

Name: \_\_\_\_\_



# **BOOTSTRAP**

**www.bootstrapworld.org**

**Student Workbook**

Class: \_\_\_\_\_

# Bootstrap Units

<b>01</b>	Videogames and Coordinate Planes	<b>06</b>	Comparing Functions
<b>02</b>	Contracts, Strings, and Images	<b>07</b>	Conditional Branching
<b>03</b>	Intro to Definitions	<b>08</b>	Collision Detection
<b>04</b>	Design Recipe	<b>09</b>	Prepping for Launch
<b>05</b>	Game Animation	<b>10</b>	Additional Material

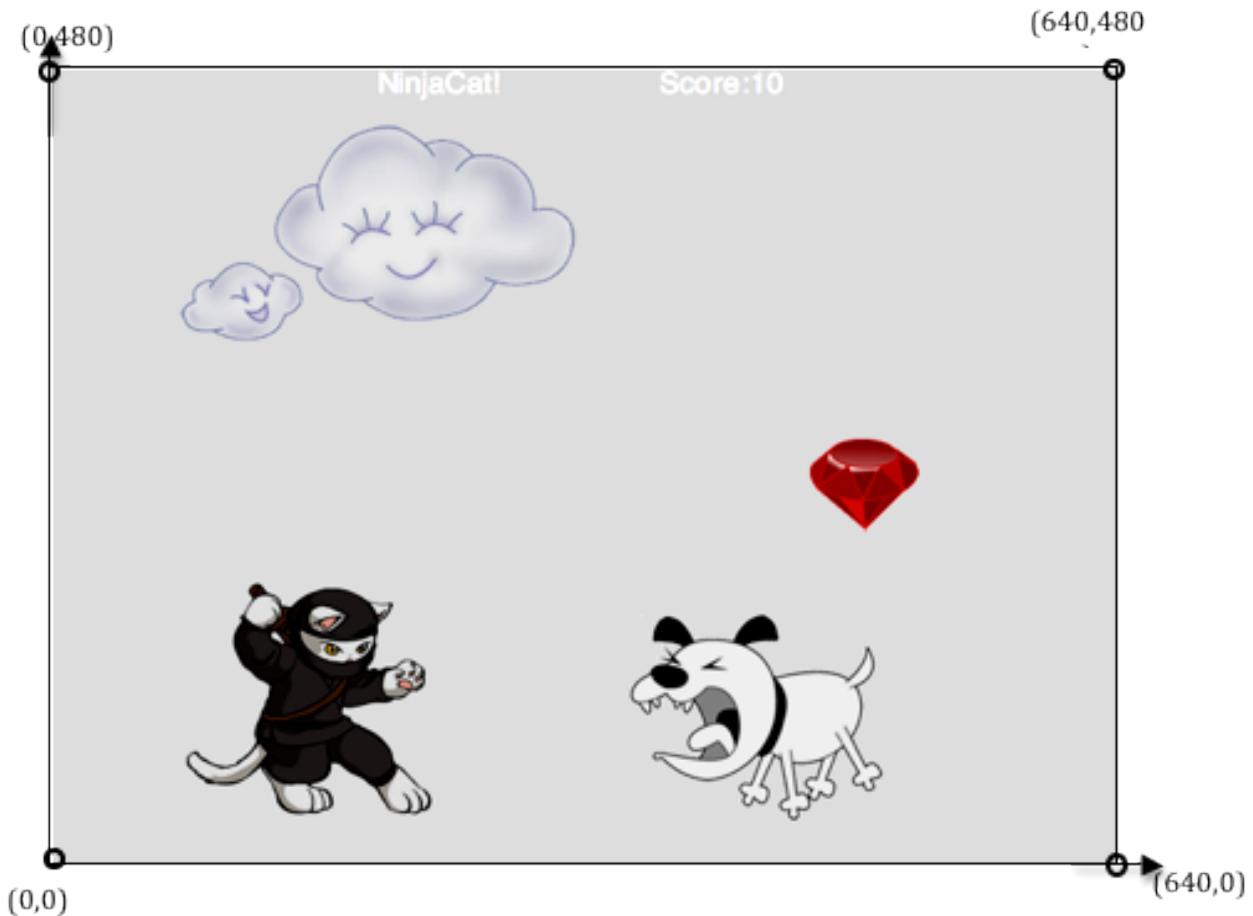
# 01 Videogames and Coordinate Planes



# Lesson 1

## Reverse-Engineering: How does NinjaCat work?

## Finding Coordinates



The coordinates for the PLAYER (NinjaCat) are: ( \_\_\_\_\_ , \_\_\_\_\_ )  
x-coordinate      y-coordinate

The coordinates for the DANGER (Dog) are: ( \_\_\_\_\_ , \_\_\_\_\_ )

The coordinates for the TARGET (Ruby) are: ( \_\_\_\_\_ , \_\_\_\_\_ )

# Our Videogame

Created by (write your names): \_\_\_\_\_

## Background

Our game takes place in: \_\_\_\_\_  
(space? the desert? a mall?)

## The Player

*The player is a \_\_\_\_\_.*

The player moves only up and down.

## The Target

*Your player GAINS points when they hit the target.*

*The Target is a \_\_\_\_\_.*

The Target moves only to the left and right.

## The Danger

*Your player LOSES points when they hit the danger.*

*The Danger is a \_\_\_\_\_.*

The Danger moves only to the left and right.

## Circle of Evaluation Practice

**Time: 5 minutes**

Don't forget to use the computer's symbols for things like multiply and divide!

