Goals of this tutorial
You should be able to...

• understand and perform Boolean algebra.
• understand and use conditional expressions.

Review: Boolean-valued functions
Boolean-valued functions produce Boolean values: true and false. These functions are also called predicates.

Standard Racket uses #t and #f, or #true and #false; these will sometimes show up in basic tests and correctness tests.

Racket provides many built-in Boolean functions (for example, to do numerical comparisons: (>= x y), (<= x y)).
Review: Boolean-valued functions

Note that comparison functions are often specific to certain data types (for example, (= a b) vs. (symbol=? x y), where a and b are numbers, but x and y are symbols).

The naming convention for most predicates and Boolean parameters is to append a question mark to the name (for example, even?, symbol?, expired?).

Review: Boolean Operators

and and or are special forms in Racket.

and and or may have two or more arguments.

Their arguments are evaluated from left to right.

and:

- If an argument evaluates to false, the entire expression evaluates to false.
- Otherwise, the next argument is evaluated.
- If there are no arguments remaining, the expression evaluates to true.

or:

- If an argument evaluates to true, the entire expression evaluates to true.
- Otherwise, the next argument is evaluated.
- If there are no arguments remaining, the expression evaluates to false.

not:

- not must have exactly one argument.
- If the argument evaluates to true, the entire expression evaluates to false.
- If the argument evaluates to false, the expression evaluates to true.
Clicker Question - Boolean Expressions
Which of the following expressions evaluates to true?
A \((= 'IAMSMART 'IAMSMART)\)
B \((\text{not} (\text{not} \text{false}))\)
C \((\text{check-expect} (\text{max} 10 (* 3 (\text{min} (+ 3 4) 8))) 21)\)
D \((\text{or} (\text{= 27 (expt 3 3)}) (< (\text{sqr} 5) 28) (+ (\text{sqr} 10) 5))\)
E \((\text{and} \text{true} (\text{not} \text{true}) (\text{not} (\text{not} \text{true}))))\)

Group Problem - valid-pin?
The Bank of Amestris has the following rules for setting a Bank PIN:

- A Bank PIN must be a 4-digit positive integer.
- For security reasons, a PIN cannot consist of the same digit appearing 4 times.

For example, 8242 is considered a valid Bank PIN, but 3333 is not. Using only boolean expressions, first write a helper function ith-digit that consumes a number n and an index i and produces the ith digit of n, counting from 0 and starting on the right. Include a purpose, contract, and examples.

Next, write the function valid-pin? that consumes a number, and produces true if the number is considered a valid PIN according to the rules above, and false otherwise. Include a purpose, contract, and examples.
Review: Conditional Expressions

The general form of a conditional expression is

\[
\text{cond}
\begin{align*}
&\quad \text{question1 answer1} \\
&\quad \text{question2 answer2} \\
&\quad \ldots \\
&\quad \text{questionk answerk}
\end{align*}
\]

where questionk could be else.

- Each of the questions must evaluate to a **boolean** value.
- The questions are evaluated from **top to bottom**.
- If a question evaluates to true, no more questions are evaluated and the cond expression is reduced to just the answer for that question.
- If none of the questions evaluate to true, then the result is the answer in the else clause.
- If there are no questions that evaluate to true and there is no else clause, then Racket will report an error.

Clicker Question - Cond Expression

What does the following cond expression evaluate to?

\[
\text{cond}
\begin{align*}
&\quad (< 180 180) \ 'yellow' \\
&\quad (\text{and (not false) (max 4 \(-6\) (sqr 2))) \ 'red'} \\
&\quad (\equiv 3 (\sqrt{9}) 1) \ 'yellow' \\
&\quad \text{else \ 'green'}
\end{align*}
\]

A. blue 
B. red 
C. yellow 
D. green 
E. Nothing. There is an error.
Group Problem - Converting cond to booleans
Rewrite the function f without cond. You may use and, or and not.

\[
\text{(define (f x)}
\text{ (cond}
\text{ [(not (p1? x)) (p2? x)]
\text{ [else (p1? x)]])}
\text{)}
\text{)
}\]

Shortening cond
Do not compare boolean variables with true/false as they can be used
directly as a condition. The examples below have the exact same behaviour:

\[
\text{(define (f x)}
\text{ (cond}
\text{ [[(boolean=\= (p1? x) true) (f1 x)]
\text{ [else (f2 x)]]]])}
\text{)}
\text{)
}\]

\[
\text{(define (f x)}
\text{ (cond}
\text{ [(p1? x) (f1 x)]
\text{ [else (f2 x)]])}
\text{)}
\text{)
}\]

Shortening cond
Nested cond expression can be shortened: the following are equivalent:

\[
\text{(define (f x)}
\text{ (cond [(p1? x) (cond [(p2? x) (f1 x)]
\text{ [else (f2 x)]])]
\text{ [else (f3 x)]])}
\text{)}
\text{)
}\]

\[
\text{(define (g x)}
\text{ (cond [(and (p1? x) (p2? x)) (f1 x)]
\text{ [(p1? x) (f2 x)]
\text{ [else (f3 x)]])}
\text{)}
\text{)}
\text{)
}\]
Group Problem - receives-discount/bool?
A warehouse store discounts its merchandise according to the following rules:

- All items in the clearance section are discounted.
- If an item has been in the store for at least 6 weeks, it is only discounted if the item is an 'appliance' or 'clothing.'
- If an item has been in the store for at least 3 weeks, but less than 6 weeks, it is only discounted if the item is a 'food.'
- All other items are not discounted.

Using only Boolean operations (and, or, or not), write a function receives-discount/bool? that consumes the number of weeks an item has been in the store, a symbol representing the type of the item, and a Boolean value representing whether the item is in the clearance section. The function produces true if the item receives a discount, and false otherwise. You only need to write the function definition.