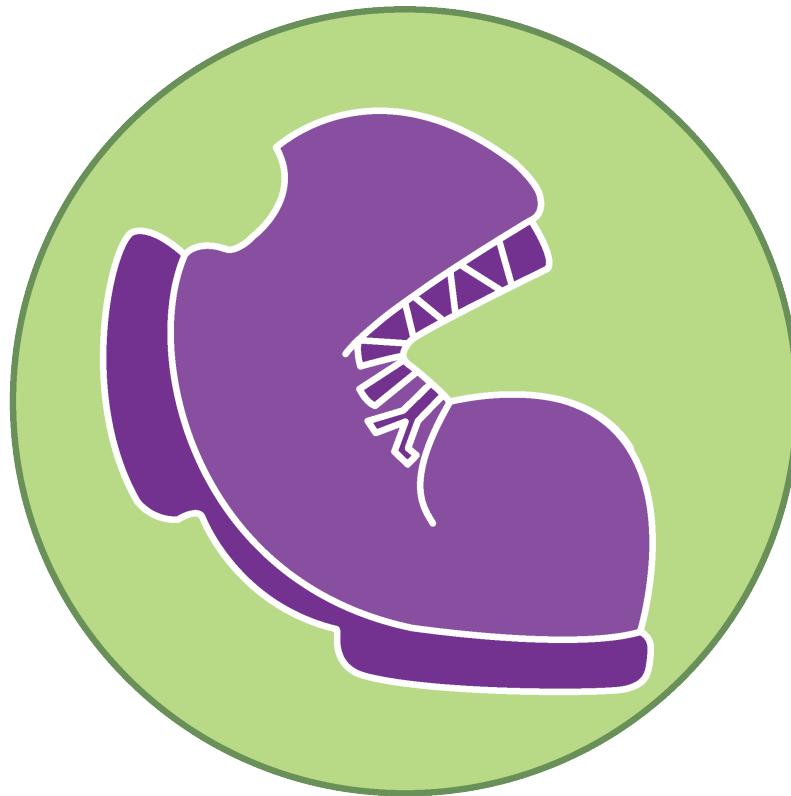


Name: \_\_\_\_\_



# **BOOTSTRAP:2**

---

[www.bootstrapworld.org](http://www.bootstrapworld.org)

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

# Lesson 1

	Racket Code	Pyret Code
<i>Numbers</i>	<pre>(define AGE 14)  (define A-NUMBER 0.6)  (define SPEED -90)</pre>	<pre>AGE = 14  A-NUMBER = 0.6  SPEED = -90  Two of your own:</pre> <hr/> <hr/>
<i>Strings</i>	<pre>(define CLASS "Bootstrap")  (define PHRASE "Coding is fun!")  (define A-STRING "2500")</pre>	<pre>CLASS = "Bootstrap"  PHRASE = "Coding is fun!"  A-STRING = "2500"  Two of your own:</pre> <hr/> <hr/>

	<pre>(define SHAPE   (triangle 40 "outline" "red"))  (define OUTLINE   (star 80 "solid" "green"))  (define SQUARE   (rectangle 50 50 "solid" "blue"))</pre>	<pre>SHAPE =   triangle(40, "outline", "red")  OUTLINE =   star(80, "solid", "green")  SQUARE =   rectangle(50, 50, "solid", "blue")  One of your own:</pre> <hr/>
<i>Booleans</i>	<pre>(define BOOL true)  (define BOOL2 false)</pre>	<pre>BOOL = true  One of your own:</pre> <hr/>
<i>Functions</i>	<pre>; double : Number -&gt; Number ; Given a number, multiply by ; 2 to double it  (EXAMPLE (double 5) (* 2 5)) (EXAMPLE (double 7) (* 2 7))  (define (double n) (* 2 n))</pre>	<pre># double : Number -&gt; Number # Given a number, multiply by # 2 to double it  examples:   double(5) is 2 * 5   double(7) is 2 * 7 end  fun double(n):   2 * n end</pre>

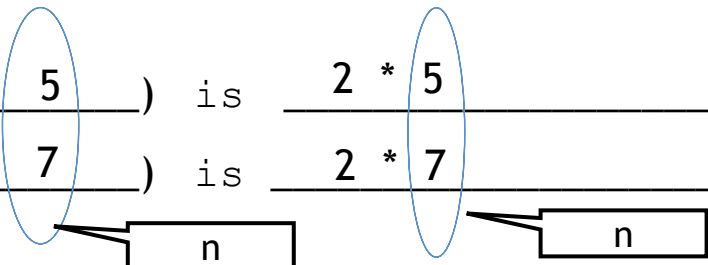
## Fast Functions!