<b>Math</b>	<b>Circle of Evaluation</b>	<b>Pyret Code</b>
$5 \times 10$		
$8 + (5 \times 10)$		
$(8 + 2) - (5 \times 10)$		
$\frac{5 \times 10}{8 - 2}$		

# 02 Contracts, Strings, and Images



# Circles Competition minutes

Time: 5

<b><i>Math</i></b>	<b><i>Circle of Evaluation</i></b>	<b><i>Pyret Code</i></b>
<i>Round 1</i>	$(3 * 7) - (1 + 2)$	
<i>Round 2</i>	$3 - (1 + 2)$	
<i>Round 3</i>	$3 - (1 + (5 * 6))$	
<i>Round 4</i>	$(1 + (5 * 6)) - 3$	

# 03

## Intro to Definitions



## Fast Functions

\_\_\_\_\_ :: \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

examples:

\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_  
\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_

end

fun \_\_\_\_\_(\_\_\_\_\_) : \_\_\_\_\_ end

\_\_\_\_\_ :: \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

examples:

\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_  
\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_

end

fun \_\_\_\_\_(\_\_\_\_\_) : \_\_\_\_\_ end

\_\_\_\_\_ :: \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

examples:

\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_  
\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_

end

fun \_\_\_\_\_(\_\_\_\_\_) : \_\_\_\_\_ end

## Fast Functions

\_\_\_\_\_ :: \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

examples:

\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_  
\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_

end

fun \_\_\_\_\_(\_\_\_\_\_) : \_\_\_\_\_ end

\_\_\_\_\_ :: \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

examples:

\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_  
\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_

end

fun \_\_\_\_\_(\_\_\_\_\_) : \_\_\_\_\_ end

\_\_\_\_\_ :: \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

examples:

\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_  
\_\_\_\_\_ (\_\_\_\_\_ ) is \_\_\_\_\_

end

fun \_\_\_\_\_(\_\_\_\_\_) : \_\_\_\_\_ end

# 04 Design Recipe

1 Contract

2 Example

3 Definition

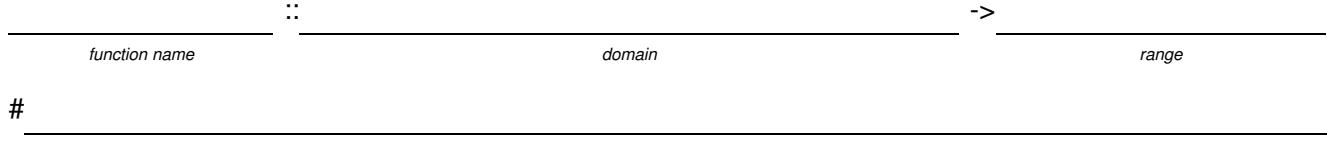


# Word Problem: rocket-height

**Directions:** A rocket blasts off, traveling at 7 meters per second. Write a function called 'rocket-height' that takes in the number of seconds that have passed since the rocket took off, and which produces the height of the rocket at that time.

## Contract and Purpose Statement

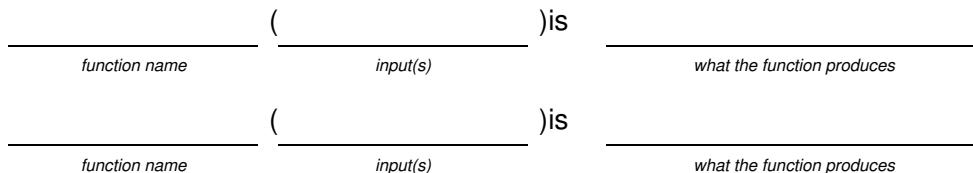
Every contract has three parts...



## Examples

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

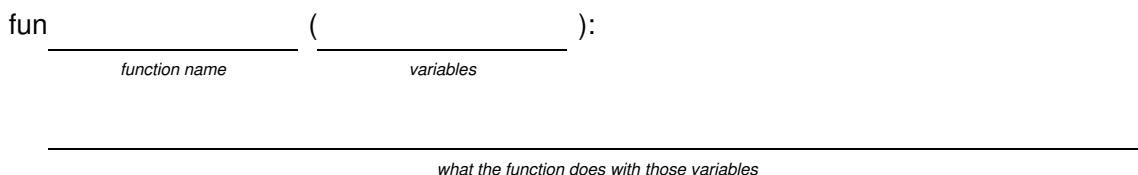
examples:



end

## Definition

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



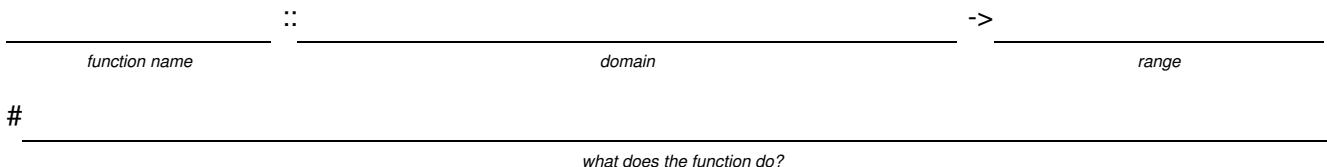
end

# Word Problem: red-square

**Directions:** Use the Design Recipe to write a function 'red-square', which takes in a number (the side of the square) and outputs a solid red rectangle whose length and width are the same size.

## Contract and Purpose Statement

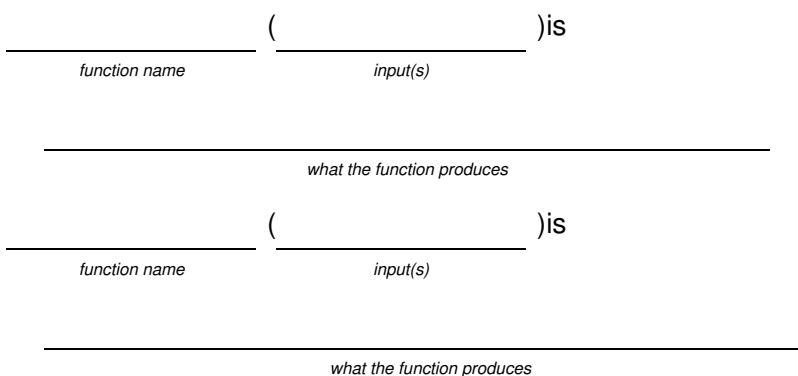
Every contract has three parts...



