

The `Graphics Expression` User Manual

我为 racket 设计了一种专用的绘图 DSL，并且为她取名为`GRAPHICS EXPRESSION`，这是一种树状的 DSL，可以轻易的进行图像的绘制。

- 1.下载并开始使用 **GRAHPICS EXPRESSION (`GRAPHICSEXP.RKT`)**
新建一个 **RACKET** 文件，首先将**#LANG RACKET** 换成 **#LANG S-EXP “GRAPHICSEXP.RKT”**
然后键入 **“HELLO , WORLD”**

1 #lang s-exp "graphicsexpr.rkt"
2 "HELLO , WORLD"

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.
HELLO , WORLD
>

图像透过 **REPL** 显示出来了。

2.PLAY WITH RECTANGLES

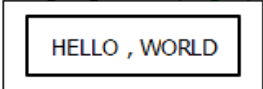
(RECT “HELLO,WORLD”)
1 #lang s-exp "graphicsexpr.rkt"
2 (RECT "HELLO , WORLD")

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.
HELLO , WORLD
>

(RECT (RECT “HELLO,WORLD”))

```
1 #lang s-exp "graphicsexpr.rkt"
2 (RECT (RECT "HELLO , WORLD"))
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.

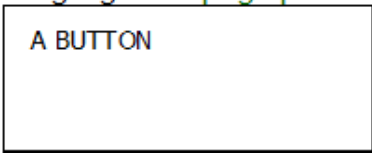


> |

(FIX-RECT 200 80 "A BUTTON")

```
1 #lang s-exp "graphicsexpr.rkt"
2 (FIX-RECT 200 80 "A BUTTON")
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.

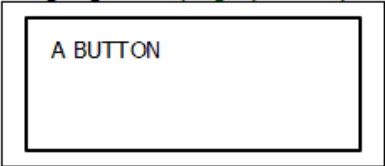


>

(RECT (FIX-RECT 200 80 "A BUTTON"))

```
1 #lang s-exp "graphicsexpr.rkt"
2 (RECT (FIX-RECT 200 80 "A BUTTON"))
```

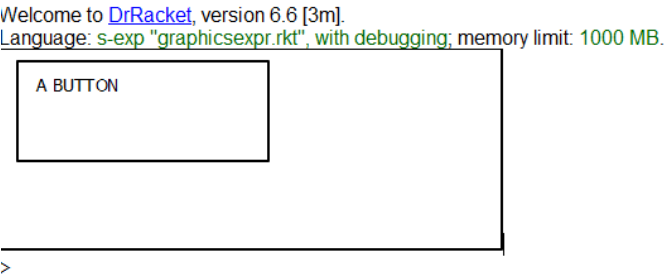
Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.



> |

(FIX-RECT 400 160 (FIX-RECT 200 80 "A BUTTON"))

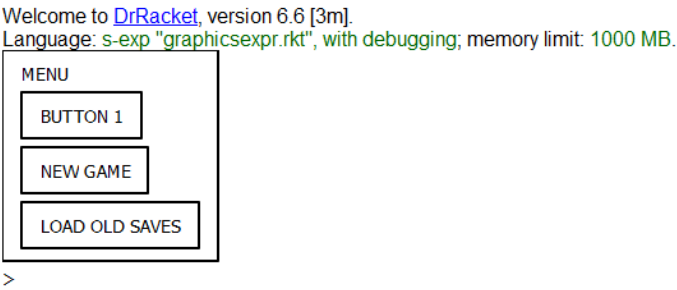
```
1 | #lang s-exp "graphicsexpr.rkt"
2 | (FIX-RECT 400 160 (FIX-RECT 200 80 "A BUTTON"))
```



3.PLAY WITH ALIGNERS

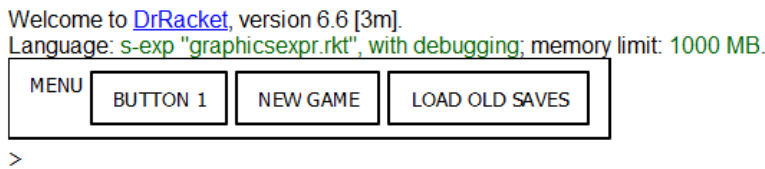
```
(RECT (V "MENU"
        (RECT "BUTTON 1")
        (RECT "NEW GAME")
        (RECT "LOAD OLD SAVES")))
```

```
1 | #lang s-exp "graphicsexpr.rkt"
2 | (RECT (V "MENU"
3 |         (RECT "BUTTON 1")
4 |         (RECT "NEW GAME")
5 |         (RECT "LOAD OLD SAVES")))|
6 |
7 |
```



```
(RECT (H "MENU"
        (RECT "BUTTON 1")
        (RECT "NEW GAME")
        (RECT "LOAD OLD SAVES")))
```