Fill out the contract for each function, then try to write two examples and the definition by yourself.

# double : Number -> Number  
name domain range

examples:

double ( 5 ) is 2 \* 5  
double ( 7 ) is 2 \* 7  
end



fun double ( n ):

2 \* n

end

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

examples:

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

end

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

\_\_\_\_\_

end

## Fast Functions!

Fill out the contract for each function, then try to write two examples and the definition by yourself.

```
# _____ : _____ -> _____  
      name      domain      range
```

examples:

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

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

end

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

\_\_\_\_\_

end

```
# _____ : _____ -> _____  
      name      domain      range
```

examples:

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

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

end

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

\_\_\_\_\_

end

## Fast Functions!

Fill out the contract for each function, then try to write two examples and the definition by yourself.

```
# _____ : _____ -> _____  
      name          domain          range
```

examples:

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

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

end

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

\_\_\_\_\_

end

```
# _____ : _____ -> _____  
      name          domain          range
```

examples:

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

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

end

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

\_\_\_\_\_

end

# Bug Hunting: Pyret Edition

#1	<pre>SECONDS = (7)  STRING = my string</pre>	<hr/> <hr/> <hr/>
#2	<pre>SHAPE1 = circle(50 "solid" "blue")  SHAPE2 = triangle(75, outline, yellow)</pre>	<hr/> <hr/> <hr/>
#3	<pre># triple : Number -&gt; Number # Multiply a given number by # 3 to triple it  examples:   triple(5) = 3 * 5   triple(7) = 3 * 7 end</pre>	
#4	<pre>fun triple(n):   3 * n</pre>	
#5	<pre># ys : Number -&gt; Number # Given a number, create a solid # yellow star of the given size  examples:   ys(99) is star(99, "solid", "yellow")   ys(33) is star(99, "solid", "yellow")  ys(size):   star(size "solid" "yellow") end</pre>	

# Lesson 2

[illegible]



# Word Problem: double-radius

Write a function *double-radius*, which takes in a radius and a color. It produces an outlined circle of whatever color was passed in, whose radius is twice as big as the input.

## Contract+Purpose Statement

Every contract has three parts:

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name Domain Range

# \_\_\_\_\_  
What does the function do?

## Give Examples

Write examples of your function in action

examples:

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

...which should become

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

...which should become

end

## Function

Circle the changes in the examples, and name the variables.

Write the code, copying everything that isn't circled, and using names where you find variables!

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

end

# Word Problem: double-width

Write a function *double-width*, which takes in a number (the length of a rectangle) and produces a rectangle whose width is twice the given length.

## Contract+Purpose Statement

Every contract has three parts:

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name Domain Range

# \_\_\_\_\_  
What does the function do?

## Give Examples

Write examples of your function in action

examples:

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

...which should become

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

...which should become

end

## Function

Circle the changes in the examples, and name the variables.

Write the code, copying everything that isn't circled, and using names where you find variables!

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

end

# Word Problem: next-position

Write a function *next-position*, which takes in two numbers (an x and y-coordinate) and returns a Coord, increasing the x-coordinate by 5 and decreasing the y-coordinate by 5.

## Contract+Purpose Statement

Every contract has three parts:

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name Domain Range

# \_\_\_\_\_  
What does the function do?

## Give Examples

Write examples of your function in action

examples:

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

...which should become

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

...which should become

end

## Function

Circle the changes in the examples, and name the variables.

Write the code, copying everything that isn't circled, and using names where you find variables!

