

# Section Review

Learn to Code with Ruby

# The if Statement

- A condition is a criteria by which something happens.  
**Conditional code** is code that runs when a condition is met.
- The **if** statement executes a section of code when its condition evaluates to true.
- Remember to add an **end** keyword for every **if** keyword.

# Truthiness and Falsiness

- A **truthy** value is one that evaluates to true in a Boolean context (such as an **if** statement). A **falsy** value is one that evaluates to false in a Boolean context.
- **false** and **nil** are falsy values. Every other value in Ruby is truthy.

# The elsif and else Keywords

- After an **if**, we can add an **elsif** keyword to check for another condition. We can add as many **elsif** statements as we'd like. Place an **end** keyword at the end of the whole construct.
- The **else** keyword executes a chunk of code when all previous **if** and **elsif** statements evaluated to false.

# The AND && OR Operators

- The `&&` operator asserts that two conditions are both true.
- The `||` operator asserts that *either* of two conditions are true.
- When writing complex logic with multiple operators, use parentheses to establish precedence (importance).
- An alternative solution to tackle AND logic is by **nesting** if statements within other if statements.

# Ternary Operator

- The **ternary operator** simplifies an **if/else** statement into a one-liner.
- Write the condition, a question mark, the value to produce in the case of truthiness, a colon, and the value to produce in the case of falsiness.
- **condition ? TRUTHY VALUE : FALSY VALUE**

# The case Statement

- The **case** statement enables us to check for multiple conditions. Use it to consolidate multiple **if** statements.
- Use the **case** keyword followed by a starter value.
- Use the **when** keyword followed by a value to validate against. If there is a match, Ruby will execute the associated code chunk.
- Ruby stops execution once a match has been made.

# The unless Keyword

- The **unless** keyword is equivalent to writing "execute if false".
- It's generally considered better to use unless instead of the ! symbol.



# Statement Modifiers

- We can simplify **if** statements with a single line of logic to a one-line statement. With this option, we can skip the **end** keyword.
- puts "Huge!" if number > 5000

# Conditional Assignment

- The **conditional assignment operator** (`||=`) assigns a variable to a new value if the variable's current value is **nil**.
- `y ||= 10`