LEARNING ABSTRACT

FOR THIS ASSIGNMENT I USED DRRACKET TO DEMONSTRATE SOME OF THE CAPABILITIES OF THE LISP PROGRAMMING LANGUAGE. I DEMONSTRATED SOME OF THE MATHEMATIC COMPUTATIONS THAT CAN BE PERFORMED. I USED THE GRAPHICS LIBRARY TO CREATE AND OVERLAY IMAGES. USING THESE SKILLS, I SOLVED THE TARGET PROBLEM AND CONSTRUCTED AN IMAGE OF THE TARGET.

INTERACTION: SIMPLE NUMERIC PROCESSING

```
#lang racket
(define ( hello reference )
(display (string-append "Hello there, " reference "!\n") )
)
```

```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (hello "Justin Stooks")
Hello there, Justin Stooks!
> 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 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
        circle_area: undefined;
 cannot reference an identifier before its definition
> circle-area
7853.981633974483
> (define scrap-area (- square-area circle-area))
> scrap-area

> scrap-area
2146.018366025517
```
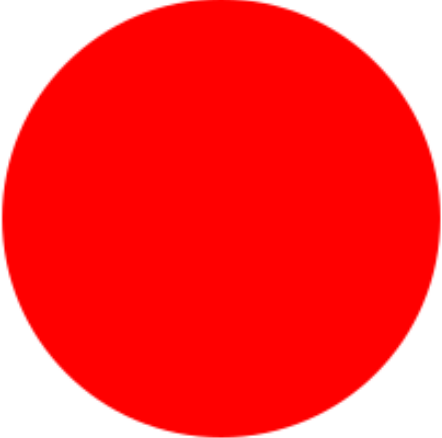
Interaction: Illustration of Scrap Problem Situation

```
> (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
```

```
>
```

Interaction: Illustration of the Target Problem Situation

```
> (define side 200)
>(define radius (/ side 2))
>(define the-large-circle (circle radius "solid" "red"))
> the-large-circle

#<procedure:>>
#<procedure:>>
#<procedure:>>
```



```
> (define side-m (* (/ 3 4) 200))
> (define radius-m (/ side-m 2))
> (define the-med-circle (circle radius-m "solid" "blue"))
> the-med-circle

#<procedure:>>
```

```
> (define radius-m (/ side-m 2))
> (define the-med-circle (circle radius-m "solid" "blue"))
> the-med-circle

#<procedure:>>
#<procedure:>>
#<procedure:>>
```



```
> (define side-s (* (/ 1 7) 200))
> (define radius-s (/ side-s 2))
> (define the-small-circle (circle radius-s "solid" "red"))
> the-small-circle

#<procedure:>>
#<procedure:>>
#<procedure:>>
```
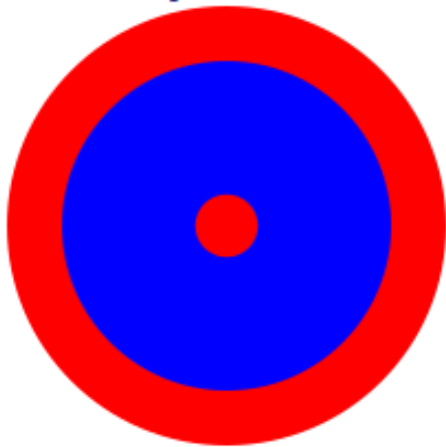
```
> (define radius-s (/ side-s 2))
> (define the-small-circle (circle radius-s "solid" "red"))
> the-small-circle

#<procedure:>>
#<procedure:>>
#<procedure:>>
```



```
> (define the-image (overlay the-small-circle the-med-circle
the-large-circle))
> the-image
```



```
>
```

Interaction: Solution to Target Problem

```
 expected a procedure that can be applied to arguments
  given: 3.141592653589793
> (define area-large (* pi radius radius))
> area-large
#<procedure:>>
31415.926535897932
> (define area-medium (* pi radius-m radius-m))
> area-medium
#<procedure:>>
17671.458676442588
> (define area-small (* pi radius-s radius-s))
> area-small
#<procedure:>>
641.141357875468
> (define area-red (- (+ area-large area-small) area-medium))
> area-red
#<procedure:>>
14385.609217330813
> (define decimal-red (/ area-red area-large))
> decimal-red
#<procedure:>>
0.45790816326530615
> (define percent-red (* decimal-red 100))
> percent-red
#<procedure:>>
45.79081632653062
> |
```