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

end

# Data Structure

# a Cake is a **flavor, color, message, layers, & is-iceCream**  
data **Cake**:

```
|  cake(_____
      _____
      _____
      _____
      _____)
```

end

To make examples of this structure, I would write:

**cake1** = \_\_\_\_\_

**cake2** = \_\_\_\_\_

To access the fields of **cake2**, I would write:

```
_____
_____
_____
_____
_____
```

# Lesson 3

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# Data Structure

# a Party is a **location, theme, and number of guests**

data **Party**:

```
|   party(_____  
          _____  
          _____)
```

end

To make examples of this structure, I would write:

**party1** = \_\_\_\_\_

**party2** = \_\_\_\_\_

To access the fields of **party2**, I would write:

```
_____  
_____  
_____
```

## Word Problem: change-flavor

Write a function called *change-flavor*, which takes in a Cake and a flavor, and returns a new Cake that is almost the same as the original, but is now the given flavor.

### Contract+Purpose Statement

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

# \_\_\_\_\_

### Give Examples

examples:

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

---

---

---

---

---

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

---

---

---

---

---

end

### Function

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

---

---

---

---

---

end

## Word Problem: will-melt

Write a function called *will-melt*, which takes in a Cake and a temperature, and returns true if the temperature is less than 32 degrees, AND the Cake is an ice cream cake.

### Contract+Purpose Statement

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

# \_\_\_\_\_

### Give Examples

examples:

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

---

---

---

---

---

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

---

---

---

---

---

end

### Function

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

---

---

---

---

---

end



# Lesson 4

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# Word Problem: update-world (Ninja World)

## Contract+Purpose Statement

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

# \_\_\_\_\_

## Give Examples

examples:

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

---

---

---

---

---

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

---

---

---

---

---

end

## Function

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

---

---

---

---

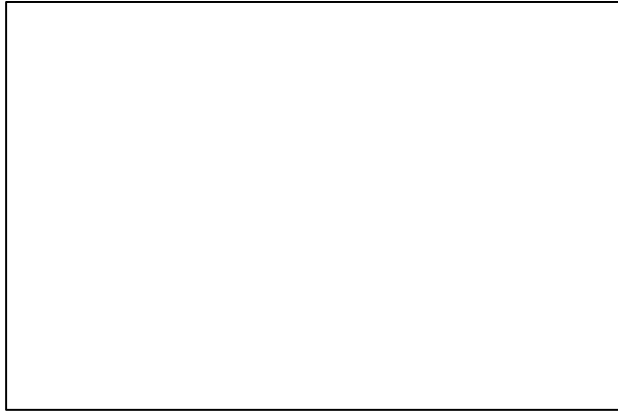
---

end

# GAME DESIGN

*“Start Simple, Get Complex”*

Draw a rough sketch of your game when it begins, and another sketch just a moment later



*A sketch at the START of the game...*



*A sketch for the very NEXT moment...*

What images will you need for your game? Name them in the 1<sup>st</sup> column, and describe them in the 2<sup>nd</sup>

BACKGROUND	

List everything that has changed from one sketch to the other. What datatype will represent it?

Changed (position, score, color, costume...)	Datatype (Number, String, Image, Boolean...)

# Data Structures

```
# a world is a _____  
data World:  
    | world( _____  
        _____  
        _____  
        _____  
        _____ )  
end
```

To make example worlds that represent my START and NEXT sketches from page 17, I would write...

**START** = \_\_\_\_\_

**NEXT** = \_\_\_\_\_

To access the fields of **START**, I would write:

---

---

---

---

---

# Lesson 5

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# Word Problem: draw-world

Contract

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

Definition

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

put-image( \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

end

# Word Problem: update-world

State the problem (What changes?):

## Contract+Purpose Statement

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
# \_\_\_\_\_

## Give Examples

examples:

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

---

---

---

---

---

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

---

---

---

---

---

end

## Function

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

---

---

---

---

---

end

# Lesson 6

[illegible]This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



# Word Problem: keypress (Ninja World)

## State the Problem

For each keypress in Ninja World, show how (keypress <world > <key>) should change the world.

