|  |  |
| --- | --- |
| **Relational Operators** | |
| **frequently used operators** | |
| **Operator** | **What does it mean? How do you use it?** |
| **==** | Equal to |
| **>** | Greater than |
| **<** | Less than |
| **>=** | Greater than or equal to |
| **<=** | Less than or equal to |
| **!=** | Not equal to |

* All relational operators have equal priority and are lower than those of arithmetic operators
* A Boolean expression is an condition that can only be evaluated to true or false
* The if statement simply states that if a Boolean statement returns true, then a block of code is executed
* A Boolean Variable can only carry the value of “true” or “false”
* The not operator tells your program to read he code directly to its right as the opposite of what it says
* A module is a container for a group of related functions, variables, and classes
* Call a function from an imported module in the following way: moduleName.className.functionName(<parameters>)
* randint() function will generate random integers but you must import random first
* % finds the remainder of 2 numbers divided by each other. For example: 8 % 6 Output: 2

**Notes Questions:**

1. Linear programs run from top to bottom but programs with conditional logic can take input and execute a function based on the input
2. Conditional logic allows us to give different responses for different inputs
3. Boolean expressions give us tell us whether something is true or false. They provide the if statement.
4. You can use it to make something password protected… for example if the password is entered incorrectly, then access will be denied but if it is entered correctly then access will be granted

* Use “else” statement to cover all other conditions not covered by the if statement
* An else statement does not need a condition in parentheses because it covers all other conditions if none of the previous ones evaluate to true
* An elif statement is a combination of else and if. It adds exclusion from one condition to the next
* Nesting is when you place a condition inside of another condition, allowing a program to make a two-tiered decision
* Recursion is when a function calls itself to restart a sequence

1. Please fill out the following chart of logical operators in Python

|  |  |
| --- | --- |
| **and** |  |
| Both conditions must be met |  |
|  |  |
| **Or**  Either condition can be met |  |
| **Not**  Neither condition must be met |  |

1. **Explain nesting in your own words. Why is it useful?**

When you put one program inside of another and this is most useful when you are dealing with multiple conditions.

1. **Explain recursion in your own words. Why is it useful?**

Recursion restarts the program if there is a mistake with the user input. This is useful to make sure user input is in the right format.