So far we learned to write simple programs. Our programs are sequences of definitions and expressions. We are very familiar with numerical expressions, e.g., \((+ 3 (* 2 -1))\) evaluating to \(\)\(\). But not all expressions involving numbers produce numbers: \((< 1 2)\) evaluates to: \(\). Such a value is called a \(\). Other expressions involve strings and string operators: sometimes they produce strings such as \((\text{substring "automaton" 2 8})\) but \((\text{string-length "meh"})\) evaluates to \(\), which is a number. Let's examine two other types of expressions. Assume \((\text{require 2htdp/image})\).

What does \((\text{empty-scene 200 200})\) evaluate to? Let's call the value \(a\).

What does \((\text{circle 30 "solid" "aquamarine"})\) evaluate to? Let's call the value \(b\).

What does \((\text{isosceles-triangle 70 80 "solid" "magenta"})\) evaluate to? Call this value \(c\).

What does \((\text{place-image b 20 60})\) evaluate to?

What does \((\text{place-image c 80 120})\) evaluate to? Call it \(d\).

What does \((\text{place-image (rectangle 10 10 "solid" "yellow") 70 160})\) evaluate to?

Assume \(\text{size}\) is a number, an integer given by the user (e.g., \(100, 200, 30\) or \(180\)). Define an expression that creates an empty scene of \(\text{size by size}\) pixels. In this empty scene place a yellow circle that has a radius of \(\frac{\text{size}}{8}\) pixels. Place the circle so the coordinates of the center are \(\left(\frac{\text{size}}{3}, \frac{3\text{size}}{4}\right)\). Run your expression for some value of \(\text{size}\). Then turn it into a function called \(\text{fun}\) that takes \(\text{size}\) as an argument and returns the value of the expression for the given \(\text{size}\). You're now ready for the second assignment.

Expressions involving pictures are not different from the other expressions involving strings and numbers. The operators are specific, but the essence is the same. Let's now examine the last type of expressions that we had planned to discuss today: expressions involving truth values.

What is \((\text{not} (< 1 2))\) and why? \(\) \(\text{not (negation)}\) is a unary operator that resembles the unary minus in arithmetic. \(\text{and (conjunction)}\) and \(\text{or (disjunction)}\) are binary operators that resemble multiplication and addition in arithmetic, even in their relative precedence as operators.

Let's look now at Exercise 9 in the book. Which of these two expressions are correct answers and why?

\[\text{(or (false? b1) (not (false? b2)))} \quad \text{and} \quad \text{(or (not b1) b2)}\]

Are the two expressions equivalent? Also: What does it mean to simplify an expression?

---

1 Proof that there's a \(\) in every automaton.
2 When finished evaluate \((\text{beside/align "bottom" (fun 160) (fun 120) (fun 280)})\)
3 http://www.ccs.neu.edu/home/matthias/HtDP2e/part_one.html
Simplify:

- (and (not a) a)
- (and a true)
- (and a false)
- (false? a)
- (if a false true)
- (and (< n 5) (< n 25))  
  assume n is an integer
- (or (< n 5) (< n 25))  
  assume n is an integer
- (or (> n 3) (< n 5))  
  assume n is an integer
- (if (< n 20) true (> n 10))  
  assume n is a number
- (and (or false false) true)  
  just evaluate this expression
- (or (and false false) true)  
  same thing (notice and and or are switched now)

Read about if/cond expressions in Fixed-Size Data section 4.1 Conditional Computations of your text.

Here are two problems to think about (we’ll work the first one together):

A year with 366 days is called a leap year. A year is a leap year if it is divisible by 4 (for example, the year 1980), except it is not a leap year if it is divisible by 100 (for example, the year 1900); however, it is a leap year if it is divisible by 400 (for example, the year 2000). There were no exceptions before the introduction of the Gregorian calendar on Oct 15, 1582 (for example, the year 1500 was a leap year).

Write a function leap of one argument (year, an integer) that determines if the year is leap or not.

Sketch the solution below:

Write a program that translates a number between 0 and 4 into the closest letter grade. For example, the number 2.8 (which might have been the average of several grades) would be converted to B-.

Break ties in favor of the better grade; for example 1.85 should be a C. Can you see why?

If not please let me know, also see below a diagrammatic hint on the associated decision process.

Sketch the solution below: