Contracts

Contracts tell us how to use a function. For example: `num-min :: (a :: Number, b :: Number) -> Number` tells us that the name of the function is `num-min`, it takes two inputs (both Numbers), and it evaluates to a `Number`. From the contract, we know `num-min(4, 6)` will evaluate to a `Number`. **Use the blank line under each contract for notes or sample code for that function!**

Name		Domain		Range
triangle	::	(side-length :: Number, style :: String, color :: String)	->	Image
circle	::	(radius :: Number, style :: String, color :: String)	->	Image
star	::	(radius :: Number, style :: String, color :: String)	->	Image
rectangle	::	(width :: Num, height :: Num, style :: Str, color :: Str)	->	Image
ellipse	::	(width :: Num, height :: Num, style :: Str, color :: Str)	->	Image
square	::	(size-length :: Number, style :: String, color :: String)	->	Image
text	::	(str :: String, size :: Number, color :: String)	->	Image
overlay	::	(img1 :: Image, img2 :: Image)	->	Image
beside	::	(img1 :: Image, img2 :: Image)	->	Image
above	::	(img1 :: Image, img2 :: Image)	->	Image
put-image	::	(img1 :: Image, x :: Number, y :: Number, img2 :: Image)	->	Image
rotate	::	(degree :: Number, img :: Image)	->	Image
scale	::	(factor :: Number, img :: Image)	->	Image