Defining Values and Functions

- We can define values in our program, giving them names that we can refer to later instead of re-typing the same thing over and over. This works the same way it does in math: $x = 5 + 1$ defines the symbol x to be the number 6.
- In our language, we can define value by writing `(define x (+ 5 1))`. Here are a few value definitions:

```
(define x (+ 5 1))
(define y (* x 7))
(define food "Pizza!")
(define dot (circle y "solid" "red"))
```

- We can also define new **functions** in our language, to make it do things it didn't do before! To do this, we use a step-by-step process called the **Design Recipe**.
  - The first step is to write the **Contract** for the function you want to build. Remember, a Contract must include the Name, Domain and Range for the function!
  - Then we write a **Purpose Statement**, which is a short note that tells us what the function *should do*. Professional programmers work hard to write good purpose statements, so that other people can understand the code they wrote!
  - The second step is to write at least two **Examples**. These are lines of code that show what the function should do for a *specific* input. Once we see examples of at least two inputs, we can *find a pattern* and see which parts are changing and which parts aren't.
  - Circle the parts that are changing, and label them with a short **variable name** that explains what they do.
  - Finally, the third step is to define the function itself! This is pretty easy after you have some examples to work from: we copy everything that didn't change, and replace the changeable stuff with the variable name!