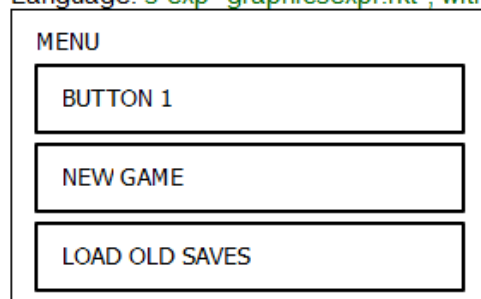
```
1 | #lang s-exp "graphicsexpr.rkt"
2 | (RECT (H "MENU"
3 |         (RECT "BUTTON 1")
4 |         (RECT "NEW GAME")
5 |         (RECT "LOAD OLD SAVES")))|
6 |
7 |
```



```
(RECT (V "MENU"
        (FIX-RECT 250 40 "BUTTON 1")
        (FIX-RECT 250 40 "NEW GAME")
        (FIX-RECT 250 40 "LOAD OLD SAVES")))
```

```
1 | #lang s-exp "graphicsexpr.rkt"
2 | (RECT (V "MENU"
3 |       (FIX-RECT 250 40 "BUTTON 1")
4 |       (FIX-RECT 250 40 "NEW GAME")
5 |       (FIX-RECT 250 40 "LOAD OLD SAVES"))))
6 |
7 |
8 | |
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: `s-exp "graphicsexpr.rkt"`, with debugging; memory limit: 1000 MB.



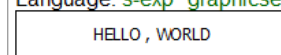
>

4 COMBINE `FIX-RECT` WITH `CENTER`

`(FIX-RECT 200 40 (CENTER-H 200 "HELLO , WORLD"))`

```
test2.rkt (define ...)
1 | #lang s-exp "graphicsexpr.rkt"
2 | (FIX-RECT 200 40 (CENTER-H 200 "HELLO , WORLD"))
3 |
4 |
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: `s-exp "graphicsexpr.rkt"`, with debugging; memory limit: 1000 MB.

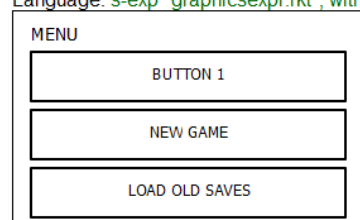


>

```
(RECT (V "MENU"
      (FIX-RECT 250 40 (CENTER-H 250 "BUTTON 1"))
      (FIX-RECT 250 40 (CENTER-H 250 "NEW GAME"))
      (FIX-RECT 250 40 (CENTER-H 250 "LOAD OLD SAVES"))))
```

```
1 | #lang s-exp "graphicsexpr.rkt"
2 | (RECT (V "MENU"
3 |       (FIX-RECT 250 40 (CENTER-H 250 "BUTTON 1"))
4 |       (FIX-RECT 250 40 (CENTER-H 250 "NEW GAME"))
5 |       (FIX-RECT 250 40 (CENTER-H 250 "LOAD OLD SAVES"))))
6 |
7 |
8 |
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: `s-exp "graphicsexpr.rkt"`, with debugging; memory limit: 1000 MB.



>

(FIX-RECT 400 400 "HELLO ,WORLD")

```
1 #lang s-exp "graphicsexpr.rkt"
2 (FIX-RECT 400 400 "HELLO ,WORLD")
3
4
5
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: [s-exp "graphicsexpr.rkt"](#), with [debugging](#); memory limit: 1000 MB.

HELLO ,WORLD

>

(FIX-RECT 400 400 (CENTER-H 400 "HELLO ,WORLD"))

