Goals of this tutorial

You should be able to...

• understand and perform Boolean algebra.
• understand and use conditional expressions.
• write sufficient test cases for functions containing 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 \((\equiv \text{green} \text{green})\)
B \((\text{not} \text{not} \text{false})\)
C \((\text{check-expect} \text{min} (+ 3 4) 8) 7\)
D \((\text{or} (\equiv 81 \text{expt} 3 4) (< \text{sqr} 5 28))\)
E \((\text{and} \text{true} \text{not} \text{true})\)

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, write a 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

\((\text{cond}
    \begin{array}{c}
    \text{[question1 answer1]}
    \text{[question2 answer2]}
    \text{...}
    \text{[questionk answerk]}
    \end{array})\)

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?

```scheme
(cond
  [(< 18 18) 'blue]
  [(and (not false) (= (max 4 −6) (sqr 2))) 'red]
  [(= (/ 3 (sqrt 9)) 2) 'yellow]
  [else 'green])
```

A 'blue
B 'red
C 'yellow
D 'green
E Nothing. There is an error.

Note: Tests for conditional expressions
• Test for each clause in the cond expression.
• If the function specifications allow for this:
  – Test all boundary points.
  – Write at least one test for each interval (not including the boundary).
• DrRacket highlights unused code.
  – Having no code highlighted does not mean that your code is fully tested.
  – However, highlighted code means your testing is incomplete.
Clicker Question - Testing

\(\text{(define } (foo \ x)}\)
\(\text{(cond}\)
\(\text{[(< } \ x \ 0) (\exp \ x)]}\)
\(\text{[(<= } \ x \ 50) (\sub1 \ x)]}\)
\(\text{[(< } \ x \ 500) (\sqr \ x)]}\)
\(\text{[(< = } \ x \ 5000) (\div \ x \ 2)]}\)
\(\text{[else (add1 \ x)])}\)

Minimally, how many tests would be required for this function?

- A 5
- B 6
- C 7
- D 8
- E 9

Group Problem - receives-discount/cond?

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 \text{cond} and no Boolean operations (so no \text{and}, \text{or}, or \text{not}), write a function \text{receives-discount/cond?} 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 \text{true} if the item receives a discount, and \text{false} otherwise. Include the full design recipe.

Group Problem - receives-discount/bool?

Now write a function \text{receives-discount/bool?}, which consumes the same parameters as \text{receives-discount/cond?} but uses only Boolean operations, and no instances of \text{cond}. You only need to write the function definition. Here are the discount rules as a reminder:

- 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.
Group Problem - loan-interest

The Bank of Amestris also issues loans to its customers, which have either a ‘standard’ or ‘premium account. Their loan policies are as follows:

- If a customer has a bad credit history, the Bank refuses to give them a loan.
- Otherwise, if the duration of a loan is no more than 3 months, or the amount of money loaned is less than $500, no interest is paid on the loan.
- Otherwise, if a customer has a ‘premium account, and the loan amount is no more than $25,000, interest is calculated based on the ‘premium rate of 10%.
- Otherwise, interest is calculated on the ‘standard rate of 15%.

Interest is calculated by multiplying the amount of the loan, the duration, and the interest rate together. You may assume that the amount and duration of a loan are both positive.

Group Problem - loan-interest

Write a function loan-interest, which consumes the amount and duration of the loan in months, a Boolean which is true if the customer has a bad credit history, and a symbol representing the account-type. The function will produce the amount of interest that the customer must pay on the loan, or $-1$ if the Bank refuses to give a loan. You do not need to include a full set of tests. Here are the policies as a reminder:

- If a customer has a bad credit history, the Bank refuses to give them a loan.
- Otherwise, if the duration of a loan is no more than 3 months, or the amount of money loaned is less than $500, no interest is paid on the loan.
- Otherwise, if a customer has a ‘premium account, and the loan amount is no more than $25,000, interest is calculated based on the ‘premium rate of 10%.
- Otherwise, interest is calculated on the ‘standard rate of 15%.