# Contracts

Contracts tell us how to use a function. For example: `ellipse :: (Number, Number, String, String) -> Image` tells us that the name of the function is `ellipse`, it takes four inputs (two Numbers and two Strings), and it evaluates to an `Image`. From the contract, we know `ellipse(100, 50, "solid", "fuchsia")` will evaluate to an `Image`.

| Name | | Domain | | Range |
|------|------|--------|------|-------|
| # string-equal | :: | (String, String) | -> | Boolean |
| *string-equal("a tree", "trees")* | | | | |
| # | :: | | -> | |
| # | | | | |
| # | :: | | -> | |
| # | | | | |
| # | :: | | -> | |
| # | | | | |
| # | :: | | -> | |
| # | | | | |
| # | :: | | -> | |
| # | | | | |
| # | :: | | -> | |
| # | | | | |