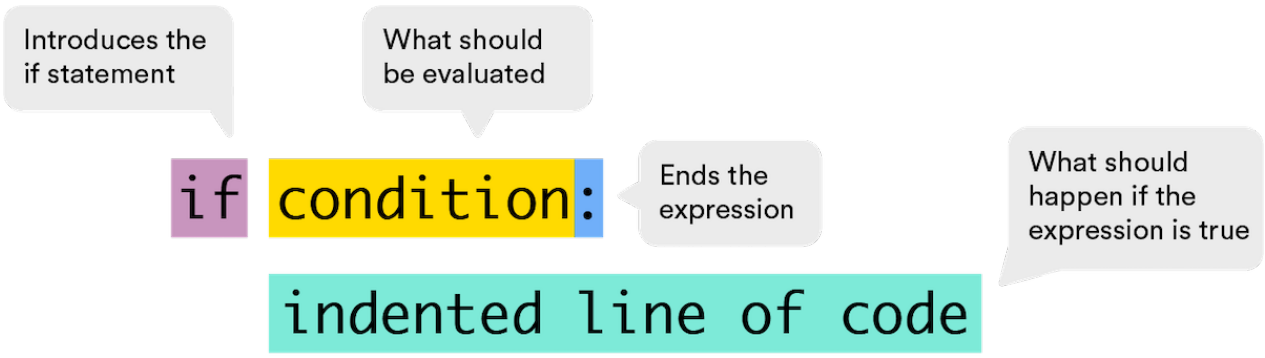
General Assembly

STUDY GUIDE

CONTROL FLOW IN PYTHON

Key Terms and Definitions

- **Expression:** Any valid piece of code that evaluates to something.
- **Control Flow:** Control flow is what keeps our programs from doing the same thing every single time. It tells our programs what to do based on different conditions.



- **if Statements:** Expressions that evaluate to a Boolean result (True or False) and test a conditional statement, such as comparing two objects. You can write code that operates a certain way only if the `if` statement evaluates to True. You can use the `elif` (else if) condition to link multiple `if` statements and the `else` condition to tell your code to do something different if the `if` statement evaluates to False.
- **Logical Operators:** Allow us to combine conditional statements. These are `and`, `or`, and `not`.
- **while Loops:** You can think about the `while` statement as a repeated `if` statement, which first evaluates the `while` expression before deciding whether to execute its code block. A `while` loop is generally used in Python if it is not known when the looping will stop or how many iterations the loop will require. The `while` loop is in many ways the most basic looping construct. All other loops can be rewritten as a while loop.
- **for Loops:** A `while` loop in which the "while" statement is automatically set to be a specific number of iterations.
- **range:** A built-in Python function that returns a list of numbers between given starting and ending points in defined increments.

Guiding Questions

1. Name a few situations in which you would use a `while` loop versus a `for` loop, and vice versa.
2. What do we mean when we say a piece of code "evaluates" to a result?
3. Think about how `if` statements relate to your decision-making process. What does the `if` statement for choosing your outfit in the morning look like (using `if`/`elif`/`else` to reach a conclusion)?