
First Racket Programming Assignment Solution

Learning Abstract

In this assignment I learned a little bit about prefix notation in Lisp. I also learned how to bind variables to values. I learned to use a Racket library to create and display shapes so that I could render the problem situations graphically. All of this took place within the Interactions pane of the DrRacket PDE.

Interaction: Simple Numeric Processing

```
Untitled 2 - DrRacket
File Edit View Language Racket Insert Scripts Tabs Help
Untitled 2 (define ...) Check Syntax Debug Macro Stepper Run Stop

#lang racket

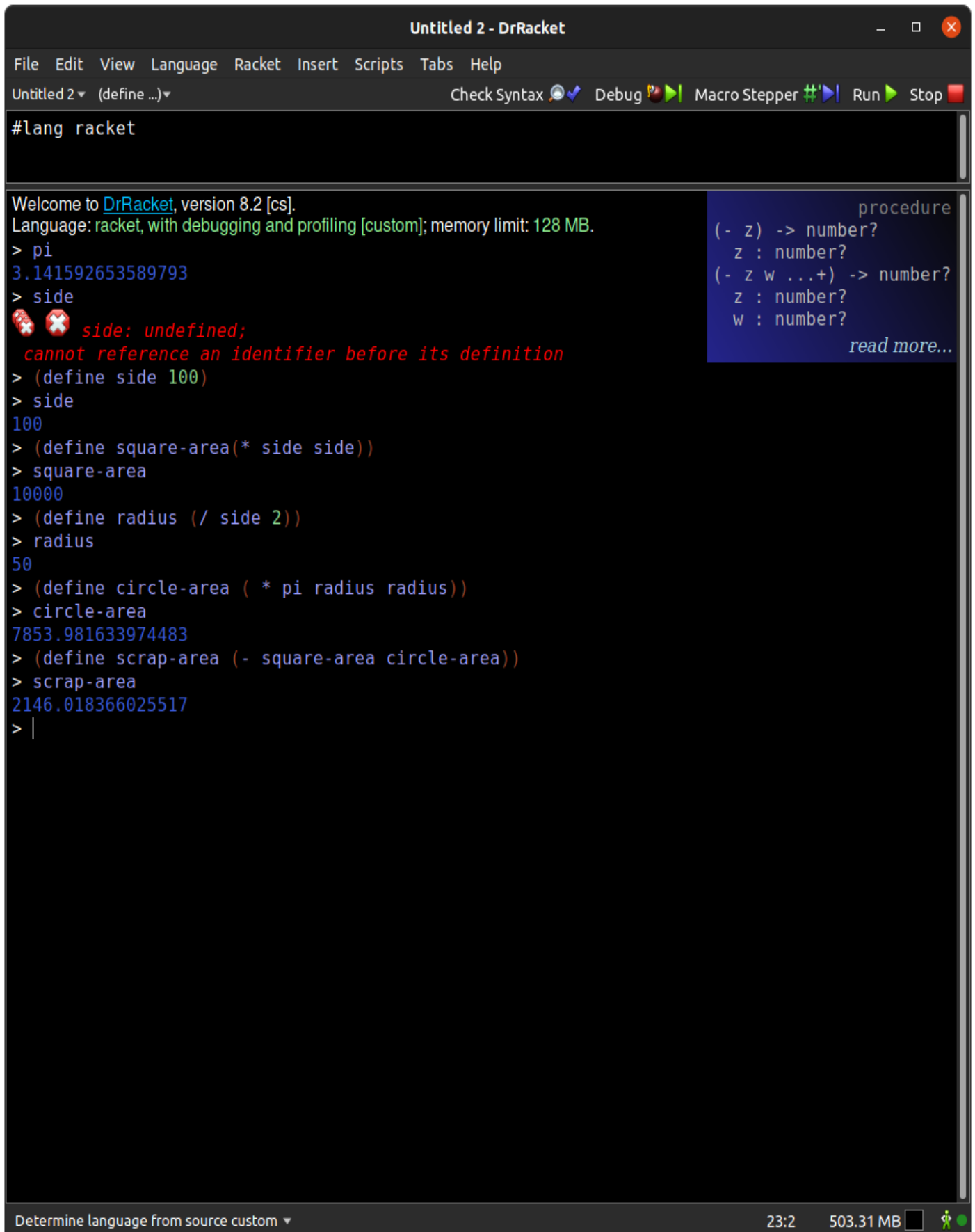
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging and profiling [custom]; memory limit: 128 MB.
> 5
5
> 5.3
5.3
> (* 3 10)
30
> (+ (* 3 10) 4)
34
> (* 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9)
12157665459056928801
>

procedure
(* z ...) -> number?
z : number?
read more...
```

Determine language from source custom 13:2 512.96 MB

Interaction: Solution to the Scrap Problem

The Scrap Problem: A circular disk of maximal size is cut from a square piece of tin of side 100 units. What is the area of the scrap?



```
Untitled 2 - DrRacket
File Edit View Language Racket Insert Scripts Tabs Help
Untitled 2 (define ...) Check Syntax Debug Macro Stepper Run Stop
#lang racket

Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging and profiling [custom]; memory limit: 128 MB.
> 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
> |

procedure
(- z) -> number?
z : number?
(- z w ...) -> number?
z : number?
w : number?
read more...
```