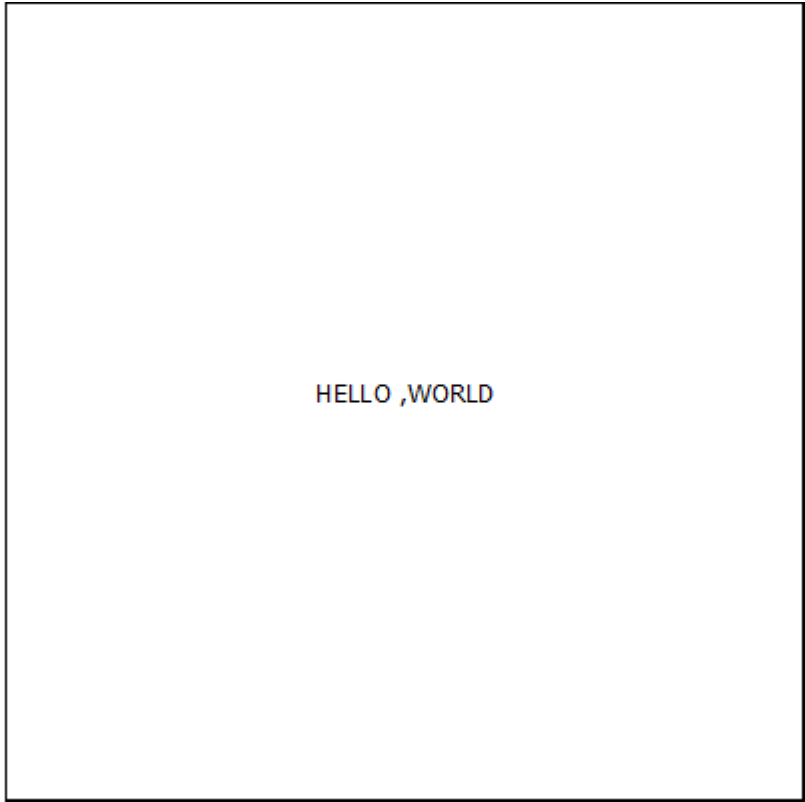
```
1 #lang s-exp "graphicsexpr.rkt"
2 (FIX-RECT 400 400 (CENTER-H 400 "HELLO ,WORLD"))
3
4
5
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: [s-exp "graphicsexpr.rkt"](#), with [debugging](#); memory limit: 1000 MB.

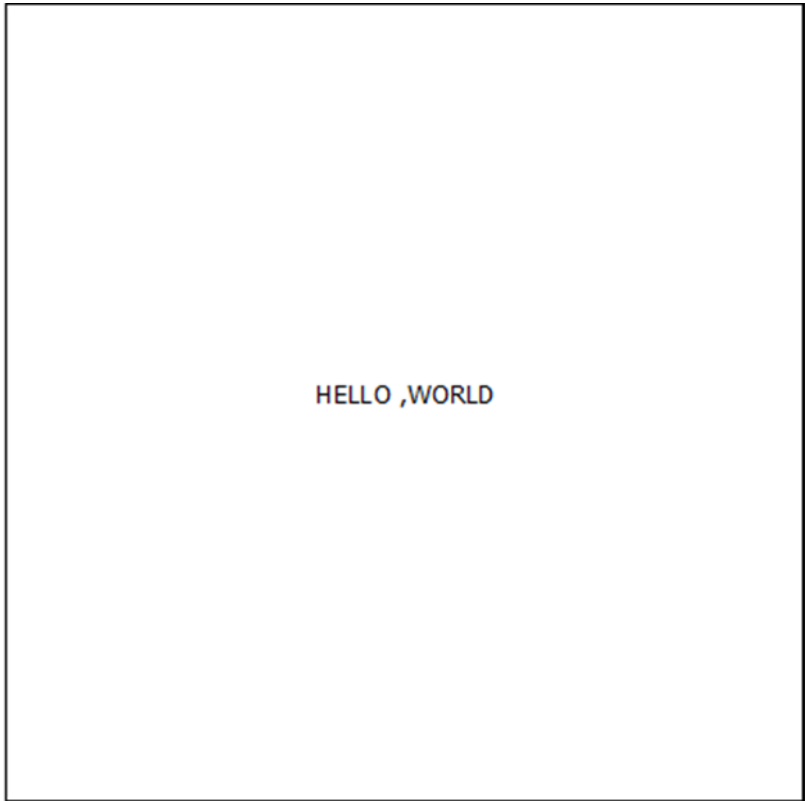
HELLO ,WORLD

>