## Examples

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

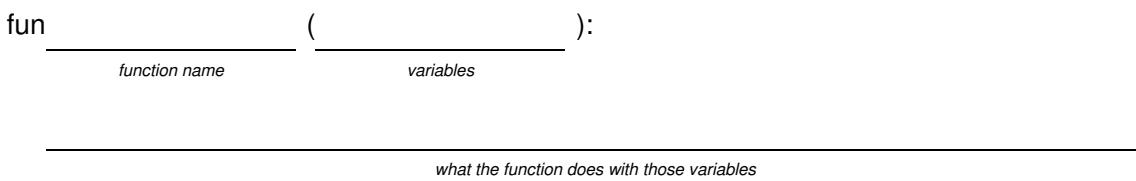
examples:



end

## Definition

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



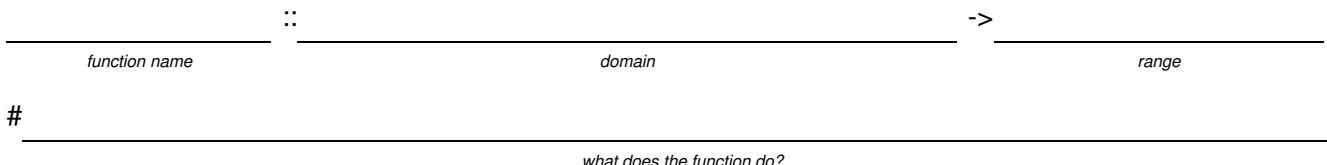
end

## Word Problem: lawn-area

**Directions:** Use the Design Recipe to write a function 'lawn-area', which takes in the width and length of a lawn, and returns the area of the lawn. (Don't forget: area = length \* width!)

### Contract and Purpose Statement

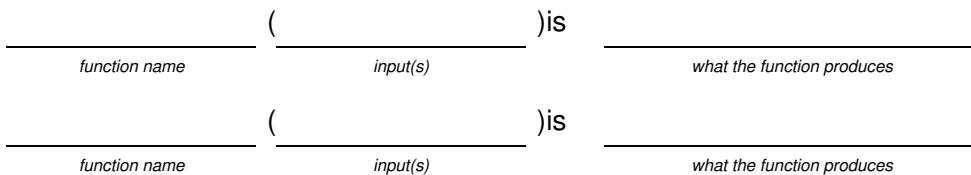
Every contract has three parts...



### Examples

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

examples:



end

### Definition

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

fun **function name** ( **variables** ):

**what the function does with those variables**

end

**target**



**danger**



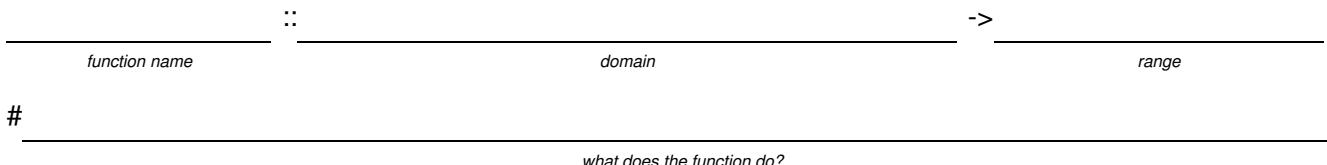
## 05 Game Animation

# Word Problem: update-danger

**Directions:** Use the Design Recipe to write a function 'update-danger', which takes in the danger's x-coordinate and y-coordinate and produces the next x-coordinate, which is 50 pixels to the left.

## Contract and Purpose Statement

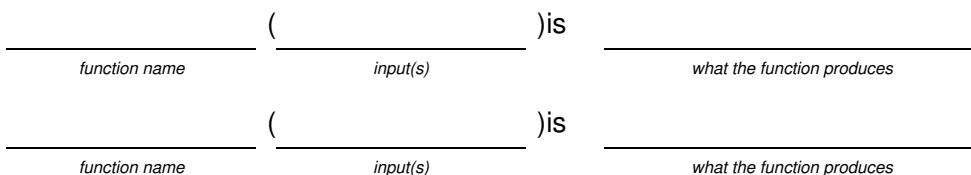
Every contract has three parts...



## Examples

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

examples:



end

## Definition

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

fun \_\_\_\_\_ (        x, y        ):  
    **function name**                          **variables**

\_\_\_\_\_  
**what the function does with those variables**

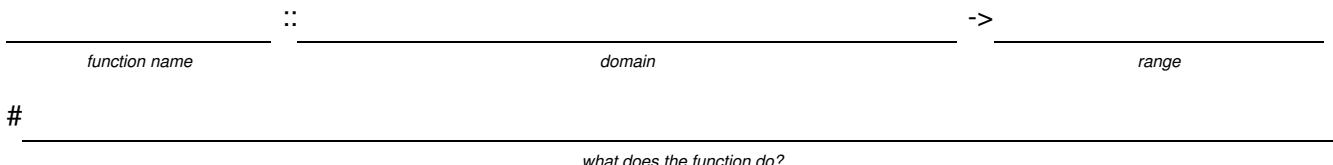
end

## Word Problem: update-target

**Directions:** Write a function 'update-target', which takes in the target's x-coordinate and y-coordinate and produces the next x-coordinate, which is 50 pixels to the right.

### Contract and Purpose Statement

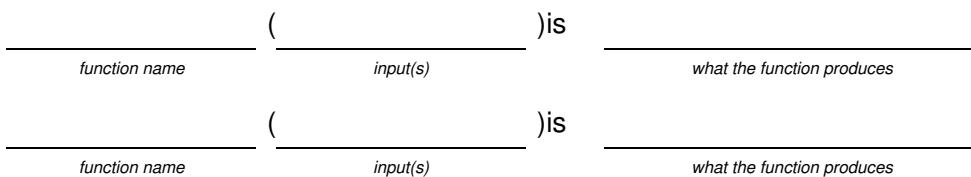
Every contract has three parts...



