

# Functions versus Constants

Calling a function:

```
(sqrt 25)
```

# Functions versus Constants

Calling a function:

`(sqrt 25)` → 5

`(+ (sqrt 25) 1)` → `(+ 5 1)` → 6

Using a constant:

`pi`

# Functions versus Constants

Calling a function:

```
(sqrt 25) → 5
```

```
(+ (sqrt 25) 1) → (+ 5 1) → 6
```

Using a constant:

```
pi → #i3.14
```

```
(+ 1 pi) → (+ 1 #i3.14) → #i4.14
```

# Defining Constants

Use `define` and *name* without parentheses around *name* to define a constant:

```
(define cake 4)
```

# Defining Constants

Use `define` and *name* without parentheses around *name* to define a constant:

```
(define cake 4)
```

No parenthesis after `define`, so `cake` is a constant

A parenthesis after `define`, so `next` is a function

```
(define (next x) (+ x 1))
```

# Defining Constants

Use `define` and *name* without parentheses around *name* to define a constant:

```
(define cake 4)
```

Use the *name* without parentheses:

```
(+ 1 cake)
```

```
→ (+ 1 4)
```

```
→ 5
```

# Defining Constants

Use `define` and *name* without parentheses around *name* to define a constant:

```
(define ~u (rotate 180 ))
```

Use the *name* without parentheses:

```
(beside ~u ~u)
```

→ → 

→ 