```
(FIX-RECT 400 400
  (CENTER-H 400 (CENTER-V 400 "HELLO ,WORLD"))))
```



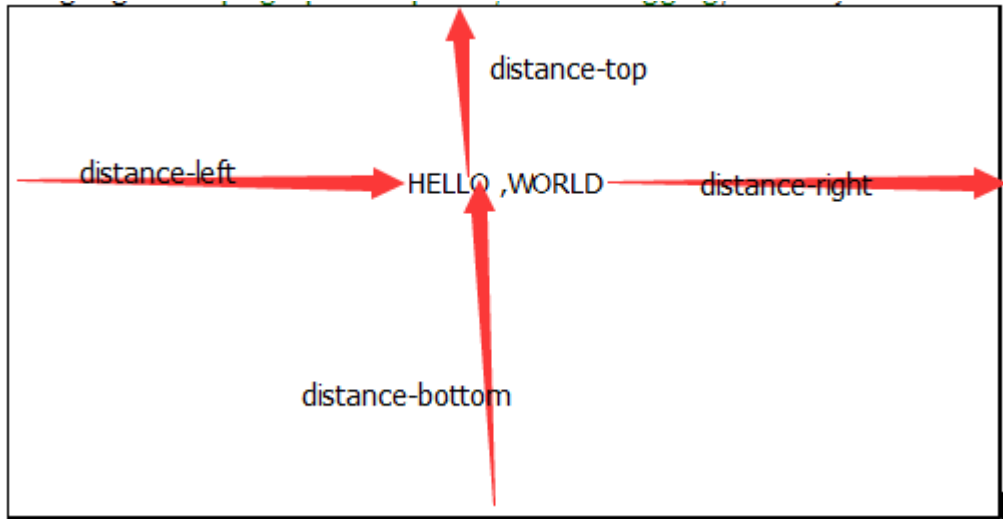
```
(FIX-RECT 400 400
  (CENTER 400 "HELLO ,WORLD"))
```



```
(CENTER X E) = (CENTER-H X (CENTER-V X E))
```

5 PLAY WITH PARAMETERS
TIPS : ALL PARAMETERS MUST BE WRITTEN IN LOWER CASE ,NOT UPPER CASE.

PARAMETER : DISTANCE-TOP ,DISTANCE-BOTTOM,DISTANCE-LEFT&DISTANCE-RIGHT



```
(PARAMETERIZE ([DISTANCE-TOP 80]
                [DISTANCE-LEFT 200]
                [DISTANCE-BOTTOM 160]
                [DISTANCE-RIGHT 200]))
(RECT "HELLO ,WORLD"))
```

1 | #lang s-exp "graphicsexpr.rkt"

2 | (parameterize ([distance-top 80]

3 | [distance-left 200]

4 | [distance-bottom 160]

5 | [distance-right 200]))

6 | (RECT "HELLO ,WORLD"))

7 |

8 |

9 |

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.

HELLO ,WORLD

>

PARAMETER : BORDER-WIDTH,BORDER-COLOR

```
(PARAMETERIZE ([BORDER-WIDTH 3]
                [BORDER-COLOR "RED"]))
(RECT "HELLO ,WORLD"))
```

1 | #lang s-exp "graphicsexpr.rkt"

2 | (parameterize ([border-width 3]

3 | [border-color "red"])

4 | (RECT "HELLO ,WORLD"))

5 |

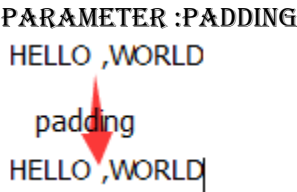
6 |

7 |

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.

HELLO ,WORLD

>



```
(PARAMETERIZE ([PADDING 40])
  (V "HELLO ,WORLD" "HELLO ,WORLD"))
```

```
1 | #lang s-exp "graphicsexpr.rkt"
2 | (parameterize ([padding 40])
3 |   (V "HELLO ,WORLD" "HELLO ,WORLD"))
4 |
5 |
6 |
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.
HELLO ,WORLD

HELLO ,WORLD
>

MORE PARAMETERS : TEXT-COLOR TEXT-STYLE TEXT-SIZE

6.PLAY WITH COLORS & MORE SHAPES

```
(DEFINE (REVERSE S)
  (PARAMETERIZE ([TEXT-COLOR "WHITE"]
                 [COLOR "BLACK"]))
  (FILLED-RECT S)))
```

(REVERSE "HELLO WORLD!!")

```
1 | #lang s-exp "graphicsexpr.rkt"
2 | (define (REVERSE s)
3 |   (parameterize ([text-color "white"]
4 |                 [color "black"]))
5 |     (FILLED-RECT s)))
6 |
7 | (REVERSE "Hello World!!")
8 |
9 |
10 |
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.

Hello World!!

>

```
(DEFINE (REVERSE S)
  (PARAMETERIZE ([TEXT-COLOR "WHITE"]
                 [COLOR "BLACK"]
                 (FILLED-RECT (BGCOLOR "BLUE" S)))))
```

(REVERSE "HELLO WORLD!!")

