

## The Design Recipe

**Directions:** Define a function called `gt`, which makes solid green triangles of whatever size we want.

## Contract and Purpose Statement

Every contract has three parts...

# gt:: (size :: Number) -> Image  
*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, giving variable names to all your input values...

```
fun gt( size ):

```

<i>function name</i>	<i>variable(s)</i>
triangle(size, "solid", "green")	

---

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

end

**Directions:** Define a function called `bc`, which makes solid blue circles of whatever radius we want.

## Contract and Purpose Statement

Every contract has three parts...

The diagram illustrates the components of a function signature. It consists of three parts connected by symbols: a hash symbol (#), a double colon (::), and a right-pointing arrow (->). Below the hash symbol is the label "function name". Below the double colon is the label "domain". Below the right-pointing arrow is the label "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, giving variable names to all your input values...

```
fun (
```

function name      variable(s)

---

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

end