

# The Design Recipe

**Directions:** Write a function `rect-perimeter` that takes in the length and width of a rectangle and returns the perimeter of that rectangle.

## Contract and Purpose Statement

Every contract has three parts...

`; rect-perimeter:` Number Number  $\rightarrow$  Number  
function name domain range

`; Takes in 2 numbers, length and width, and returns the double of the sum of both numbers`  
what does the function do?

## Examples

Write some examples, then circle and label what changes...

(EXAMPLE (`rect-perimeter` 10 20) (`* 2 (+ 10 20)`))  
function name input(s) what the function produces

(EXAMPLE (`rect-perimeter` 200 350) (`* 2 (+ 200 350)`))  
function name input(s) what the function produces

## Definition

Write the definition, giving variable names to all your input values...

(define (`rect-perimeter` length width)  
function name variable(s)

(`* 2 (+ length width)`)  
what the function does with those variable(s)

**Directions:** Write a function `rectprism-vol` that takes in the length, width, and height of a rectangular prism and returns the Volume of a rectangular prism.

## Contract and Purpose Statement

Every contract has three parts...

`; rectprism-vol :` Number Number Number  $\rightarrow$  Number  
function name domain range

`; Takes in 3 numbers, length, width, and height, and multiplies them to return that value`  
what does the function do?

## Examples

Write some examples, then circle and label what changes...

(EXAMPLE (`rectprism-vol` 10 20 30) (`* 10 (* 20 30)`))  
function name input(s) what the function produces

(EXAMPLE (`rectprism-vol` 100 250 350) (`* 100 (* 250 350)`))  
function name input(s) what the function produces

## Definition

Write the definition, giving variable names to all your input values...

(define (`rectprism-vol` length width height)  
function name variable(s)

(`* length (* width height)`)  
what the function does with those variable(s)