## Contract+Purpose Statement

#	keypress	:	World	String	->	World
---	----------	---	-------	--------	----	-------

# Given a world and a key, produce a new world with NinjaCat's position  
# moved by 10 pixels, depending on which arrow key was pressed

## Give Examples

examples:

keypress(START, "up") is  
world(START.dogX, START.coinX, START.catX, START.catY + 10)

keypress(START, "down") is  
world(START.dogX, START.coinX, START.catX, START.catY - 10)

keypress(NEXT, "left") is  
world(NEXT.dogX, NEXT.coinX, NEXT.catX - 10, NEXT.catY)

keypress(NEXT, "right") is  
world(NEXT.dogX, NEXT.coinX, NEXT.catX + 10, NEXT.catY)

end

```

fun keypress(current-world, key) :

  ask:
    | string-equal(key, "up") then:

      world(current-world.dogX, current-world.coinX,
        current-world.catX, current-world.catY + 10)

    | string-equal(key, "down") then:

      world(current-world.dogX, current-world.coinX,
        current-world.catX, current-world.catY + 10)

    | string-equal(key, "left") then:

      world(current-world.dogX, current-world.coinX,
        current-world.catX - 10, current-world.catY)

    | string-equal(key, "right") then:

      world(current-world.dogX, current-world.coinX,
        current-world.catX + 10, current-world.catY)

    | otherwise: current-world

  end
end

```

## Word Problem: keypress (My game)

For each keypress in your game, show how (keypress START <key>) should change your world.

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

# \_\_\_\_\_

### Give Examples

examples:

keypress(START, \_\_\_\_\_) is

---

---

---

---

---

keypress(START, \_\_\_\_\_) is

---

---

---

---

---

keypress(START, \_\_\_\_\_) is

---

---

---

---

---

end

## Function

```
fun _____(_____)
  ask:
    | _____ then:
      _____
    | _____ then:
      _____
    | _____ then:
      _____
    | _____ then:
      _____
    | _____ then:
      _____
    | _____ then:
      _____
```

end  
end

## Word Problem: red-shape

Write a function *red-shape*, which takes in the name of a shape (such as "circle", "triangle", "star", or "rectangle"), and draws that shape. All shapes should be solid and red, and can be whatever size you choose.

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

# \_\_\_\_\_

### Give Examples

examples:

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

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

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

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

end

### Function

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

ask:

| \_\_\_\_\_ then:

\_\_\_\_\_

| \_\_\_\_\_ then:

\_\_\_\_\_

| \_\_\_\_\_ then:

\_\_\_\_\_

| \_\_\_\_\_ then:

end

end

# Lesson 7

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

## Word Problem: strong-password

Websites have strict password requirements. Write a function *strong-password*, which takes in a username and password, and checks to make sure they aren't the same, and then checks the string-length of the password to make sure it is greater than 8 characters. The function should return a message to the user letting them know if their password is strong enough.

```
# _____ : _____ -> _____
```

```
# _____
```

### Give Examples

examples:

```
_____ ( _____ ) is
```

```
_____
```

```
_____ ( _____ ) is
```

```
_____
```

```
_____ ( _____ ) is
```

```
_____
```

end

### Function

```
fun _____ ( _____ ) :
```

```
  ask:
```

```
    | _____ then:
```

```
    _____
```

```
    | _____ then:
```

```
    _____
```

```
    | otherwise: _____
```

```
  end
```

end



# Building Your Helper Functions

```
# is-off-right : _____ -> _____
```

examples:

```
_____ (_____) is
```

```
_____
```

```
_____ (_____) is
```

```
_____
```

end

```
fun _____ (_____) :
```

```
_____
```

end

```
# is-off-left : _____ -> _____
```

examples:

```
_____ (_____) is
```

```
_____
```

```
_____ (_____) is
```

```
_____
```

end

```
fun _____ (_____) :
```

```
_____
```

end

```
# _____:_____ -> _____
```

examples:

```
_____ (_____) is
```

```
_____
```

```
_____ (_____) is
```

```
_____
```

end

```
fun _____ (_____) :
```

```
_____
```

end

```
# _____:_____ -> _____
```

examples:

```
_____ (_____) is
```

```
_____
```

```
_____ (_____) is
```

```
_____
```

end

```
fun _____ (_____) :
```

```
_____
```

end

## Using Helpers inside `update-world`:

How does the World structure change when....?

TEST	RESULT
	world( _____ _____ _____ _____ _____)
	world( _____ _____ _____ _____ _____)
	world( _____ _____ _____ _____ _____)
	world( _____ _____ _____ _____ _____)

TEST	RESULT
	world( _____ _____ _____ _____ _____)
	world( _____ _____ _____ _____ _____)
	world( _____ _____ _____ _____ _____)
	world( _____ _____ _____ _____ _____)

# Using Helpers inside draw-world:

What changes the *appearance* of your game?

