Module 2: Programming with Basic Logical Structures

Quick Reference Guide

Learning Outcomes

* Recognize and use basic data operators on Booleans.
* Interpret and write the correct syntax for conditionals.
* Recognize, interpret, and write programs with conditionals.
* Recognize and interpret programs with multiple interacting functions.
* Create programs to solve simple problems.
* Identify and solve programming errors through established debugging strategies.

Overview

A Boolean, and a new type of statement, a conditional will let the program make a choice between multiple actions based on provided data.​

Timeline

Description automatically generated

Advanced Functions

We can use keyword arguments to give the parameters in a function definition default values.

def say\_hi(name, title=""):

greeting = "Hi, " + title + " " + name

return greeting

Another interesting property of functions is that they can be designed to take any number of arguments, instead of a set number of arguments.​

result = print("Try this!")

print(result)

print("One", "Two", "Three")

If Statements

The most basic form of a conditional is an if statement. The expression immediately to the right of the 'if' keyword must evaluate to a Boolean.​ If that Boolean is True, Python runs through the code inside the if body normally.​ If the Boolean is False, the conditional body is skipped entirely.​ If the program does not enter the if body, it will immediately enter the else body instead.​ And if the program enters the if body, it will skip the else body entirely.​ Only one of the two sets of statements gets to run on any given program execution.​

​Diagram

Description automatically generated