Lecture 7:

CONDITIONAL STATEMENTS

CSC111: Introduction to CS through Programming

R. Jordan Crouser

Assistant Professor of Computer Science

Smith College

Announcements

- A note about getting credit for **labs**:
  - Turn in what you have **before** you leave the lab
  - **Both partners** must submit (your submissions can be identical)
  - If you want, you may revise/resubmit for up to 3 days (not required)

- A note about the **homework**:
  - Make sure that you follow the formatting instructions **exactly**
    - Enter first number
    - Enter second number
    - Enter operation
    - **`The answer is: <ANS>`**
  - Failure to do so will result in a delay in grading your assignment

Overview of the week

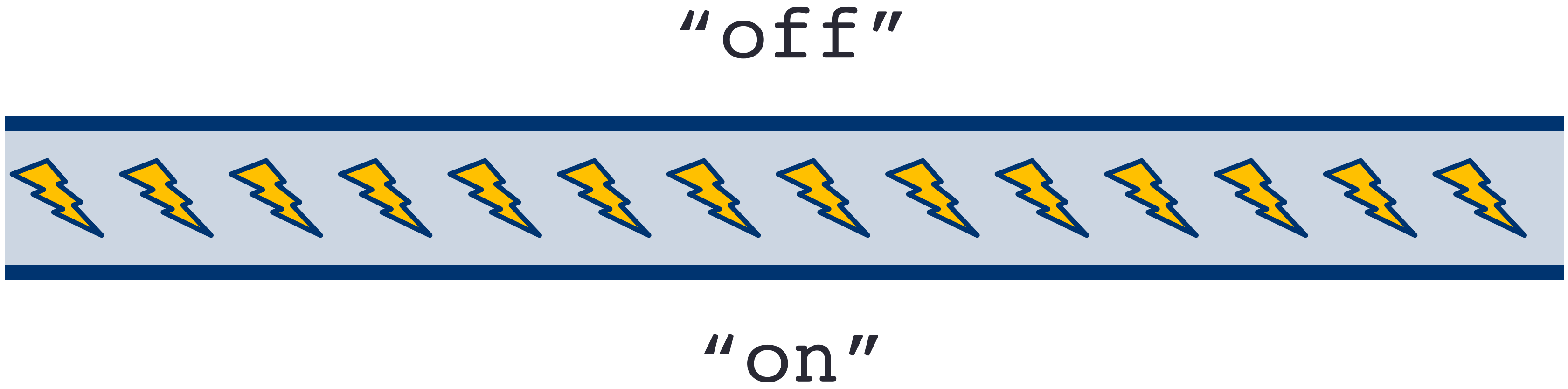
- ✓More about variables
- ✓Numeric values and basic operations
- ✓More mathematical operators
  - ✓Revisiting ints and floats
  - ✓The **`math`** module
  - ✓Formatting
- ✓Lab: Cash Machine
- Conditional ("if") statements

RECAP

How is information represented

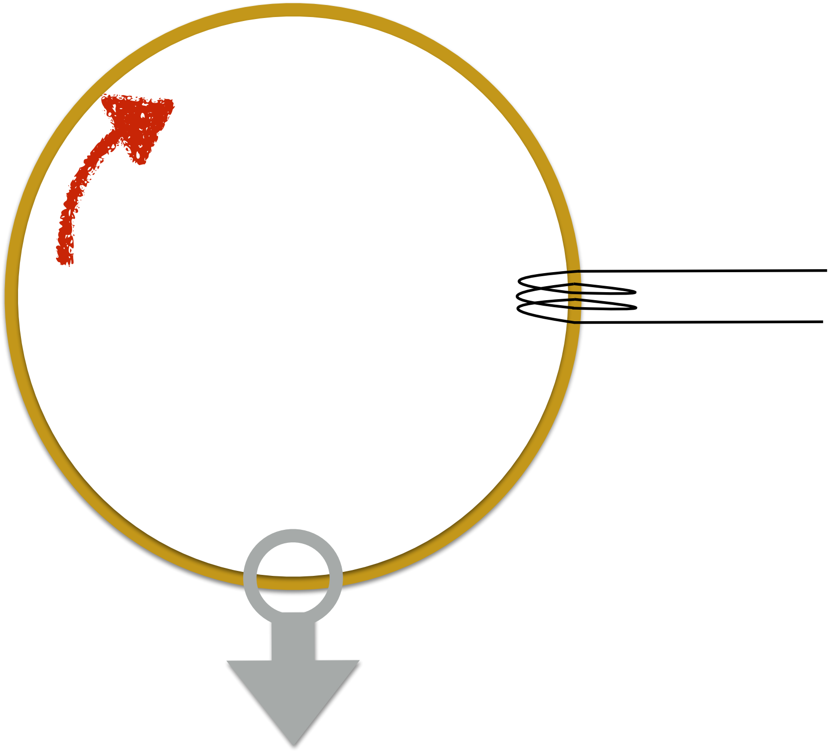
using **electricity**?

