Conditionals

Some uses of **cond** where not based on the data definition:

```
(define (posn-big-part p)
  (cond
   [(> (posn-x p) (posn-y p)) (posn-x p)]
   [else (posn-y p)]))
```

For these, we use **if** in Java:

```
class Posn { ...
  double bigPart() {
    if (this.x > this.y)
      return this.x;
    else
      return this.y;
  }
}
```

Conditionals

In general:

```
(cond
 [question1 answer1]
 [question2 answer2]
 ...
 [else answerN])
```

⇒

```
if question1
   return answer1;
else if question2
   return answer2;
...
else
   return answerN;
```

Primitive Operations on Numbers

```
1 + 2   → 3
1 - 2   → -1
1 * 2   → 2
1 / 2   → 0
1.0 / 2.0   → 0.5

1 < 2   → true
1 > 2   → false
1 <= 2   → true
1 >= 2   → false
1 == 2   → false
1 == 1   → true
```

Primitive Operations on Booleans

```
!true → false
!false → true

true && true → true
true && false → false
true || false → true
false || false → false
```

Primitive Operations on Strings

```
"hello".equals("bye") → false
"hello".equals("hello") → true

"good".concat(" bye") → "good bye"

"good bye".startsWith("good") → true
"good bye".startsWith("bye") → false

"good bye".endsWith("good") → false
"good bye".endsWith("bye") → true
```

These operations are really method calls, and that's why **`String`** is capitalized