

## The Design Recipe

For the word problems below, assume `animalA` and `animalB` are defined as the data rows for Felix and Midnight, respectively.

**Directions:** Define a function called `is-cat`, which consumes a `Row` of the `animals` table and *computes* whether the animal is a cat.

## Contract and Purpose Statement

*Every contract has three parts...*

#	is-cat::	( r :: Row)	->	Boolean
	<i>function name</i>	<i>domain</i>		<i>range</i>

```
# Consumes an animal, and computes whether the species == "cat"
```

---

*what does the function do?*

## Examples

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

examples:

`is-cat ( "animalA" ) is` \_\_\_\_\_  
*function name*      *input(s)*      *what the function produces*

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

end

## Definition

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

```

fun is-cat( r ):

```

---

*what the function does with those variable(s)*

end

**Directions:** Define a function called `is-young`, which consumes a Row of the `animals` table and *computes* whether it is less than four years old.

## Contract and Purpose Statement

*Every contract has three parts...*

# :: ->

function name domain range

# \_\_\_\_\_  
 \_\_\_\_\_  
*what does the function do?*

Examples 1

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

examples:

[illegible]

end

## Definition

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

**fun**                      (                      ) :

*function name*                      *variable(s)*

---

*what the function does with those variable(s)*

end