TEST	RESULT
	put-image(_____ put-image(_____ put-image(_____ put-image(_____ put-image(_____ 
	put-image(_____ put-image(_____ put-image(_____ put-image(_____ put-image(_____ 
	put-image(_____ put-image(_____ put-image(_____ put-image(_____ put-image(_____ 
	put-image(_____ put-image(_____ put-image(_____ put-image(_____ put-image(_____ 

TEST	RESULT
	put-image(_____ put-image(_____ put-image(_____ put-image(_____ put-image(_____
	put-image(_____ put-image(_____ put-image(_____ put-image(_____ put-image(_____
	put-image(_____ put-image(_____ put-image(_____ put-image(_____ put-image(_____
	put-image(_____ put-image(_____ put-image(_____ put-image(_____ put-image(_____

# Lesson 8

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

Word Problem: line-length

Write a function called *line-length*, which takes in two numbers and returns the difference between them. It should always subtract the smaller number from the bigger one.

## Contract+Purpose Statement

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

# \_\_\_\_\_

Give Examples

examples:

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

---

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

---

end

## Function Header

```
fun _____ ( _____ ) :
```

function name                      variable names

\_\_\_\_\_

end

end



## Distance:

The Player is at (4, 2) and the Target is at (0, 5).

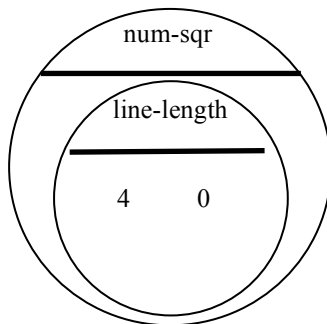
Distance takes in the player's x, player's y, character's x and character's y.

Use the formula below to fill in the EXAMPLE:

$$\sqrt{(\text{line-length } 4 \ 0)^2 + (\text{line-length } 2 \ 5)^2}$$

---

Convert it into a Circle of Evaluation. (We've already gotten you started!)



---

Convert it into Pyret code:

# Word Problem: distance

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:

$$\text{Distance}^2 = (\text{line-length } px \text{ } cx)^2 + (\text{line-length } py \text{ } cy)^2$$

## Contract+Purpose Statement

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

# \_\_\_\_\_

## Give Examples

Write examples of your function in action

examples:

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

\_\_\_\_\_

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

\_\_\_\_\_

end

## Function

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

\_\_\_\_\_

\_\_\_\_\_

end

## Word Problem: is-collision

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

It should return true if the coordinates of the player are within **50 pixels** of the coordinates of the other character. Otherwise, false.

### Contract+Purpose Statement

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_

# \_\_\_\_\_

### Give Examples

Write examples of your function in action

examples:

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

\_\_\_\_\_  
\_\_\_\_\_

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

\_\_\_\_\_  
\_\_\_\_\_

end

### Function

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

\_\_\_\_\_  
\_\_\_\_\_

end

# Supplemental

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

# DESIGN RECIPE

## Contract+Purpose Statement

Every contract has three parts:

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name Domain Range

# \_\_\_\_\_  
What does the function do?

## Give Examples

Write examples of your function in action

examples:

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

\_\_\_\_\_ ...which should become

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

\_\_\_\_\_ ...which should become

end

## Function

Circle the changes in the examples, and name the variables.

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

end

# DESIGN RECIPE

## Contract+Purpose Statement

Every contract has three parts:

# \_\_\_\_\_ : \_\_\_\_\_ -> \_\_\_\_\_  
name Domain Range

# \_\_\_\_\_  
What does the function do?

## Give Examples

Write examples of your function in action

examples:

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

\_\_\_\_\_ ...which should become

\_\_\_\_\_ ( \_\_\_\_\_ ) is  
the user types...

\_\_\_\_\_ ...which should become

end

## Function

Circle the changes in the examples, and name the variables.

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

end

# Contracts

Name	Domain	Range	example
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	

# Contracts

Name	Domain	Range	example
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	
#	:	↑	