

The Design Recipe

Directions: Getting a gym membership costs \$150, and then there's a \$45/month fee after that. Write a function `globo-gym` that takes in a number of months and produces the cost of a membership for that many months.

Contract and Purpose Statement

Every contract has three parts...

The diagram illustrates the components of a function signature. It consists of three horizontal lines. The top line is labeled 'function name' on the left, 'domain' in the middle, and 'range' on the right. The middle line is labeled 'domain' and the right line is labeled 'range'. The bottom line is labeled 'what does the function do?'. The lines are connected by a vertical line on the left and a vertical line on the right. The top line ends with a colon ':' and the bottom line ends with a semicolon ';'. The middle line ends with a right-pointing arrow '->'.

Examples

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

(EXAMPLE ())
 function name *input(s)* *what the function produces*

(EXAMPLE ())
 function name *input(s)* *what the function produces*

Definition

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

```
(define ( function name variable(s) )
  what the function does with those variable(s)
)
```

Directions: The cost of a ride is a starting price of \$2.50, plus \$1.50/mile. Write a function `rideshare`, that takes in a number of miles and produces the cost of that ride.

Contract and Purpose Statement

Every contract has three parts...

The diagram illustrates the components of a function signature. It consists of three horizontal lines. The top line is labeled 'function name' on the left, followed by a colon ':', then a long horizontal line labeled 'domain', followed by an arrow '→', and finally a horizontal line labeled 'range'. Below the 'domain' line, there is a semicolon ';' and a horizontal line labeled 'what does the function do?'.

Examples □

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

(EXAMPLE ())
 function name *input(s)* *what the function produces*

(EXAMPLE ())
 function name *input(s)* *what the function produces*

Definition □

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

```
(define ( function name variable(s) )
  what the function does with those variable(s)
)
```