Racket Programming Assignment # 1: First Interactions

Abstract: The work done in this assignment allowed me to get more familiar with Racket Programming. The assignment was done with the interactive interpreter of DrRacket. The first section featured simple processing involving numbers while the second and third sections involved finding the area of square and a circle. The latter section focusing on finding the area visually.

Interaction: Simple Numeric Processing

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> 55
55
> 55.2
55.2
> pi
3.141592653589793
> ( * 3 8)
24
> ( + ( * 3 8 ) 6 )
30
> ( expt 2 8 )
256
> ( * pi ( expt 7 2 ) )
153.93804002589985
> ( expt 9 50 )
515377520732011331036461129765621272702107522001
> x
x: undefined;
 cannot reference an identifier before its definition
>
```

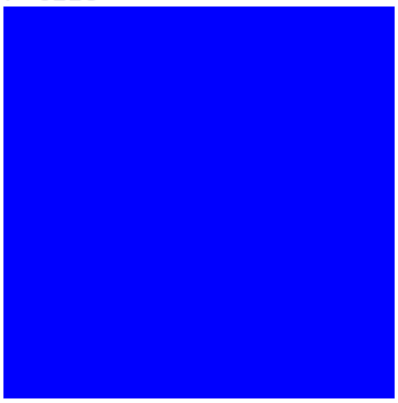
Interaction: Solution to the blue and red tile area problem

The blue and red tile area problem: A tile of side 200 is blue, except for a centered red disk of radius one-third the side of the tile. What is the area of the tile which is blue?

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( define side-of-tile 200)
> ( define diameter-of-dot ( / side-of-tile 3 ) )
> ( define radius-of-dot ( / diameter-of-dot 2 ) )
> ( define total-tile-area ( expt side-of-tile 2 ) )
> ( define red-dot-area ( * pi ( expt radius-of-dot 2 ) ) )
> ( define blue-tile-area ( - total-tile-area red-dot-area ) )
> side-of-tile
200
> diameter-of-dot
66 2/3
> radius-of-dot
33 1/3
> total-tile-area
40000
> red-dot-area
3490.658503988659
> blue-tile-area
36509.341496011344
>
```

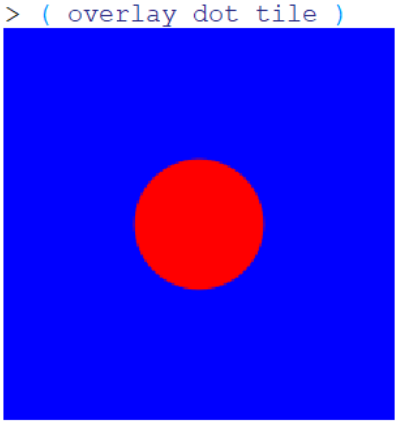
Interaction: Painting the blue and red tile

```
> tile
```



```
> ( define dot ( circle radius-of-dot "solid" "red" ) )
> dot
```

```
> ( overlay dot tile )
```