One wire: a “bit”



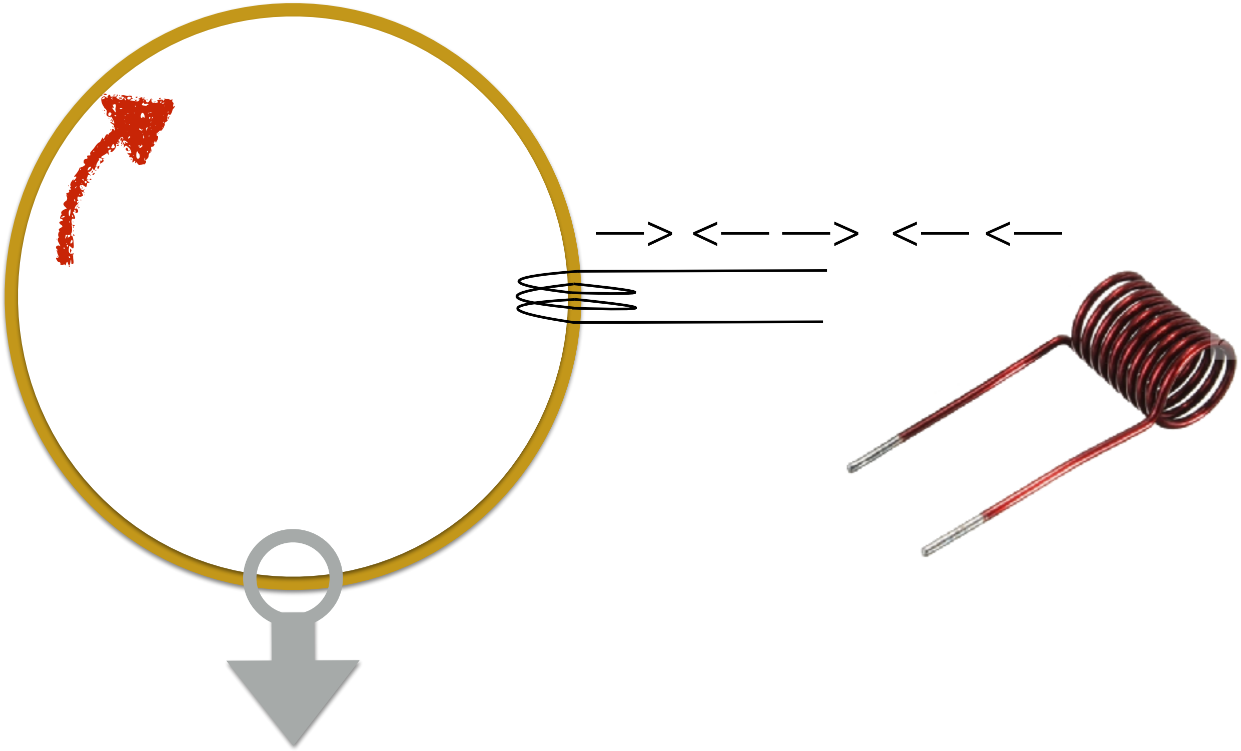
So how does a hard drive work?

RECAP: how does a hard drive work?