### Examples

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

examples:



end

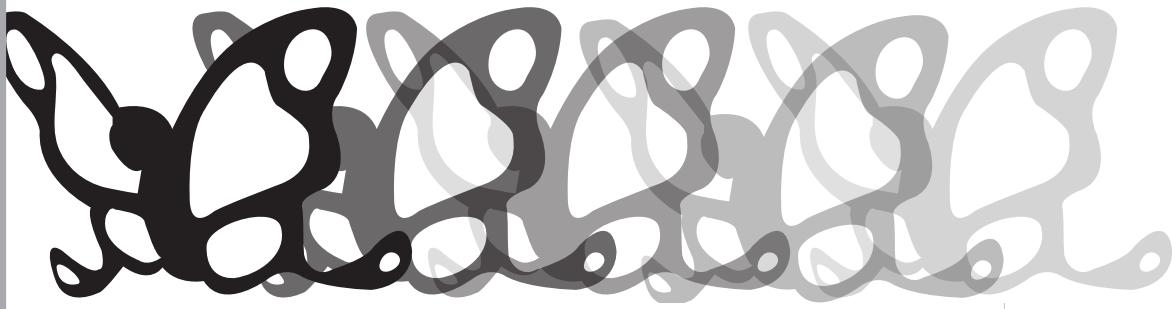
### Definition

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

fun \_\_\_\_\_ ( \_\_\_\_\_ ):  
  *function name*                   *variables*

\_\_\_\_\_  
  *what the function does with those variables*

end



***is-safe-left***

## 06 Comparing Functions

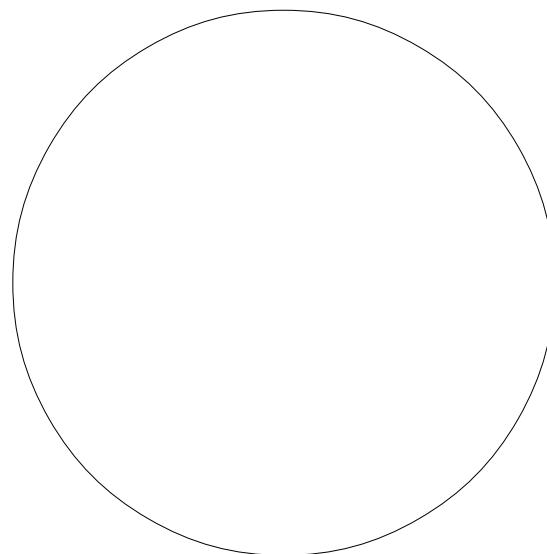
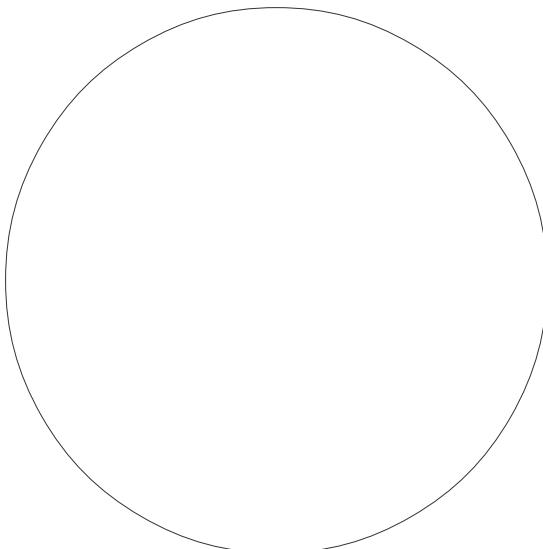
## DESIGN RECIPE

Sam is in a 640 x 480 yard. How far he can go to the left and right before he's out of sight?

1. A piece of Sam is still visible on the left as long as...  $x > -50$  \_\_\_\_\_

2. A piece of Sam is still visible on the right as long as... \_\_\_\_\_

3. Draw the Circle of Evaluation for these two expressions in the circles below:

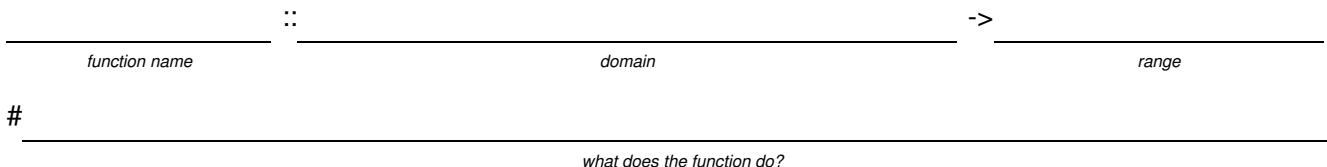


# Word Problem: is-safe-left

**Directions:** Use the Design Recipe to write a function 'is-safe-left', which takes in an x-coordinate and checks to see if the x-coordinate is greater than -50

## Contract and Purpose Statement

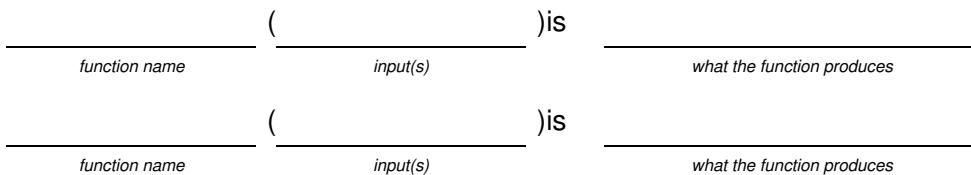
Every contract has three parts...



## Examples

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

examples:



end

## Definition

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

fun \_\_\_\_\_ ( \_\_\_\_\_ ):  
  **function name**                   **variables**

\_\_\_\_\_  
*what the function does with those variables*

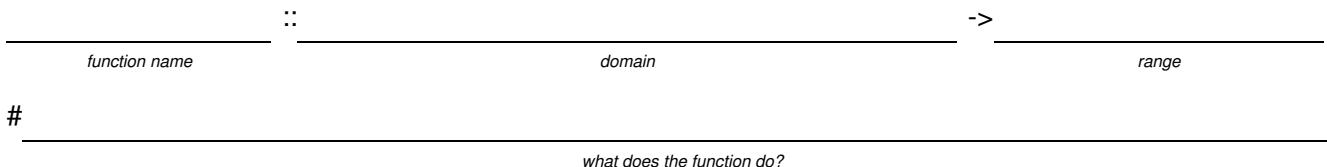
end

# Word Problem: is-safe-right

**Directions:** Use the Design Recipe to write a function 'is-safe-right', which takes in an x-coordinate and checks to see if the x-coordinate is less than 690.