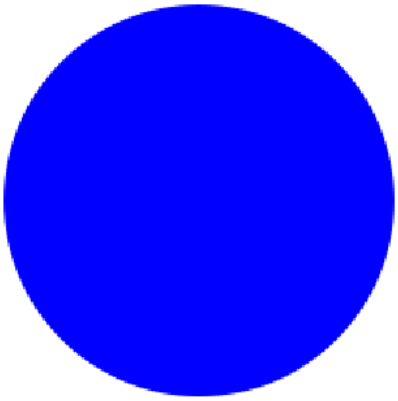


```
>
```

Interaction: Painting the blue and red concentric disks image

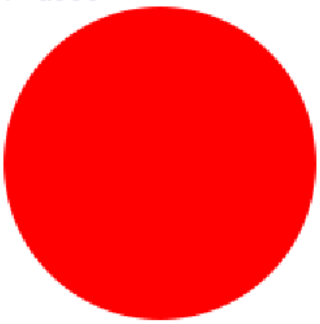
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.

```
> ( require 2htdp/image )
> ( define bottom-dot-diameter 200)
> ( define bottom-dot-radius ( / bottom-dot-diameter 2) )
> ( define dot4-radius ( - bottom-dot-radius 20 ) )
> ( define dot3-radius ( - dot4-radius 20 ) )
> ( define dot2-radius ( - dot3-radius 20 ) )
> (define top-dot-radius ( - dot2-radius 20 ) )
> bottom-dot-radius
100
> dot4-radius
80
> dot3-radius
60
> dot2-radius
40
> top-dot-radius
20
> ( define bottom-dot ( circle bottom-dot-radius "solid" "blue" ) )
> bottom-dot
```



```
> ( define dot4 ( circle dot4-radius "solid" "red" ) )
> dot4
```

```
> dot4
```



```
> (define dot3 ( circle dot3-radius "solid" "blue" ) )
> dot3
```

```
> ( define dot2 ( circle dot2-radius "solid" "red" ) )
> dot 2
```

```
 dot: undefined;
 cannot reference an identifier before its definition
> dot2
```

```
> ( define top-dot ( circle top-dot-radius "solid" "blue" ) )
> top-dot
```

Interaction: Computing the area of the concentric disks image which is blue

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( define bottom-dot-diameter 200)
> ( define bottom-dot-radius ( / bottom-dot-diameter 2) )
> ( define dot4-radius ( - bottom-dot-radius 20 ) )
> ( define dot3-radius ( - dot4-radius 20 ) )
> ( define dot2-radius ( - dot3-radius 20 ) )
> (define top-dot-radius ( - dot2-radius 20 ) )
> bottom-dot-radius
100
> dot4-radius
80
> dot3-radius
60
> dot2-radius
40
> top-dot-radius
20
> ( define bottom-dot-area ( * pi ( expt bottom-dot-radius 2 ) ) )
> ( define dot4-area ( * pi ( expt dot4-radius 2 ) ) )
> ( define dot3-area ( * pi ( expt dot3-radius 2 ) ) )
> ( define dot2-area ( * pi ( expt dot2-radius 2 ) ) )
> ( define top-dot-area ( * pi ( expt top-dot-radius 2 ) ) )
> bottom-dot-area
31415.926535897932
> dot4-area
20106.192982974677
> dot3-area
11309.733552923255
> dot2-area
5026.548245743669
> top-dot-area
1256.6370614359173
> ( define blue-layer1-area (- bottom-dot-area dot4-area ) )
> ( define blue-layer2-area (- dot3-area dot2-area ) )
> ( define blue-layer3-area top-dot-area )
> blue-layer1-area
```

```
> dot4-radius
80
> dot3-radius
60
> dot2-radius
40
> top-dot-radius
20
> ( define bottom-dot-area ( * pi ( expt bottom-dot-radius 2 ) ) )
> ( define dot4-area ( * pi ( expt dot4-radius 2 ) ) )
> ( define dot3-area ( * pi ( expt dot3-radius 2 ) ) )
> ( define dot2-area ( * pi ( expt dot2-radius 2 ) ) )
> ( define top-dot-area ( * pi ( expt top-dot-radius 2 ) ) )
> bottom-dot-area
31415.926535897932
> dot4-area
20106.192982974677
> dot3-area
11309.733552923255
> dot2-area
5026.548245743669
> top-dot-area
1256.6370614359173
> ( define blue-layer1-area (- bottom-dot-area dot4-area ) )
> ( define blue-layer2-area (- dot3-area dot2-area ) )
> ( define blue-layer3-area top-dot-area )
> blue-layer1-area
11309.733552923255
> blue-layer2-area
6283.185307179586
> blue-layer3-area
1256.6370614359173
> ( define total-blue-area (+ blue-layer1-area blue-layer2-area
blue-layer3-area ) )
> total-blue-area
18849.55592153876
>
```