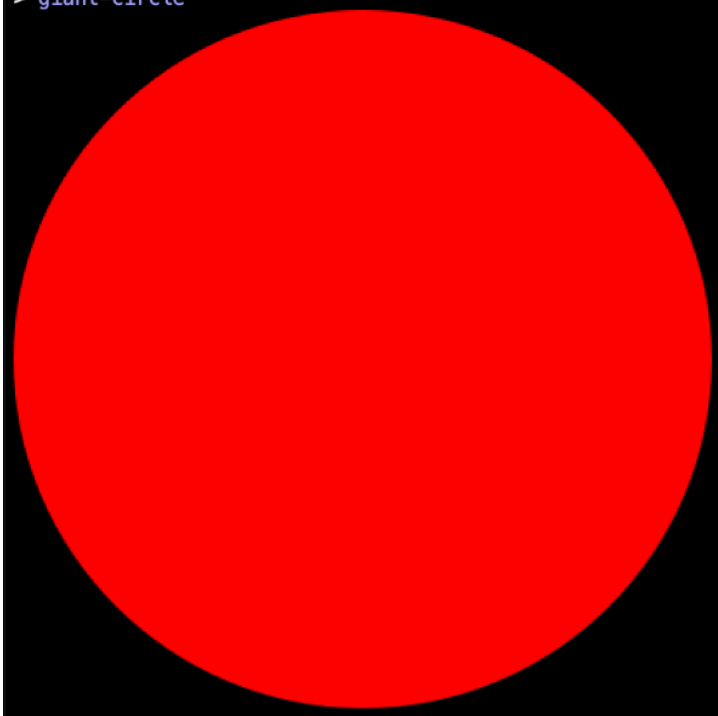
```
> (define radius ( / side 2))
> (define the-circle (circle radius "solid" "white"))
> (define the-image (overlay the-circle the-square))
> the-image
```

```
>
```

Determine language from source

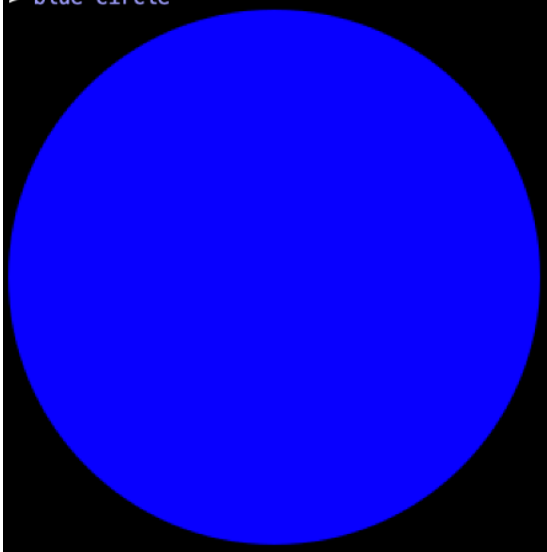
Illustration of the Target Problem

```
> (require 2htdp/image)
> (define giantDiameter 200)
> (define giant-circle(circle giantDiameter "solid" "red"))
> giant-circle
```



```
>
```

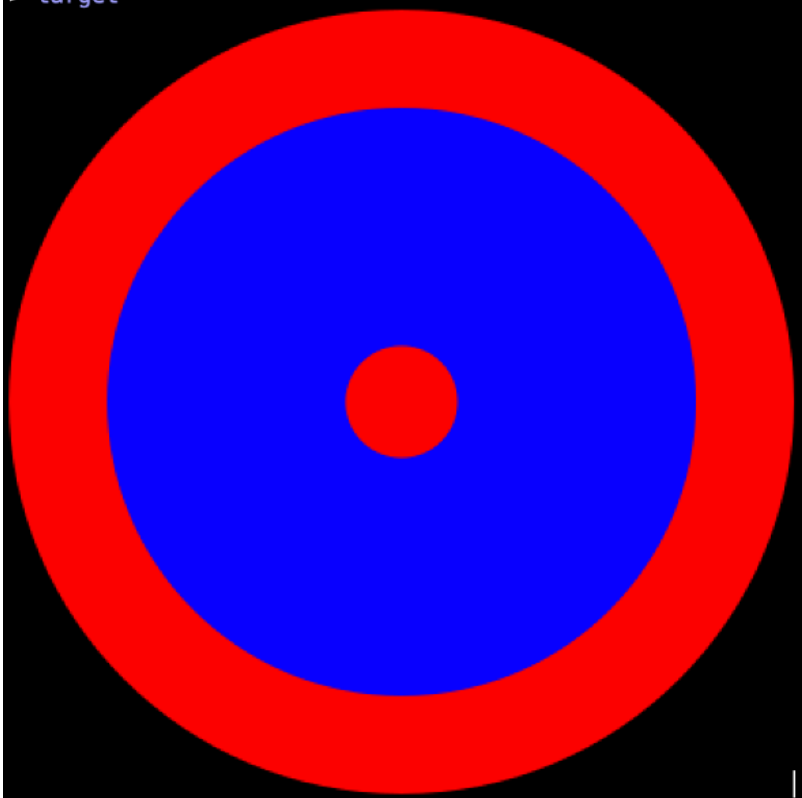
```
> (define blue-diameter ( * giantDiameter 3( / 4)))
> blue-diameter
150
> (define blue-circle (circle blue-diameter "solid" "blue"))
> blue-circle
```



```
> (define small-diameter (/ ( * giantDiameter 1) 7))
> small-diameter
28 4/7
> (define small-circle (circle small-diameter "solid" "red"))
> small-circle
```

```
>
```

```
> (define target (overlay small-circle blue-circle giant-circle))
> target
```



Solution to the Target Problem

```
> giantDiameter
200
> blue-diameter
150
> small-diameter
28 4/7
> (define big-radius( / giantDiameter 2))
> (define giantCircle-area( * pi big-radius big-radius))
> giantCircle-area
31415.926535897932
> (define blue-radius (/ blue-diameter 2))
> blue-radius
75
> (define blue-area( * pi blue-radius blue-radius))
> blue-area
17671.458676442588
> (define small-radius (/ small-diameter 2))
> (define small-area ( * pi small-radius small-radius))
> small-area
641.141357875468
> (define red-area( + ( - giantCircle-area blue-area) small-area))
> red-area
14385.609217330812
> (define red-percent ( * ( / red-area giantCircle-area) 100))
> red-percent
45.79081632653061
>
```