## Contract and Purpose Statement

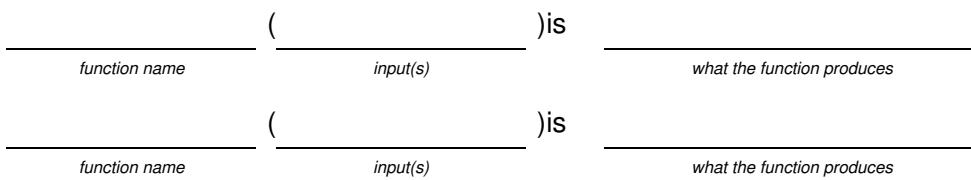
Every contract has three parts...



## Examples

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

examples:



end

## Definition

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

fun \_\_\_\_\_ ( \_\_\_\_\_ ):  
function name variables

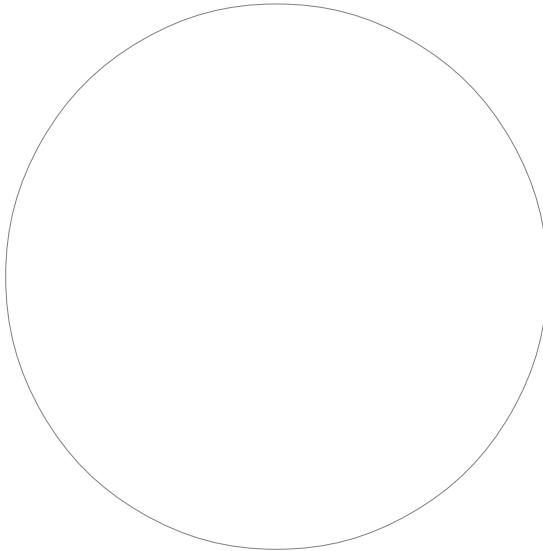
\_\_\_\_\_ what the function does with those variables

end

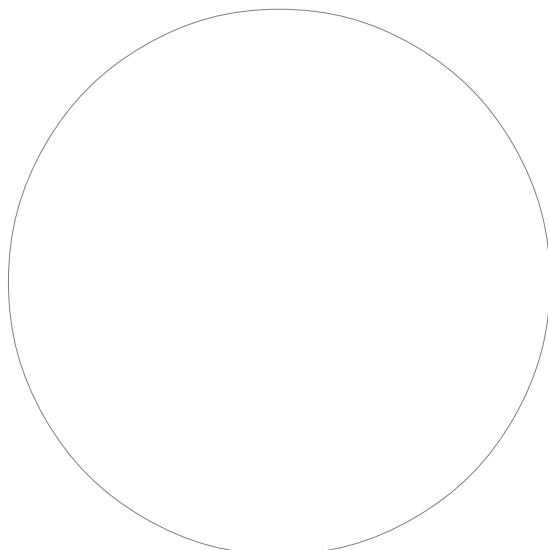
and / or

**Write the Circles of Evaluation for these statements, and then convert them to Pyret**

1. Two is less than five, and zero is equal to six.



2. Two is less than four or four is equal to six.

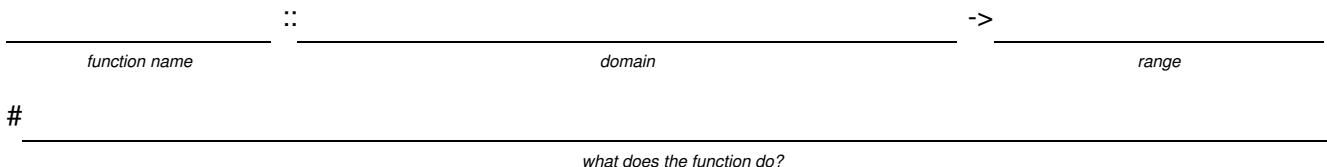


# Word Problem: is-onscreen

**Directions:** Use the Design Recipe to write a function 'is-onscreen', which takes in an x- and y-coordinate and checks to see if Sam is safe on the left AND safe on the right.

## Contract and Purpose Statement

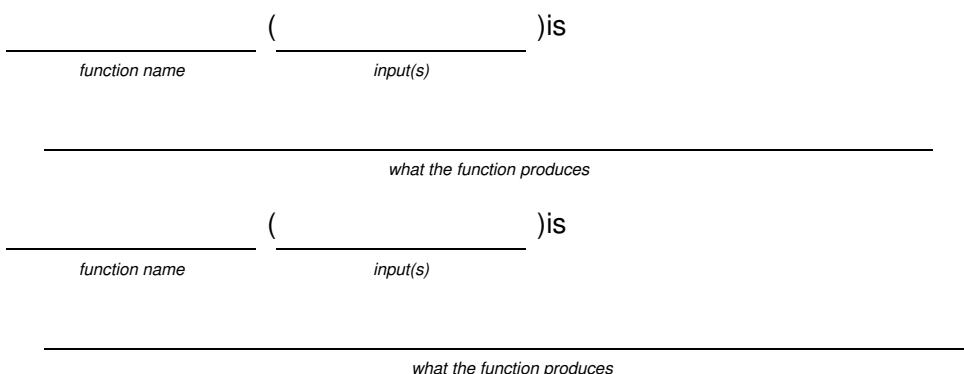
Every contract has three parts...



