Racket Programming Assignment #1: First Interactions

Learning Abstract:

This document covers several simple processes that can be accomplished using the Racket programming language. All interactions are simple enough to do within the interactive pane, similar to a command prompt! Part one contains simple numeric calculations and a basic format for calling functions. Part two sets a bunch of variables and does some calculations upon those variables. Part three uses the '2htdp/image' library to create a visual representation of the calculations in part two.

Interaction: Simple Numeric Processing

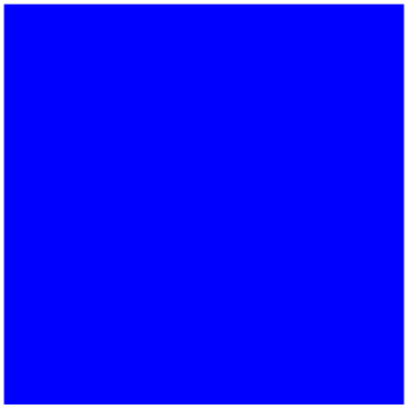
```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> 55
55
> 55.2
55.2
> pi
3.141592653589793
> x
x: undefined;
 cannot reference an identifier before its definition
> ( * 3 8 )
24
> ( +( * 3 8 ) 6 )
30
> ( expt 2 8 )
256
> ( * pi ( expt 7 2 ) )
153.93804002589985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

Interaction: Solution to the blue and red tile area problem

```
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
>
```

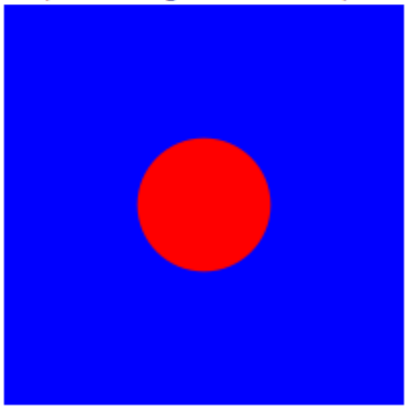
Interaction: Painting the blue and red tile

```
Welcome to DrRacket, version 8.6 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( require 2htdp/image )
> ( define side-of-tile 200 )
> ( define diameter-of-dot ( / side-of-tile 3 ) )
> ( define radius-of-dot ( / diameter-of-dot 2 ) )
> ( define tile ( square side-of-tile "solid" "blue" ) )
> tile
```



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

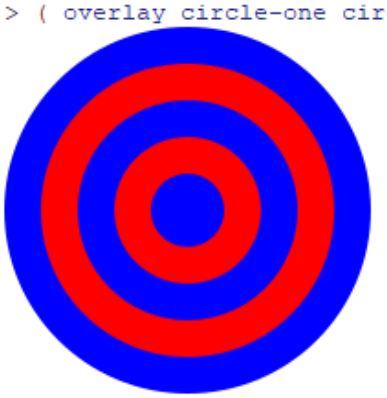
```
> ( overlay dot tile )
```



```
>
```

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 radius-circle-one 20 )
> ( define circle-one ( circle radius-circle-one "solid" "blue" ) )
> ( define radius-circle-two ( * radius-circle-one 2 ) )
> ( define circle-two ( circle radius-circle-two "solid" "red" ) )
> ( define radius-circle-three ( * radius-circle-one 3 ) )
> ( define circle-three ( circle radius-circle-three 'solid 'blue ) )
> ( define radius-circle-four ( * radius-circle-one 4 ) )
> ( define circle-four ( circle radius-circle-four 'solid 'red ) )
> ( define radius-circle-five ( * radius-circle-one 5 ) )
> ( define circle-five ( circle radius-circle-five 'solid 'blue ) )
> ( overlay circle-one circle-two circle-three circle-four circle-five )
```



```
>
```

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 radius-circle-one 20 )
> ( define area-circle-one ( * pi ( expt radius-circle-one 2 ) ) )
> ( define radius-circle-two ( * radius-circle-one 2 ) )
> ( define area-circle-two ( * pi ( expt radius-circle-two 2 ) ) )
> ( define radius-circle-three ( * radius-circle-one 3 ) )
> ( define area-circle-three ( * pi ( expt radius-circle-three 2 ) ) )
> ( define radius-circle-four ( * radius-circle-one 4 ) )
> ( define area-circle-four ( * pi ( expt radius-circle-four 2 ) ) )
> ( define radius-circle-five ( * radius-circle-one 5 ) )
> ( define area-circle-five ( * pi ( expt radius-circle-five 2 ) ) )
> ( + ( - area-circle-five area-circle-four ) ( - area-circle-three area-circle-two) area-circle-one )
18849.55592153876
>
```