Notation

Instead of traditional arithmetic notation, we'll use ***Racket*** notation

a.k.a. ***Lisp***, ***Scheme***, ***Beginning Student***...

Traditional	**Racket**
f(x) = cos(x) + 2	`(define (f x)` `  (+ (cos x) 2))`

Racket Expression Notation

- Put all operators at the front
- Start every operation with an open parenthesis
- Put a close parenthesis after the last argument
- Never add extra parentheses

Traditional	**Racket**
1 + 2	`(+ 1 2)`
4 + 2 × 3	`(+ 4 (* 2 3))`
cos(0) + 1	`(+ (cos 0) 1)`

Racket Definition Notation

- Use **`define`** instead of =
- Put **`define`** at the front, and group with parentheses
- Move open parenthesis from after function name to before

Traditional	Racket
f(x) = cos(x) + 2	`(define (f x) (+ (cos x) 2))`

- Move open parenthesis in function calls

Traditional	Racket
f(0)	`(f 0)`
f(2+3)	`(f (+ 2 3))`

Evaluation is the Same as Before

```
(define (f x) (+ (cos x) 2))

(f 0)
```

Evaluation is the Same as Before

```
(define (f x) (+ (cos x) 2))

(f 0)
→ (+ (cos 0) 2)
```

Evaluation is the Same as Before

```
(define (f x) (+ (cos x) 2))

(f 0)
→ (+ (cos 0) 2)
→ (+ 1 2)
```

Evaluation is the Same as Before

```
(define (f x) (+ (cos x) 2))

(f 0)
→ (+ (cos 0) 2)
→ (+ 1 2)
→ 3
```