## Examples

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

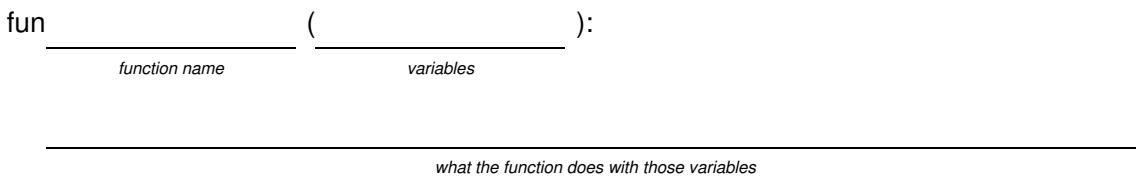
examples:



end

## Definition

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



end

# 07 Conditional Branching

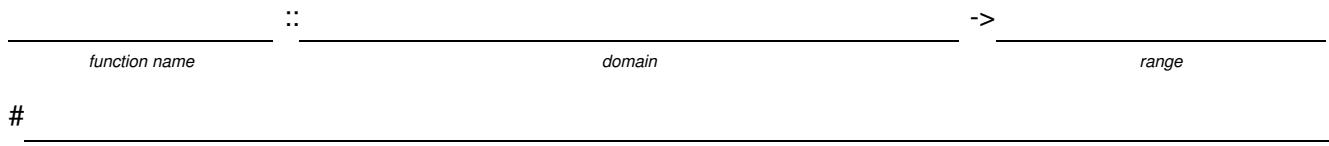


# Word Problem: cost

**Directions:** Luigi's Pizza has hired you as a programmer. They offer Pepperoni (\$10.50), Cheese (\$9.00), Chicken (\$11.25) and Broccoli (\$10.25). Write a function called "cost" which takes in the name of a topping and outputs the cost of a pizza with that topping.

## Contract and Purpose Statement

Every contract has three parts...



## Examples

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

examples:

cost	(	"pepperoni"	)is	
function name		input(s)		what the function produces
	(		)is	
function name		input(s)		what the function produces
	(		)is	
function name		input(s)		what the function produces
	(		)is	
function name		input(s)		what the function produces

end

## Definition

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

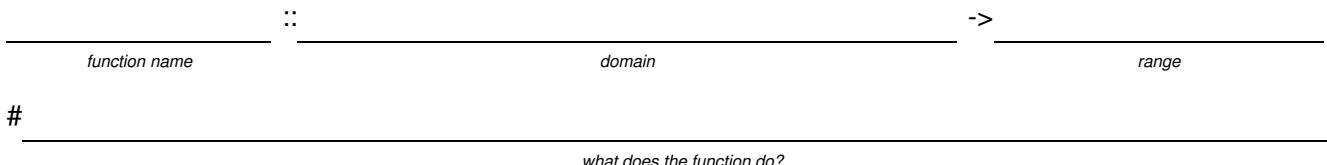
```
fun _____ ( _____ ):  
    function name           variables  
  
    if _____ : _____  
  
    else _____ : _____  
    if _____ : _____  
    else _____ : _____  
    if _____ : _____  
    else: _____  
  
end  
end
```

# Word Problem: update-player

**Directions:** Write a function called "update-player", which takes in the player's x-coordinate and y-coordinate, and the name of the key pressed, and returns the new y-coordinate.

## Contract and Purpose Statement

Every contract has three parts...



## Examples

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

examples:

update-player	( 100, 320, "up" )	is	_____	what the function produces
function name	input(s)			
update-player	( 200, 100, "up" )	is	_____	what the function produces
function name	input(s)			
(		)	is	_____
function name	input(s)			what the function produces
(		)	is	_____
function name	input(s)			what the function produces

end

## Definition

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

```
fun _____ ( _____ ):  
    function name           variables  
    if _____ :  
    else _____ :  
    else:  
    end  
end
```

# 08 Collision Detection

# collision



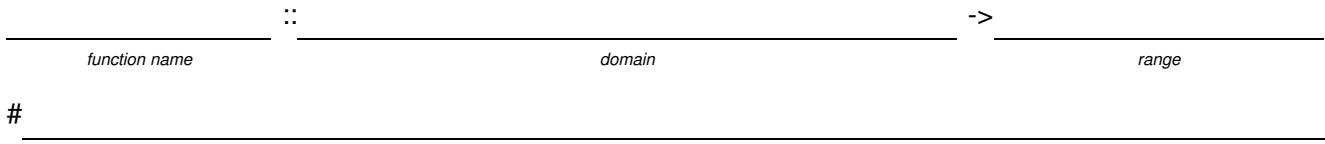
distance

# Word Problem: line-length

**Directions:** Write a function called 'line-length', which takes in two numbers and returns the \*positive difference\* between them. It should always subtract the smaller number from the bigger one, and if they are equal it should return zero.

## Contract and Purpose Statement

Every contract has three parts...



## Examples

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

examples:

line-length	(	10, 5	) is	10 - 5	
function name		input(s)		what the function produces	
line-length	(	2, 8	) is	8 - 2	

end

## Definition

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

fun \_\_\_\_\_ ( \_\_\_\_\_ ):  
  function name                                  variables  
  if \_\_\_\_\_ :  
    else: \_\_\_\_\_

  end

end

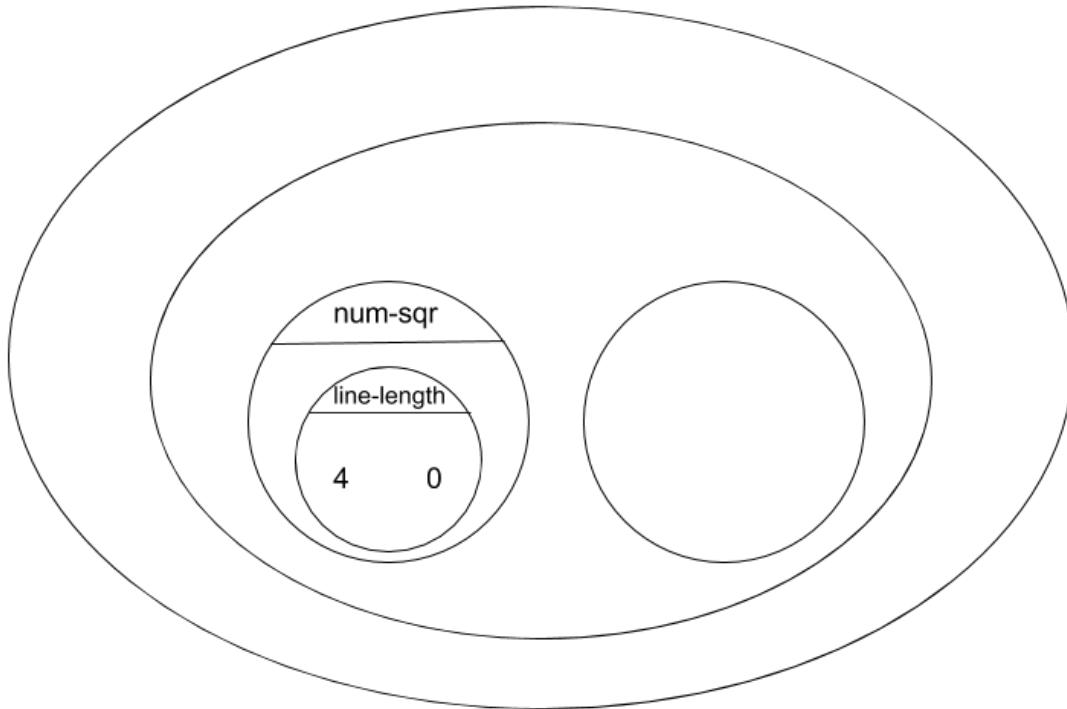
## The Distance Formula (an example)