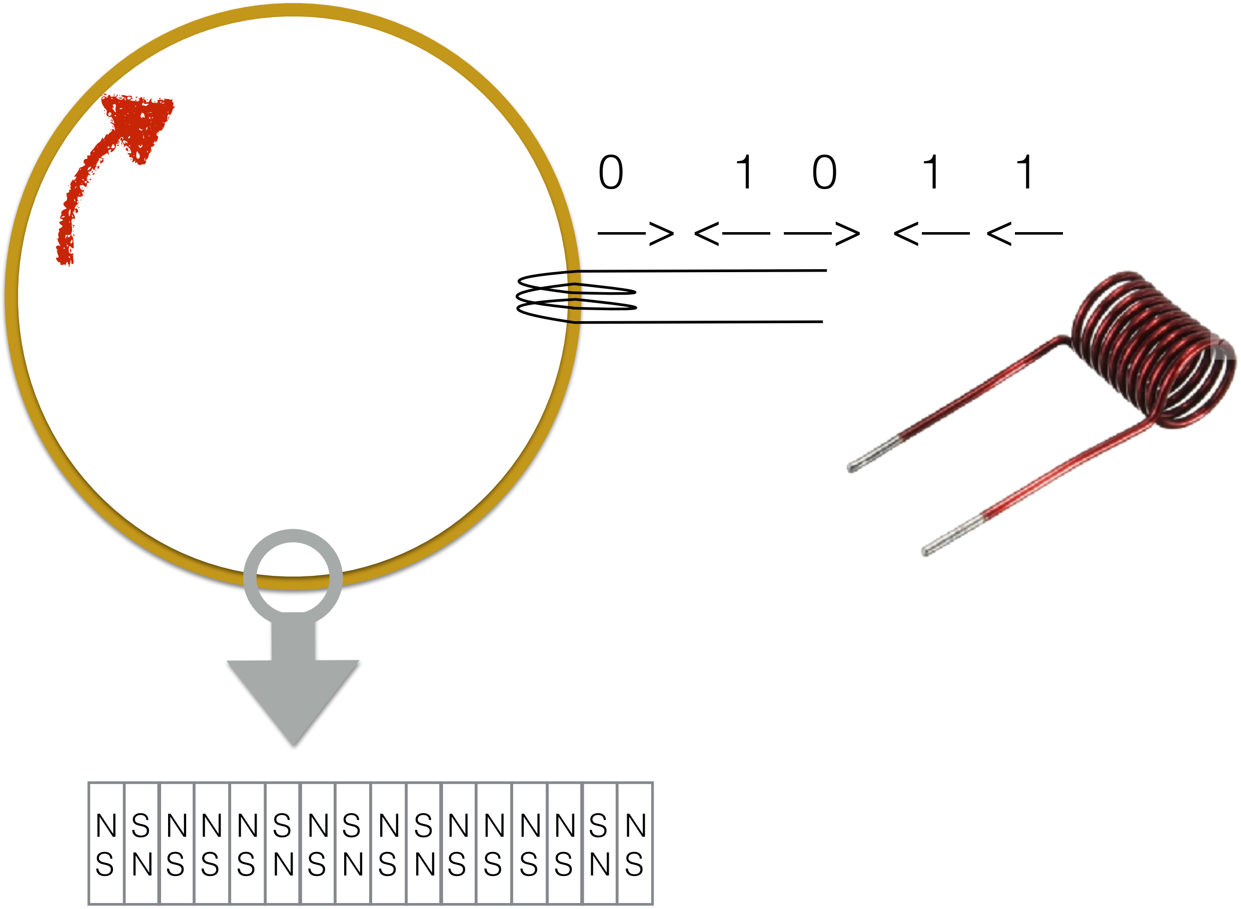
N	S	N	N	N	S	N	S	N	S	N	N	N	N	S	N
S	N	S	S	S	N	S	N	S	N	S	S	S	S	N	S

RECAP: how does a hard drive work?



N	S	N	N	N	S	N	S	N	S	N	N	N	N	S	N
S	N	S	S	S	N	S	N	S	N	S	S	S	S	N	S

RECAP: how does a hard drive work?



