In the first lecture we used DrRacket interactively in the Interactions area:

Welcome to DrRacket, version 6.3.0.7 [3m].
Language: Intermediate Student; memory limit: 4096 MB.
>
That ">" means DrRacket is waiting for us to enter an expression.

What will DrRacket say when we enter:
> 123

What will DrRacket say when we copy and paste this star image into DrRacket and press enter/return:
> *

If we now enter the following we get an error message:
> scale
  scale: this variable is not defined

But we did use the scale operation in lecture, in the following form:
> (scale 2)
  scale: this function is not defined

That still says DrRacket is not recognizing the word 'scale'.
What did I forget to enter?
>
Now if we enter the word 'scale' what will DrRacket say:
> scale

To perform an operation we gather the operation and the things to operate on inside parentheses. The operation is always the first component.
We separate the components with single spaces:

> (scale 2)

What will DrRacket say when we enter:
> scale 2

If I enter the following I get an error message:
> (2 + 3)
  function call: expected a function after the open parenthesis, but found a number
How did I break the rules for performing an operation?

DrRacket does know something named ‘+’:

> +  
> +

How do you think we would write an expression to add 2 and 3?

> 

The operation that multiply numbers is named ‘*’. How do you think we would multiply 123 and 104:

> 

If I enter the following I get an error message:

> (*2 3)

*2: this function is not defined

How did I break the rules for performing an operation?

There is an operation that squares a number, called ‘sqr’. If I enter the following I get an error message:

> (sqr(3))

function call: expected a function after the open parenthesis, but found a number

What mistake did I make? How would we use 'sqr' to square the number 3:

> 

What would we enter to have DrRacket add 2 and 3 and then square that sum?

> 

What would we enter to have DrRacket square 2 and 3, and then add their squares?

> 

What will DrRacket say when we enter the following three images:

> 

In the first lecture we used the functions 'beside' and 'above' to combine images. They operate on two or more image arguments, for example:

> (beside

> (above

Notice the images are centred in the result image, which is especially noticeable for the small circle. Also notice there is no extra space between the images in the combined image result, which is especially noticeable for the star beside the circle. But space that was in a component image is still there, which is especially noticeable for the star above the circle.

Beside each of the following expressions show the result value:

> 

> 

>
Recall from lecture the function ‘rotate-cw’, which rotates an image clockwise:

> (rotate-cw 🐔)

To evaluate an expression that contains nested expressions, the inner expressions are evaluated first.

The expression `(above (rotate-cw 🐔) (rotate-cw 🐔))` is evaluated in the following steps:

1. `(above 🐔 (rotate-cw 🐔))`
2. `(above 🐔 🐔)`
3. `(above 🐔)`
4. `(above)`

Notice that each intermediate step is itself a valid expression.

What would be the intermediate expression step(s) and final result value for this expression:

`(rotate-cw (above 🐔 🐔))`

For each of the following expressions, show the result value, including any intermediate expression step(s).

1. `(above ♣ ♦ ♣)`
2. `(beside ♣ ♦)`
3. `(beside ♠ ♦)`
4. `(above (beside ♠ ♦) △)`
5. `(above ♠ (beside ♦ △))`
6. `(above ♠ ♦)`
7. `(beside (above ♠ ♦) (above △ ♦))`
Here are two more image functions being used:

> (flip-vertical)

> (flip-horizontal)

For each of the following expressions, show the result value, including any intermediate expression step(s).

(flip-vertical)

(above)

(beside (flip-vertical) (flip-horizontal))

(above (flip-horizontal) (flip-horizontal))

(flip-horizontal (beside))

(beside (flip-horizontal) (flip-horizontal))

(rotate-cw (flip-vertical))

(flip-vertical (rotate-cw))

(flip-horizontal (flip-vertical))

(flip-vertical (flip-horizontal))
Using only \( U \), numbers, and image functions, write expressions to produce the following values: