



Buğra Gedik's Courses @ Bilkent University

cs315:scheme_examples

Scheme examples, with their Python counterparts

Factorial:

```

1 (define (fact n)
2   (if (= n 1)
3       1
4       (* n (fact (- n 1)))))
5 )
6 )
7 (fact 5)
8 >> 120

```

```

1 def fact(n):
2     if n==1:
3         return 1
4     else:
5         return n * fact(n-1)
6 fact(5)
7 >> 120

```

List length:

```

1 (define (length n)
2   (if (null? n)
3       0
4       (+ 1 (length (cdr n)))))
5 )
6 )
7 (length '(3 1 8))
8 >> 3

```

```

1 def length(n):
2     if n==[]:
3         return 0
4     else:
5         return 1 + length(n[1:])
6 length([3, 1, 8])
7 >> 3

```

Reverse list:

```

1 (define (reverse n)
2   (if (null? n)
3       '()
4       (let*
5         (
6           (head (car n))
7           (tail (cdr n))
8           (revTail (reverse tail)))
9         )
10    (append revTail (list head)))
11 )
12 )
13 )
14 (reverse '(3 1 5))
15 >> (5 1 3)

```

```

1 def reverse(n):
2     if n==[]:
3         return []
4     else:
5         head = n[0]
6         tail = n[1:]
7         revTail = reverse(tail)
8         return revTail + [head]
9 reverse([3, 1, 5])
10 >> [5, 1, 3]

```

Comparison (eqv? does shallow comparison, equal? does deep comparison):

```

1 (let*
2   (
3     (x '(1 3))
4     (y '(1 3))
5     (z x))
6 )

```

```

1 x = [1, 3]
2 y = [1, 3]
3 z = x
4 print x is y
5 print x is z
6 print x == y

```

<pre> 7 (display (eqv? x y)) (newline) 8 (display (eqv? x z)) (newline) 9 (display (equal? x y)) (newline) 10 (display (equal? x z)) (newline) 11) 12 >> #f 13 #t 14 #t 15 #t </pre>	<pre> 7 print x == z 8 >> False 9 True 10 True 11 True </pre>
--	--

Find indices:

```

1  (define (find needle haystack)
2    (letrec
3      (
4        (findi
5          (lambda (needle haystack iter)
6            (if
7              (null? haystack)
8              '()
9              (let*
10                 (
11                   (current (car haystack))
12                   (remains (cdr haystack))
13                   (nextIt (+ iter 1))
14                   (subres (findi needle remains nextIt))
15                 )
16                 (if
17                   (equal? needle current)
18                   (cons iter subres)
19                   subres
20                 )
21               )
22             )
23           )
24         )
25       )
26     (findi needle haystack 0)
27   )
28 )
29 (find '(3) '(1 2 (3) 4 5 (3) 7 (8 (3) 9) ))
30 >> (2, 5)

```

```

1  def find(needle, haystack):
2    def findi(needle, haystack, iter):
3      if haystack==[]:
4        return []
5      else:
6        current = haystack[0]
7        remains = haystack[1:]
8        nextIt = iter + 1
9        subres = findi(needle, remains, nextIt)
10       if needle == current:
11         return [iter] + subres
12       else:
13         return subres
14     return findi(needle, haystack, 0)
15 find([3], [1, 2, [3], 4, 5, [3], 7, [8, [3], 9]])
16 >> [2, 5]

```

cs315/scheme_examples.txt · Last modified: 2013/12/19 20:27 by bgedik