The distance between the points (0, 0) and (4, 3) is given by:

$$\sqrt{(line-length\ 4\ 0)^2 + (line-length\ 3\ 0)^2}$$

---

Turn the formula above into a Circle of Evaluation. (We've already gotten you started!)



---

Convert the Circle of Evaluation into Pyret code:

## Word Problem: distance

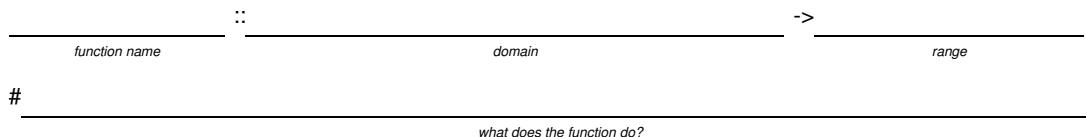
**Directions:** Write a function "distance", which takes FOUR inputs:

- px: The x-coordinate of the player
- py: The y-coordinate of the player
- cx: the x-coordinate of another game character
- cy: the y-coordinate of another game character

It should return the distance between the two, using the Distance formula. (HINT: look at what you did on the previous page!)

### Contract and Purpose Statement

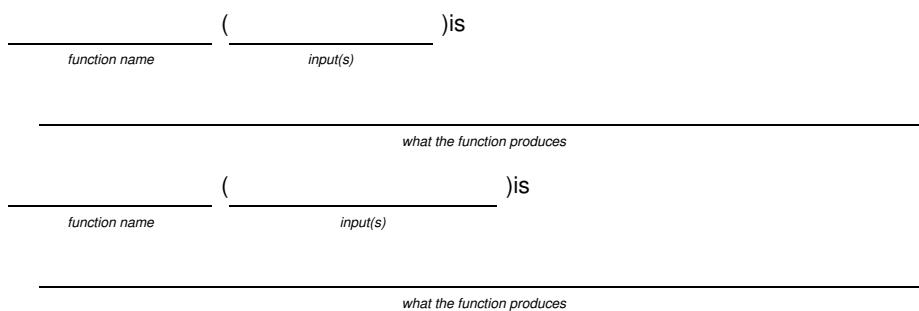
Every contract has three parts...



### Examples

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

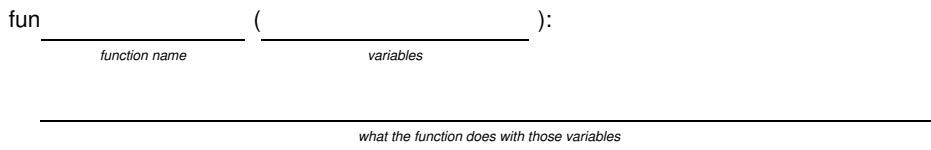
examples:



end

### Definition

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



end

# Word Problem: is-collision

**Directions:** Write a function "is-collision", which takes FOUR inputs:

- *px: The x-coordinate of the player*
- *py: The y-coordinate of the player*
- *cx: the x-coordinate of another game character*
- *cy: the y-coordinate of another game character*

Are the coordinates of the player within 50 pixels of the coordinates of the other character?

## Contract and Purpose Statement

Every contract has three parts...

\_\_\_\_ :: \_\_\_\_\_ -> \_\_\_\_\_  
function name domain range  
  
# \_\_\_\_\_  
what does the function do?

## Examples

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

examples:

\_\_\_\_ ( \_\_\_\_\_ )is  
function name input(s)  
  
\_\_\_\_\_  
what the function produces  
  
\_\_\_\_ ( \_\_\_\_\_ )is  
function name input(s)  
  
\_\_\_\_\_  
what the function produces

end

## Definition

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

fun \_\_\_\_\_ ( \_\_\_\_\_ ):  
function name variables  
  
\_\_\_\_\_  
what the function does with those variables

end



## 09 Presentation Preparation



# Lesson 9

Catchy Intro:

---

---

---

Name, Age, Grade:

---

Game Title:

---

---

---

---

Back Story:

---

---

---

---

Characters:

---

---

---

---

Explain a piece of your code:



## Presentation Feedback

*For each question, circle the answer that fits best.*

Was the introduction catchy?      No way!      A little.      Definitely!

Did they talk about their characters?    No way!      A little.      Definitely!

Did they explain the code well?    No way!      A little.      Definitely!

Did they speak slowly enough?    No way!      A little.      Definitely!

Did they speak loudly enough?    No way!      A little.      Definitely!

Were they standing confidently?    No way!      A little.      Definitely!

Did they make eye contact?      No way!      A little.      Definitely!

## Presentation Feedback

*For each question, circle the answer that fits best.*

Was the introduction catchy?      No way!      A little.      Definitely!

Did they talk about their characters?    No way!      A little.      Definitely!

Did they explain the code well?    No way!      A little.      Definitely!

Did they speak slowly enough?    No way!      A little.      Definitely!

Did they speak loudly enough?    No way!      A little.      Definitely!

Were they standing confidently?    No way!      A little.      Definitely!