```
1 | #lang s-exp "graphicsexpr.rkt"
2 | (define (REVERSE s)
3 |   (parameterize ([text-color "white"]
4 |                 [color "black"]
5 |                 (FILLED-RECT (BGCOLOR "BLUE" s)))))
6 |
7 | (REVERSE "Hello World!!")
8 |
9 |
10 |
```

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.

Hello World!!

>


```
(DEFINE T1 "START A NEW GAME")
(DEFINE T2 "LOAD OLD SAVES")
(DEFINE T3 "ABOUT")
(DEFINE T4 "EXIT")

(BGCOLOR "LIGHTBLUE" (PARAMETERIZE ([TEXT-COLOR "WHITE"] [TEXT-SIZE 16])
  (FIX-RECT 400 280 (FIX-RECT 380 260
    (V (CENTER-H 360 "MENU")
      (FIX-FILLED-ROUNDED-RECT 360 40 (CENTER-H 360 T1))
      (FIX-ROUNDED-RECT 360 40 (CENTER-H 360 T2))
      (FIX-ROUNDED-RECT 360 40 (CENTER-H 360 T3))
      (FIX-ROUNDED-RECT 360 40 (CENTER-H 360 T4))
    )))))
```

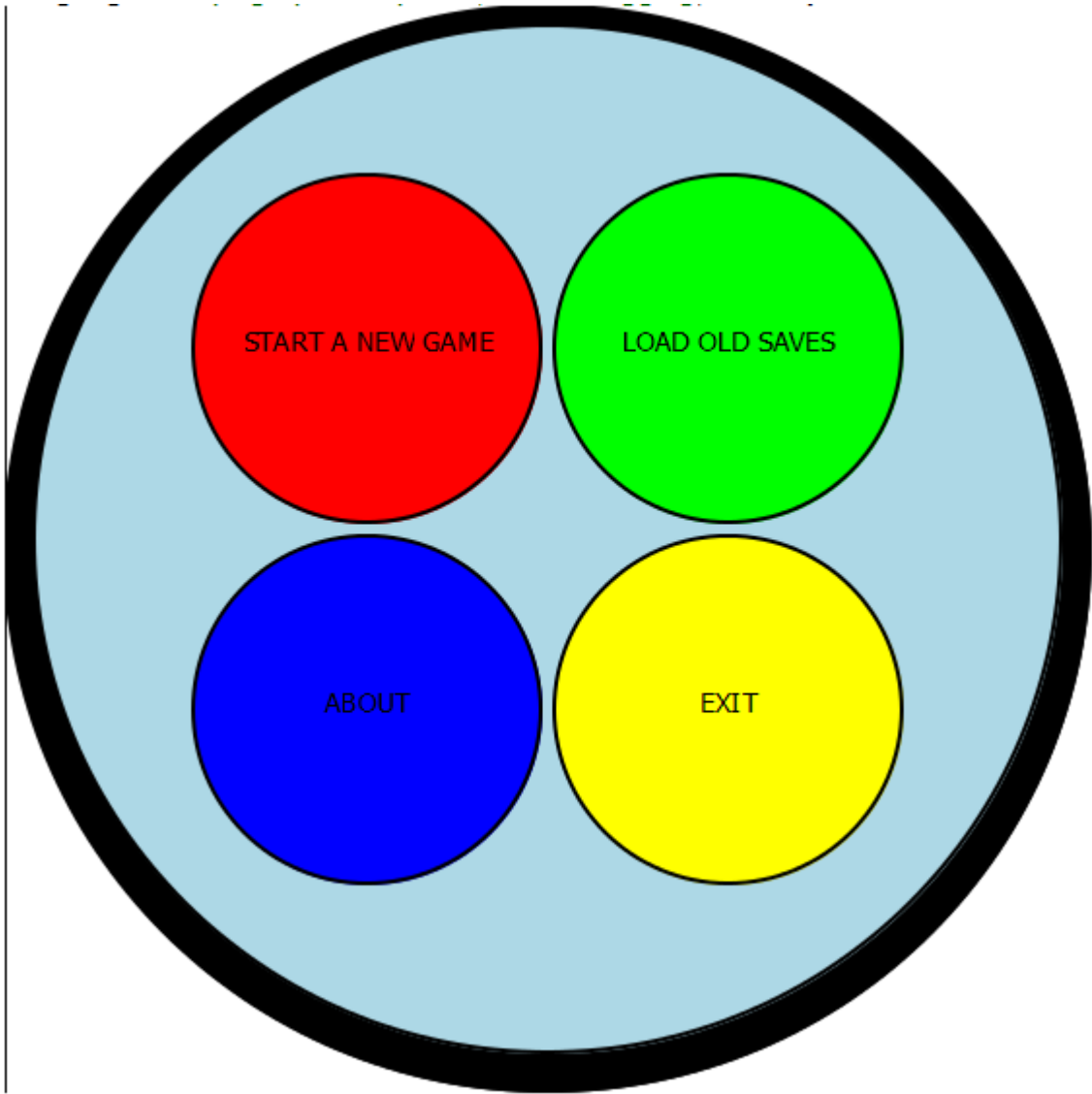


CIRCLES

```
(DEFINE SPACE 150)
(DEFINE T1 (CENTER SPACE "START A NEW GAME"))
(DEFINE T2 (CENTER SPACE "LOAD OLD SAVES"))
(DEFINE T3 (CENTER SPACE "ABOUT"))
(DEFINE T4 (CENTER SPACE "EXIT"))

(DEFINE COLORED-CIRCLE
  (LAMBDA (C E)
    (PARAMETERIZE ([COLOR C])
      (FILLED-CIRCLE E))))

(PARAMETERIZE ([TEXT-SIZE 16])
  (COLORED-CIRCLE "BLACK"
    (COLORED-CIRCLE "LIGHTBLUE"
      (CENTER 500 (V (H (COLORED-CIRCLE "RED" T1)
        (COLORED-CIRCLE "GREEN" T2))
        (H (COLORED-CIRCLE "BLUE" T3)
          (COLORED-CIRCLE "YELLOW" T4)
        ))))))))
```



7 .PARAGRAPH

(P 400 "LISP 依依鸭(XXXXXXXXXX) 10:32:46 我这一生都服姓马的人,第一个马克思改变了我的思想,第二个马化腾改变了我的交流方式,第三个马云改变了我的消费观念,第四个马蓉颠覆了我的人生观,第五个马赛克停止了我对人类文明的探索。ATOM(XXXXXXX) 10:33:19 可以 自由建客<XXXXXXXXXXXX> 10:42:01 出去天氣、工作、狀態等,你是不是要考慮下麵條是真是假? 自由建客<XXXXXXXXXXXX> 10:43:11 鹽可激發腎陽,在一定程度上感覺不到疲勞,但飢餓是沒法消除的。PROBLUE(XXXXXXX) 10:46:23 序对本身也是抽象的,它是基于 CONS、CAR 和 CDR 的。CONS 返回的也许用 DEFINE 可以引用之,但是其本身啥的确是不知道 SS-X<XXXXXXXXXXXX> 10:47:27 那是他的实现方式是抽象的 SS-X<XXXXXXXXXXXX> 10:47:40 序对本身的意义是明确的,并不抽象 SS-X<XXXXXXXXXXXX> 10:48:06 可以用各种不同的具体实现方式来实现这个明确的意义")

Welcome to [DrRacket](#), version 6.6 [3m].
Language: s-exp "graphicsexpr.rkt", with debugging; memory limit: 1000 MB.
Lisp依依鸭(xxxxxxxxxx) 10:32:46我这一生都服姓马的人,第一个马克思改变了我的思想,第二个马化腾改变了我的交流方式,第三个马云改变了我的消费观念,第四个马蓉颠覆了我的人生观,第五个马赛克停止了我对人类文明的探索。atom(xxxxxxx) 10:33:19可以 自由建客<xxxxxxxxxxxx> 10:42:01出去天氣、工作、狀態等,你是不是要考慮下麵條是真是假? 自由建客<xxxxxxxxxxxx> 10:43:11鹽可激發腎陽,在一定程度上感覺不到疲勞,但飢餓是沒法消除的。problue(xxxxxxxxxx) 10:46:23序对本身也是抽象的,它是基于cons、car和cdr的。cons返回的也许用define可以引用之,但是其本身啥的确是不知道SS-X<xxxxxxxxxxxx> 10:47:27那是他的实现方式是抽象的SS-X<xxxxxxxxxxxx> 10:47:40序对本身的意义是明确的,并不抽象SS-X<xxxxxxxxxxxx> 10:48:06可以用各种不同的具体实现方式来实现这个明确的意义

>

欢迎提出意见和 BUG!!