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
string-repeat	::	(text :: String, repeat :: Number)	->	String
string-repeat("cheetah ", 5)				
string-contains	::	(text :: String, search-for :: String)	->	Boolean
string-contains("rockstar", "star")				
num-sqr	::	(n :: Number)	->	Number
num-sqr(9)				
num-sqrt	::	(n :: Number)	->	Number
num-sqrt(25)				
num-min	::	(a :: Number, b:: Number)	->	Number
num-min(80, 20)				
num-max	::	(a :: Number, b:: Number)	->	Number
num-max(80, 20)				
count	::	(t :: Table, col :: String)	->	Table
count(animals-table, "species")				
mean	::	(t :: Table, col :: String)	->	Number
mean(animals-table, "age")				
median	::	(t :: Table, col :: String)	->	Number
median(animals-table, "age")				
modes	::	(t :: Table, col :: String)	->	List<Number>
modes(animals-table, "age")				
bar-chart	::	(t :: Table, col :: String)	->	Image
bar-chart(animals-table, "legs")				
pie-chart	::	(t :: Table, col :: String)	->	Image
pie-chart(animals-table, "species")				
histogram	::	(t :: Table, values :: String, bin-width :: Number)	->	Image
histogram(animals-table, "age", 2)				