

# Conditional Expressions

There are two *conditional* expressions:

```
(if <test> <test-true exp> <test-false exp>)
```

Note that you are required to have all three parts of the if-expression; you can't say something like "if x < 3 return 'small else do nothing"

```
(cond  
  [test1 exp1]  
  [test2 exp2]  
  ... )
```

You can use the symbol *else* for the final test.

Note that the square bracket is just an alternative parenthesis.

This look through the test conditions until it finds one that is true, then returns the corresponding expression.

E.g.

```
(if (< 1 2) 3 4) => 3
```

```
(cond
```

```
  [(< 2 1) 3]
```

```
  [(< 5 6) 4]
```

```
  [else 5])
```

```
=> 4
```

We can put all of this together to get our first interesting procedure:

```
(define f (lambda (x)
  (cond
    [(= x 1) 1]
    [else (* x (f (- x 1)))])))
```