Determine language from source custom 23:2 503.31 MB

Interaction: Illustration of Scrap Problem Situation

Untitled 2 - DrRacket

File Edit View Language Racket Insert Scripts Tabs Help

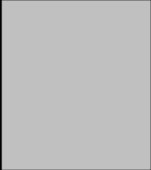
Untitled 2 ▾ (define ...) ▾

Check Syntax Debug Macro Stepper Run Stop


```
#lang racket
```

Welcome to [DrRacket](#), version 8.2 [cs].
Language: racket, with debugging and profiling [custom]; memory limit: 128 M

```
> (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
```



```
> |
```

```
(define id expr)                                syntax
(define (head args) body ...+)

head = id
  | (head args)

args = arg ...
  | arg ... . rest-id

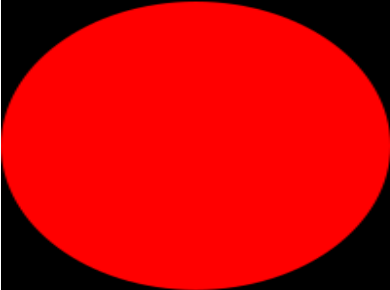
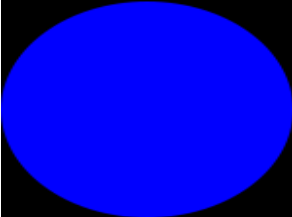
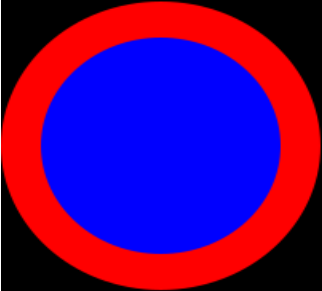
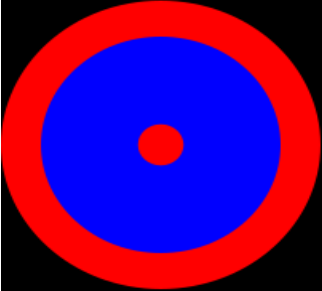
arg = arg-id
  | [arg-id default-expr]
  | keyword arg-id
  | keyword [arg-id default-expr]
                                     read more...
```

Determine language from source custom ▾

13:2 522.35 MB

Interaction: Illustration of the Target Problem Situation

A “target” consists of a red disc of some diameter, containing a blue disc of diameter $\frac{3}{4}$ that of the bigger disc, which, in turn, contains another red disc, this one of diameter $\frac{1}{7}$ that of the biggest disc.

```
Welcome to DrRacket, version 8.2 [cs].
Language: Determine language from source [custom]; memory limit: 128 MB.
> (require 2htdp/image)
> (define radius 100)
> (define the-circle (circle radius "solid" "red"))
> the-circle

> (define b-radius (* radius (/ 3 4)))
> (define the-b-circle(circle b-radius "solid" "blue"))

❗ ❗ application: not a procedure;
expected a procedure that can be applied to arguments
given: 75
> > (define the-b-circle(circle b-radius "solid" "blue"))
#<procedure:>>
> the-b-circle

> (define two-circle (overlay the-b-circle the-circle))
> two-circle

> (define c-radius (* radius (/ 1 7)))
> (define the-c-circle (circle c-radius "solid" "red"))
> (define three-circle (overlay the-c-circle two-circle))
> three-circle
>
```

Interaction: Solution to Target Problem

What percentage of the target is red?

```
#lang racket
```

```
Welcome to DrRacket, version 8.2 [cs].
```

```
Language: racket, with debugging and profiling [custom]; memory limit: 128 MB.
```

```
> (define red-disc-diameter 100)
```

```
> (define red-disc-area (* pi (* (/ red-disc-diameter 2) (/ red-disc-diameter 2))))
```

```
> red-disc-area
```

```
7853.981633974483
```

```
> (define blue-disc-diameter (* red-disc-diameter (/ 3 4)))
```

```
> blue-disc-diameter
```

```
75
```

```
> (define blue-disc-area (* pi (* (/ blue-disc-diameter 2) (/ blue-disc-diameter 2))))
```

```
> blue-disc-area
```

```
4417.864669110647
```

```
> (define small-red-diameter (* red-disc-diameter (/ 1 7)))
```

```
> small-red-diameter
```

```
 $14\frac{2}{7}$ 
```

```
> (define small-red-area (* pi (* (/ small-red-diameter 2) (/ small-red-diameter 2))))
```

```
> small-red-area
```

```
160.285339468867
```

```
> (define percentage-of-red (* (/ (- (+ red-disc-area small-red-area) blue-disc-area )
```

```
red-disc-area) 100))
```

```
> percentage-of-red
```

```
45.79081632653062
```

```
> |
```