

Decision Statements

CAC 180

Amber Wagner

Birmingham-Southern College

Decision Statements

- Also called **branching**
- Refers to the program executing one branch or another depending on a specific condition



if Statement

- One condition to be met, path only differs for that one condition

if expression:
task to do

- Notice colon and indention
- Example:

```
if needGas == True:  
    stopForGas  
driveToSchool
```

if-else Statement

- The tasks to be accomplished differ depending on a specific condition
- Example: Coffeehouse members are charged \$2.00 for a cup of coffee, non-members are charged \$3.00

```
if member == True:  
    coffee = 2.00  
else:  
    coffee = 3.00
```

A person is either
a member or not
a member, there is
no other option

Relational and Equality Operators

- Comparison operators

Relation and equality operators	Description
$a < b$	a is less-than b
$a > b$	a is greater-than b
$a \leq b$	a is less-than-or-equal-to b
$a \geq b$	a is greater-than-or-equal-to b
$a == b$	a is equal to b
$a != b$	a is not-equal to b

Comparisons

- Python supports operator chaining
- Ex: $a < b < c$
- Typically written as
 $a < b$ and $b < c$
- Evaluates left to right
- Numbers are compared arithmetically
- Strings are compared alphabetically (based on ASCII value)
- Lists and tuples are compared element by element
- Dictionaries are sorted and then the keys and values are compared as lists

if-elif Statements

- Sometimes you need to test more than one condition
- Classic example:
if grade \geq 90 and grade \leq 100:
 letterGrade = 'A'
else:
 letterGrade = 'Not an A'
- Will just an else suffice? How would you solve it?

if-elif Statement

- Syntax:

```
if expression:  
    task  
elif expression:  
    task  
else:  
    task
```

Can have as
many elif statements
as necessary

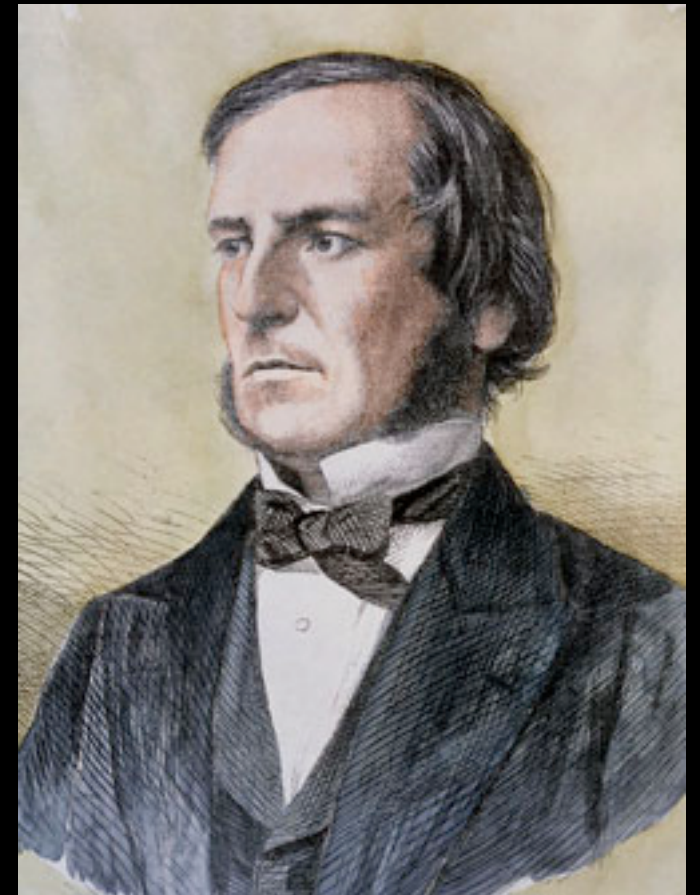
if-elif Statements

- Let's write the grade example together

Will a sequence of just if
statements work the
same way?

Boolean Operators

- A boolean is a true or false value
- Used in Boolean Logic (Boolean Algebra)
- Created by George Boole in 1850's - *The Laws of Thought* (1854)
- “No general method for the solution of questions in the theory of probabilities can be established which does not explicitly recognize, not only the special numerical base of the science, but also those universal laws of thought which are the basis of all reasoning, and which, whatever they may be as to their essence, are at least mathematical as to their form.”



Boolean Operators

Boolean operator	Description
a and b	Boolean AND: True when both operands are True
a or b	Boolean OR: True when at least one operand is True.
not a	Boolean NOT (opposite): True when the single operand is False (and False when operand is True).

- Truth Table

Helpful Operators

- Membership operator: in/not in
 - Can be used to test if a value is **in** or **not in** a list, tuple, or dictionary
- Identity operator: is/is not
 - Do not confuse with equality
 - This tests to see if two variable names are pointing to the same object

Practice

- Together: Let's write a program that allows the user to enter a year, and we print out if it is a leap year or not. A leap year is a year that is divisible by 4, BUT if it is divisible by 100, it must also be divisible by 400.
- You write: Write a program that determines the name of a shape from its number of sides. Read the number of sides from the user and then report the appropriate name as part of a meaningful message. Your program should support shapes with anywhere from 3 up to (and including) 10 sides. If a number of sides outside of this range is entered then your program should display an appropriate error message.

Next Class

- Continue reading and completing activities in Zybooks
- Loops on Wednesday!