N	S	N	N	N	S	N	S	N	S	N	N	N	N	S	N
S	N	S	S	S	N	S	N	S	N	S	S	S	S	N	S

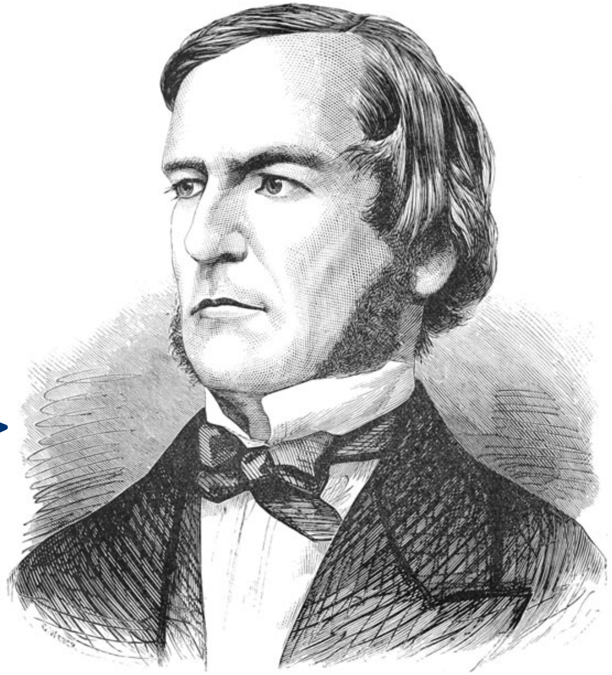
Bits and booleans

- **Bits**: 0 and 1
- **Boolean values**: **`True`** and **`False`**
- **Boolean switches**: imagine a world where every decision has a binary choice:

Go out or **stay in**?

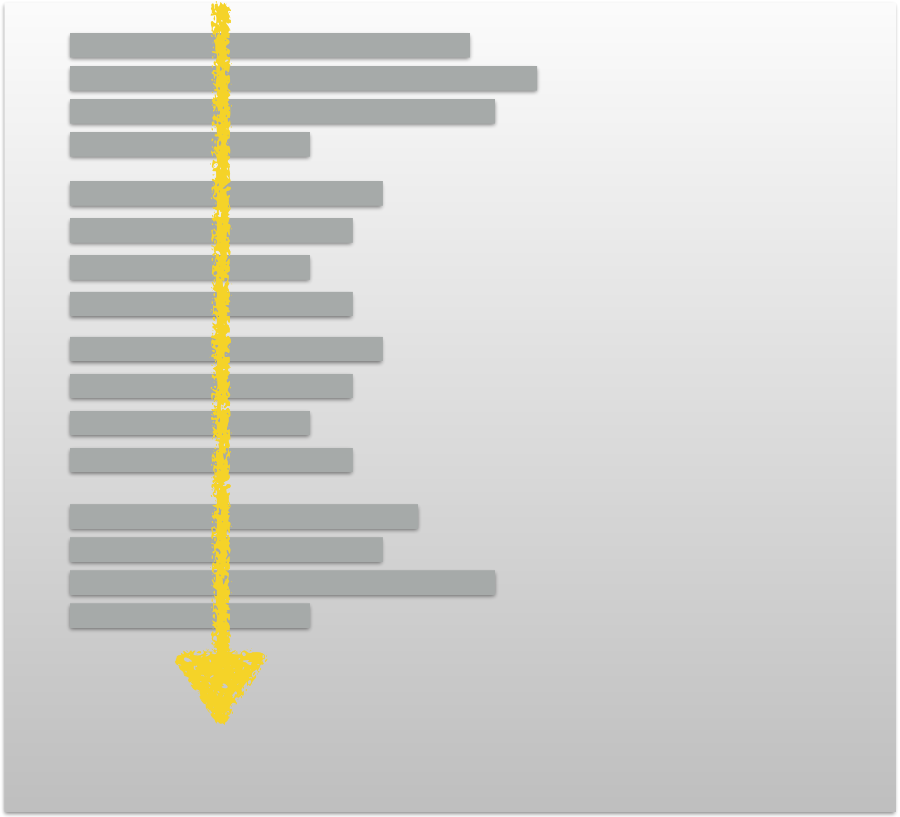
Walk or **take the car**?

Batman or **Spiderman**?

