

CS 2500 - Lecture Notes 02

Wednesday, Jan 14

Overhead

Should have found the course site and done the reading by now.
Should have found the assignment by now.

Review

```
(+ 1 1)
(string-append "hello" "purple")
```

Naming Things

```
(define name thing)
(define three (+ 1 1))
(+ 1 three)
(define hello "Hello There")
```

Functions

```
(define (add-one x) (+ x 1))
(add-one 7)
(add-one three)
; comments
f(x) = 2x + 5? radius(c) = pi*r^2?
```

```
(define (join-words x y)
  (string-append x " " y))
(join-words "hello" "there")
```

Images

In Racket, you can paste images straight into your source code. They're a kind of value that we can work with just like numbers and strings.

The operations aren't built in, so we need to pull in a library:
(require 2htdp/image)

```
(define ROCKET ...)
```

Constant naming convention. The x in add-one is a variable.

```
(image-height ROCKET)
(image-width ROCKET)
```

```
(rectangle 100 50 "solid" "red")
```

```
(circle 40 "solid" "green")
```

```
(overlay ROCKET
```

```
  (circle (image-height ROCKET) "solid" "blue"))
```

Making donuts:

```
(overlay (circle 20 "solid" "white")
```

```
  (circle 40 "solid" "green"))
```

No donut command?

```
(define (donut size color)
```

```
  (overlay (... (/ size 2) ...) (... size ...))
```

```
(empty-scene 300 300)
```

```
(define dot (circle 20 "solid" "green"))
```

```
(define background (empty-scene 300 300))
```

```
(place-image dot 150 80 background)
```

```
tweak with 150, 80
```

Launching the Rocket

Now let's launch that rocket we had.

programming ~ creating the instructions that solve a (class of) problem(s)

computation ~ using the instructions for a specific problem

problem ~ we represent problems with data, numbers at first

We model problems with data. Models don't have to be correct, just useful. Putting a man on the moon used an already-obsolete 300 year old model of the physical rules of the world.

If we want to launch the rocket, we need to model its motion.

We need a function that calculates its position over time: $\text{height}(t) = f(t)$

Assume constant speed of 8 meters/second.

```
(define (height t) ...)
```

```
(height 0)
```

```
(height 40)
```

```
(check-expect (height 0) 0)
```

```
(check-expect (height 40) 320)
```

```
(define BG (empty-scene 300 800))  
(define (draw-rocket t)  
  (place-image ROCKET 150 (height t) BG))  
(draw-rocket 40)
```

```
(require 2htdp/universe)  
(animate draw-rocket)
```

Rocket went the wrong way. How do we fix it?