

# RACKET

---

## Scope

## Scope

- Racket is statically or lexically scoped.
- You can look at the program text and determine the value of a variable
- Lambda expressions create procedures.
- Additionally, they bind variables
- Those variables are known only inside of the lambda expression.
- Example:  

```
(lambda (x) (+ x 1))
```

x is not known outside of the lambda expression
- Example:  

```
(lambda (x) (lambda (y) (+ x y)))
```

y is not known outside of the inner lambda expression

## Contour Diagrams

```
(lambda (x)
  (lambda (y)
    ((lambda (x)
      (x y))
     x)
  )
)
```

## Practice Contour Diagrams

```
(define contour
  (lambda (x y)
    (let ([z (+ x y)])
      (lambda (x z)
        (let ([x (let ([x (+ x y z)]
                       [y 11])
                  (+ x y z))])
          (+ x y z))))))
```

## Practice Contour Diagrams

```

(define contour
  (lambda (x y)
    (let ([z (+ x y)])
      (lambda (x z)
        (let ([x (let ([x (+ x y z)]
                       [y 11])
                  (+ x y z))]
              (+ x y z))]
          (+ x y z))))))

```