

The first part of this assignment was to copy what was shown to us on Racket and to understand the syntax of what was shown.

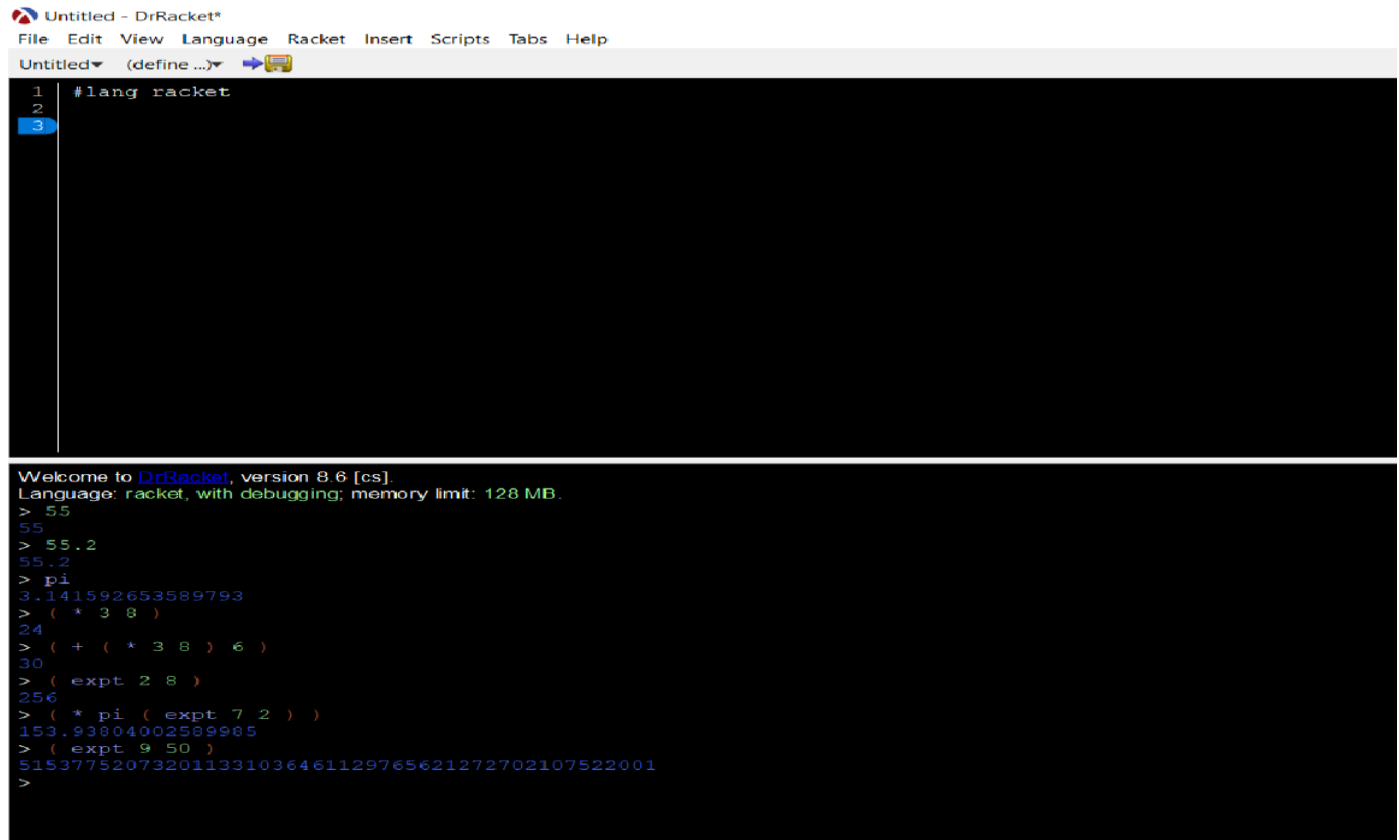
The second part of this assignment was to again copy what was shown but this time we introduced to functions/definitions. We defined the size of a square tile which we then used as a base to compute the sizes, radius', diameters, and areas of other defined shapes.

The third part of this assignment introduces imaging. We painted a red dot and blue tile and then used a built-in function to overlay the red dot on the blue tile.

The fourth part requires us to paint a target with the skills we learned and used in the last three parts. We were given a base radius of 200 which I then used to compute the rest of the circles.

The final part asked us to calculate the area of the blue circles. To accomplish this, I defined each individual area, then added up all the red and blue areas together respectively, then subtracted the red and blue areas to get my final answer.

