Objectives

- Even more conditionals!
- Text process, manipulation
  - String operations, processing, methods

Review

- How do you tell Python to make decisions in code?
  - What are the different ways to express these decisions?

More Complex Conditions

- Boolean
  - Two logical values: True and False
- Combine conditions with Boolean operators
  - **and** – True only if **both** operands are True
  - **or** – True if **at least one** operand is True
  - **not** – True if the operand is not True
- English examples
  - If it is raining **and** it is cold
  - If it is Saturday **or** it is Sunday
  - If the shirt is on sale **or** the shirt is purple

What is the output?

Focus: how operations work
Not good variable names

```
x = 2
y = 3
z = 4

b = x==2
c = not b
d = (y<4) and (z<3)
print("d=",d)
d = (y<4) or (z<3)
print("d=",d)

d = not d
print(b, c, d)
```

Because of precedence, we don't *need* parentheses

eval_cond.py

Truth Tables

operands

A	B	A and B	A or B	not A	not B	not A and B	A or not B
T	T						
T	F						
F	T						
F	F						

Truth Tables

operands

A	B	A and B	A or B	not A	notB	not A and B	A or not B
T	T	T	T				
T	F	F	T				
F	T	F	T				
F	F	F	F				

Truth Tables

operands

A	B	A and B	A or B	not A	notB	not A and B	A or not B
T	T	T	T	F	F		
T	F	F	T	F	T		
F	T	F	T	T	F		
F	F	F	F	T	T		

Truth Tables

operands

A	B	A and B	A or B	not A	notB	not A and B	A or not B
T	T	T	T	F	F	F	T
T	F	F	T	F	T	F	T
F	T	F	T	T	F	T	F
F	F	F	F	T	T	F	T

Problem

- We are grading a science fair.
- There is different criteria for winning first place, depending on what grade you are.
- Given the variables `scienceScore` and `grade`
  - Write a condition that will evaluate to `True` if (and only if) the student's score is above the first place threshold of 60 points and the student's grade is 8.
    - Otherwise, the condition should evaluate to False

Problem

- We are grading a science fair.
- There is different criteria for winning first place, depending on what grade you are.
- Given the variables `scienceScore` and `grade`
  - Write a condition that will evaluate to `True` if (and only if) the student's score is above the first place threshold of 60 points and the student's grade is 8.
    - Otherwise, the condition should evaluate to False

```
scienceScore > 60 and grade == 8
```


Practice: Numeric Grade Input Range

- Enforce that user must input a numeric grade between 0 and 100
  - In Python, we can't (always) write a condition like 0 <= num_grade <= 100, so we need to break it into two conditions
- Write an appropriate condition for this check on the numeric grade
  - Using **and**
  - Using **or**

Practice: Numeric Grade Input Range

- Enforce that user must input a numeric grade between 0 and 100
  - Using **and**

```
if num_grade >= 0 and num_grade <= 100:
    computation
else:
    print error message
```

  - Using **or**

```
if num_grade < 0 or num_grade > 100:
    print error message
else:
    computation
```

Short-circuit Evaluation

- Don't necessarily need to evaluate all expressions in a compound expression
- A `and` B
  - If A is `False`, compound expression is `False`
- A `or` B
  - If A is `True`, compound expression is `True`
- No need to evaluate B
  - Put more important/limiting expression first
  - Example:

```
if count != 0 and sum/count > 10:
    do something
```


TEXT PROCESSING

Motivation: Text Processing

- Mostly focused on numbers so far
  - A little on graphics
- We can manipulate strings to do useful work
  - Web search: finding most relevant documents to a query
  - Analyzing web logs (who is looking at my web page?)
  - Many, many others
- **Today's Focus**: the `str` data type and what you can do with them

Strings: `str`

- Used for text
- Indicated by double quotes "" or single quotes ''
  - In general, I'll use double quotes
  - Empty string: "" or ''
- Use triple quotes """ for strings that go across multiple lines

```
"""This string
is long.
Like, really, really long"""
```

STRING OPERATIONS

String Operations

Operand	Syntax	Meaning
+	`str1 + str2`	Concatenate two strings into one string
*	`str * num`	Concatenate string `num` times

- Examples:
  - `"I feel " + "sleepy"`
    - Evaluates to `"I feel sleepy"`
  - `"Oops! " * 3`
    - Evaluates to `"Oops! Oops! Oops! "`

More Motivating Constants

- I have a survey program that asks people to rate something on a scale of 1 to 10
- It asks people to rate 100 different things
- I could create the prompt

```
"Rank " + thing + " on a scale of 1 to 10"
```

- But what if my scale changes, and I want it to be on a scale of 1 to 100?
  - I want to make sure the ranking is within my range

Practice

- Given the following code

```
SCALE_MIN = 1
SCALE_MAX = 10
prompt = …
rating = eval(input( prompt ))
```

- Create the **`str`** variable `prompt` for the `input` statement so that it prompts the user:

On a scale of 1 to 10, how much do you like Matt Damon?

String Comparisons

- Same operations as with numbers:
  - ==, !=
  - <, <=
  - >, >=

  } Alphabetical comparison

- Use in conditions in `if` statements

```
if courseChoice == "CSCI111":
    print("Good choice!")
else:
    print("Maybe next semester")
```


Looking Ahead

- Lab tomorrow
- Exam Friday
  - Preparation document on line
  - Content: up through Conditionals