

Getting Started with Scheme with Dr Racket

1. We will use Dr Racket software. It is available in JD 2214 or JD 2217.
2. You can easily download and install Dr Racket from www.racket-lang.org
 - a. Once installed, open up Dr Racket. Click tab Choose Language. Select "Use language destination in source". Your source files should automatically have as the first line `#lang racket`
3. Under Dr Racket Help tab you can get Racket documentation including The Racket Guide and The Racket Reference. These are both guides to Scheme.
4. Open Dr Racket. You should have two windows. The top window (edit window) is for Scheme definitions. The bottom window is the interpreter . [If you see only one window, Choose Racket- Run.
5. Type in some Scheme expressions in the interpreter window to see what happens. Use prefix notation. Create expressions that give errors.

```
( * 3 4 ( + 5 6 100 ) )  
( min 2 9 16 -4 )  
'( + 5 7 9 )  
(define a 23)  
(define b 2.3)  
(- a b)  
( / 3 5 7 )  
(sqrt -7)
```

6. Next create a function definition in the edit window.

```
( define range  
  ( lambda (a b c)  
    ( - ( max a b c ) ( min a b c ) )  
  )  
)
```

Save file under name *gettingStarted* . The .rkt suffix will be added.

Run file with Racket – Run

Go to Interpreter window to test program. Note use of prefix notation.

```
>(range 3 8 2)
```

7. Using global variables. Type in edit window and then run.

```
(define a 20)
(define addUp (lambda (x) (+ x a)))

> (addUp 4) => 24
```

You can also put (addUp 4) in edit window and then Run. The answer 24 appears in the interpreter window.

8. Recursion Example. Also try built-in (gcd x y z ..)
; Find greatest common divisor of a and b where a, b >= 0

```
(define gcd1
  (lambda (a b)
    (if (= b 0)
        a
        (gcd1 b (remainder a b) )
    )))

> (gcd1 (* 7 5 3 19) (* 29 31 63 97))
21
```

9. Save all definitions before exiting Dr Racket. On return to Dr Racket you can open a saved definition file or create a new one.

Untitled - DrRacket*

File Edit View Language Racket Insert Tabs Help

Untitled (define ...) Save Check Syntax Debug Macro Stepper Run Stop

```
#lang racket

( define g
  ( lambda (x y)
    ( if ( < x y ) ( + x y ) ( * x y ) )
  )
)

( define quadsolver
  (lambda ( a b c)
    (let*
      ([ disc ( - (* b b) (* 4 a c ) ) ]
       [root1 ( / ( + ( - b ) (sqrt disc)) (* 2 a))]
       [root2 ( / ( - ( - b ) (sqrt disc)) (* 2 a))] )
      (list root1 root2) )))
```

Welcome to [DrRacket](#), version 5.2.1 [3m].
Language: racket; memory limit: 128 MB.
> (quadsolver 2 -11 5)
'(5 $\frac{1}{2}$)
> (quadsolver 2 1 -6)
'($1\frac{1}{2}$ -2)
> (quadsolver 2 1.0 -6)
'(1.5 -2.0)
>

Determine language from source 15:59