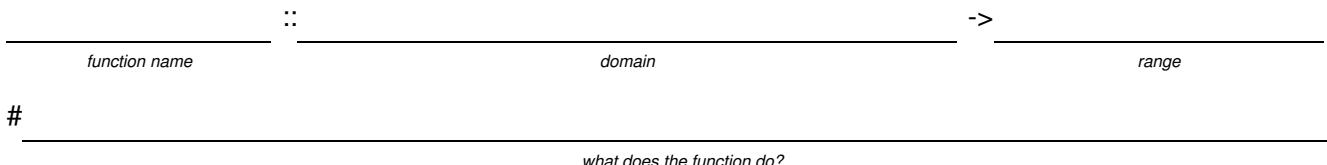
Did they make eye contact?      No way!      A little.      Definitely!

# Word Problem: red-shape

**Directions:** Write a function called "red-shape", which takes in the name of a shape and draws that shape (solid and red). Add an otherwise clause that produces a sensible output.

## Contract and Purpose Statement

Every contract has three parts...



## Examples

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

examples:

(                          )is  
function name              input(s)

what the function produces

(                          )is  
function name              input(s)

what the function produces

(                          )is  
function name              input(s)

what the function produces

(                          )is  
function name              input(s)

what the function produces

(                          )is              what the function produces  
function name              input(s)

end

## Definition

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

fun \_\_\_\_\_ ( \_\_\_\_\_ ):

*function name*                   *variables*

if \_\_\_\_\_ :

else \_\_\_\_\_ :

if

else \_\_\_\_\_ :

if

else \_\_\_\_\_ :

if

else:

end

end

# Translating into Algebra

## Value Definitions

Pyret Code	Algebra
<code>x = 10</code>	$x = 10$
<code>y = x * 2</code>	$y = x \cdot 2$
<code>z = x / y</code>	
<code>w = num-sqrt(num-sqr(x) + 1)</code>	
<code>days = (age * 12) * 30</code>	
<code>y = (v * x) + x0</code>	
<code>y = ((0.5 * a) * num-sqr(x)) + y0</code>	

## Function Definitions

Pyret Code	Algebra
<code>fun area(length, width):     length * width end</code>	$\text{area}(\text{length}, \text{width}) = \text{length} \cdot \text{width}$
<code>fun circle-area(radius):     pi * radius end</code>	
<code>fun distance(x1, y1, x2, y2):     num-sqrt(         num-sqr(x1 - x2)         + num-sqr(y1 - y2)     ) end</code>	

# Design Recipe

A rocket is flying from Earth to Mars at 80 miles per second. Write a function that describes the **distance**  $D$  that the rocket has traveled, as a function of **time**  $t$ .

## I. Contract+Purpose Statement

Every contract has three parts:

D \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name Domain Range  
# \_\_\_\_\_  
*What does the function do?*

## II. Give Examples

Write an example of your function for some sample inputs

D( 1 )      is \_\_\_\_\_

Use the function here      What should the function produce? \_\_\_\_\_

D( 2 )      =      is \_\_\_\_\_

Use the function here      What should the function produce? \_\_\_\_\_

D(    )      is \_\_\_\_\_  
Use the function here      What should the function produce? \_\_\_\_\_

                is \_\_\_\_\_  
Use the function here      What should the function produce? \_\_\_\_\_

## III. Definition

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

fun D(        ) :  
end

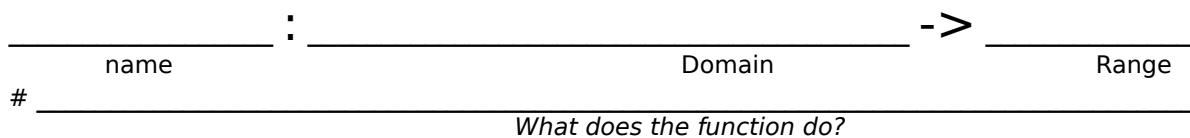


# Design Recipe

A rocket is traveling from Earth to Mars at 80 miles per second. Write a function that describes the ***time*** the rocket has been traveling, as a function of ***distance***.

## I. Contract+Purpose Statement

Every contract has three parts:



## II. Give Examples

Write an example of your function for some sample inputs

is Use the function here	What should the function produce?
is Use the function here	What should the function produce?
is Use the function here	What should the function produce?
is Use the function here	What should the function produce?

## III. Definition

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

fun                    (        ) :  
end

# Design Recipe

A rocket leaves Earth, headed for Mars at 80 miles per second. **At the exact same time**, an asteroid leaves Mars traveling towards Earth, moving at 70 miles per second. If the distance from the Earth to Mars is 50,000,000 miles, how long will it take for them to meet?

## I. Contract+Purpose Statement

Every contract has three parts:

\_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name Domain Range  
# \_\_\_\_\_  
*What does the function do?*

## II. Give Examples

Write an example of your function for some sample inputs

is  
Use the function here      What should the function produce?  
is  
Use the function here      What should the function produce?  
is  
Use the function here      What should the function produce?  
is  
Use the function here      What should the function produce?

## III. Definition

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

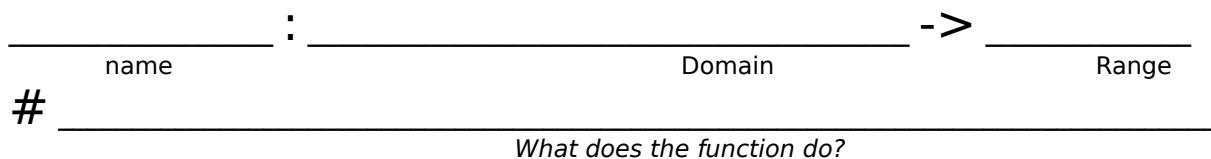
fun            (        ):  
end

# Design Recipe

A rocket leaves Earth, headed for Mars at 80 miles per second. **At the exact same time**, an asteroid leaves Mars traveling towards Earth, moving at 70 miles per second. If the distance from the Earth to Mars is 50,000,000 miles, how long will it take for them to meet?

## I. Contract+Purpose Statement

Every contract has three parts:



## II. Give Examples

Write an example of your function for some sample inputs

is	_____
Use the function here	What should the function produce?
is	_____
Use the function here	What should the function produce?
is	_____
Use the function here	What should the function produce?
is	_____
Use the function here	What should the function produce?

## III. Definition

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

fun ( ):  
end

## Contracts

## Contracts