Warm-up (Do together on the board)

1. `3` is a…
    a. Number
    b. Function
    c. Variable
2. `+` is a …
    a. Number
    b. Function
    c. Variable
3. Draw the circles of evaluation for (3*4)+5
4. What is the corresponding Racket code? (Should probably decide on Racket vs. Scheme to stay consistent) `(+ (* 3 4) 5)`

Strings and Images

- More than just `Numbers`
    - Two new types: `Strings` and `Images`
    - `Strings` are surrounded by quotations (i.e. “solid”)
    - `Images` are shapes and/or pictures
- Try it out!
    - `(star 50 "solid" "purple")`
    - `(star (+ 1 3) "outline" "blue")`

Contracts

- From our experiments, we have seen that difference functions take different inputs
    - `star` takes in a `Number` and two `Strings`
    - `+` takes in two `Numbers`
- The expected inputs of the function are its domain
- The expected outputs of the function are its range
- Contracts for Functions (let’s the programmer know how to use them)
    - Name
    - Domain (number of things and their types)
    - Range (number of outputs and their types)
    - If we think of a language as a collection of lego pieces, the Contracts are like the tabs and slots that tell us how each piece can connect.
    - `; star: Number String String -> Image`
    - `; +: Number Number -> Number`
    - Try it out!
        - `; rectangle: Number Number String String -> Image`
            - What is the name, domain, and range?
        - Write the contract of `*`
- Function composition puts together multiple functions by matching up the domain and ranges

    - `(+ (* 3 4) 5)`
    - Like a puzzle where we match up the domain of one function with the range of another
    - Try it out!
        - Draw a circle of radius 25 using only the number 1-10
            - `(circle (+ 10 (+ 10 5)) "solid" "green")`
            - `(circle (* 5 5) "solid" "green")`
- Error Messages
    - Rectangle example
        - `(rectangle 50 "solid" "red")`
    - Circle example where fill and color are switched
        - `(circle 10 "green" "solid")`

Create a picture!

- Reminder overlay function contract
  - `; overlay: Image Image -> Image`
  - `(overlay`
  - `(star 25 "solid" "magenta")`
  - `(star 50 "solid" "blue"))`
  -
- Useful functions for making images
  - `; bitmap/url String -> Image` (String must be the url of an image)
  - `; star: Number String String -> Image`
  - `; rectangle Number Number String String -> Image`
  - `; circle Number String String -> Image`
  - `; triangle Number String String -> Image`
  - `; right-triangle Number Number String String -> Image`
  - `; rhombus Number Number String String -> Image`
  - `; radial-star Number Number Number String String -> Image`
- Useful functions of changing images
  - `; flip-vertical : Image -> Image`
  - `; flip-horizontal : Image -> Image`
  - `; scale : Number Image -> Image`
  - `; rotate : Number Image -> Image`
  - `; put-image : Image Number Number Image -> Image`

Link to code

http://www.wescheme.org/view?publicId=AsmT16xHhW