## Task A: Numeric Computations



The screenshot shows the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The title bar reads "Untitled - DrRacket\*". The editor window contains a Racket script with three lines: `1 #lang racket`, `2`, and `3` (which is highlighted with a blue selection bar). Below the editor is a console window displaying the following text:

```
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
>
```

## Task B: Solution to the area problem

```

1 #lang racket
2 | define side-of-tile 200 |
3 | define diameter-of-dot (/ side-of-tile 2) |
4 | define radius-of-dot (/ diameter-of-dot 2) |
5 | define total-tile-area (* side-of-tile 2) |
6 | define red-dot-area (* pi (* radius-of-dot 2)) |
7 | define blue-tile-area (- total-tile-area red-dot-area) |




Welcome to Racket, version 8.6 [cs]
Language: racket, with debugging, memory limit 128 MB.
> side-of-tile
200
> diameter-of-dot
100
> radius-of-dot
50
> total-tile-area
40000
> red-dot-area
31415.92653589793
> blue-tile-area
39685.80346410207
>

```

## Task C: Illustration of the area problem situation

```

1 #lang racket
2 | require http/image |
3 | define side-of-tile 200 |
4 | define diameter-of-dot (/ side-of-tile 2) |
5 | define radius-of-dot (/ diameter-of-dot 2) |
6 | define tile (square side-of-tile "solid" "blue") |
7 | define dot (circle radius-of-dot "solid" "red") |

Welcome to Racket, version 8.6 [cs]
Language: racket, with debugging, memory limit 128 MB.
> tile

> dot

> (overlay dot tile)

>

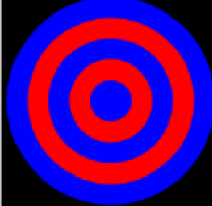
```

## Task D: Circles Image

```
Untitled - DrRacket®
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define -) [icon] Check Syntax [icon]

1 #lang racket
2 (require 2htdp/image)
3 (define disk1 100)
4 (define disk2 (- disk1 20))
5 (define disk3 (- disk2 20))
6 (define disk4 (- disk3 20))
7 (define disk5 (- disk4 20))
8
9 (define circle1 (circle disk1 "solid" "blue"))
10 (define circle2 (circle disk2 "solid" "red"))
11 (define circle3 (circle disk3 "solid" "blue"))
12 (define circle4 (circle disk4 "solid" "red"))
13 (define circle5 (circle disk5 "solid" "blue"))
14
15
```

Welcome to DrRacket, version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.  
> (overlay circle5 circle4 circle3 circle2 circle1)



> |

## Task E: Compute area of circle image

```
1 #lang racket
2 (require 2htdp/image)
3 (define disk1 100)
4 (define disk2 (- disk1 20))
5 (define disk3 (- disk2 20))
6 (define disk4 (- disk3 20))
7 (define disk5 (- disk4 20))
8
9 (define circle1 (circle disk1 "solid" "blue"))
10 (define circle2 (circle disk2 "solid" "red"))
11 (define circle3 (circle disk3 "solid" "blue"))
12 (define circle4 (circle disk4 "solid" "red"))
13 (define circle5 (circle disk5 "solid" "blue"))
14
15 (define smallblue (* pi (expt disk5 2)))
16 (define smallred (* pi (expt disk4 2)))
17 (define midblue (* pi (expt disk3 2)))
18 (define bigred (* pi (expt disk2 2)))
19 (define bigblue (* pi (expt disk1 2)))
20
21 (define smallredarea (- smallred smallblue))
22 (define midbluearea (- midblue smallred))
23 (define bigredarea (- bigred midblue))
24 (define bigbluearea (- bigblue bigred))
25
26 (define totalredarea (+ smallredarea bigredarea))
27 (define totalbluearea (+ smallblue midbluearea bigbluearea))
28 (define bluearea (- totalbluearea totalredarea))
29
```

Welcome to DrRacket, version 8.6 [cs].  
Language: racket, with debugging; memory limit: 128 MB.  
> smallblue  
1256.4370454359173  
> smallred  
5026.549285783449  
> midblue  
11309.733552921255  
> bigred  
20286.192362974473  
> bigblue  
31415.926535897932  
> smallredarea  
3770.1122403475317  
> midbluearea  
4283.115577175586  
> bigredarea  
2794.450435051422  
> bigbluearea  
11309.733552921255  
> totalredarea  
12566.170834359173  
> totalbluearea  
18845.53552153879  
> bluearea  
4283.185377179589  
>