

Using Simple Macro

macro as find-and-replace

x ← a

y ← tmp

```
(macro (swap x y)
  (let ([tmp x])
    (set! x y)
    (set! y tmp)))
```

(define a 1)

(define tmp 2)

...

(println a)

(println tmp)

(let ([tmp a])

(set! a tmp)

(set! tmp tmp)))

>>> 1

>>> 1

Using Hygienic Macro

names in macros never
conflict with supplied names

$x' \leftarrow a$
 $y' \leftarrow tmp$

```
(macro (swap x' y')  
  (let ([tmp' x'])  
    (set! x' y')  
    (set! y' tmp')))
```

```
(define a 1)  
(define tmp 2)  ← (let ([tmp' a])  
...              (set! a y')  
                  (set! b tmp')))  
(println a)  
(println tmp)
```

→ 2

→ 1