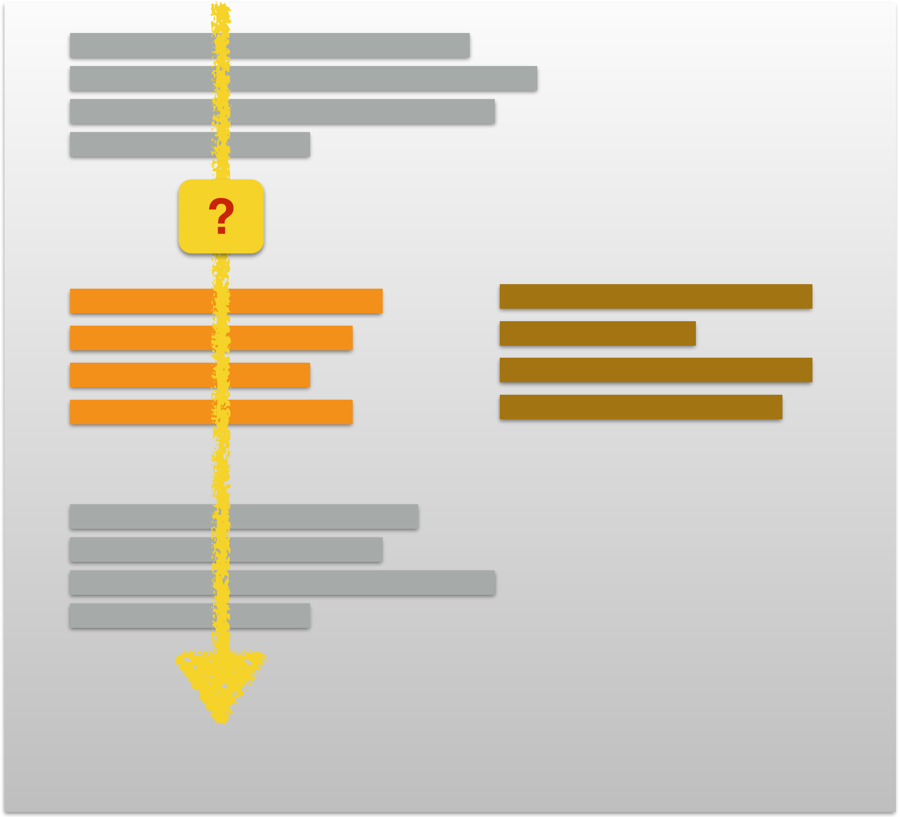


George Boole
1815 - 1864

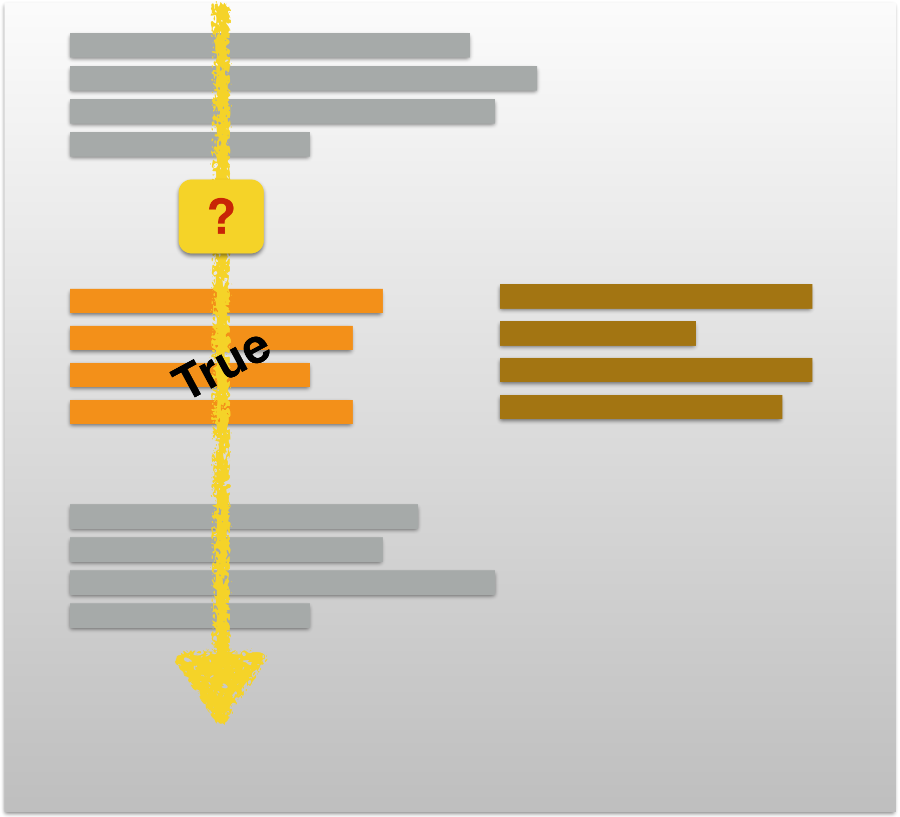
So far: linear programs



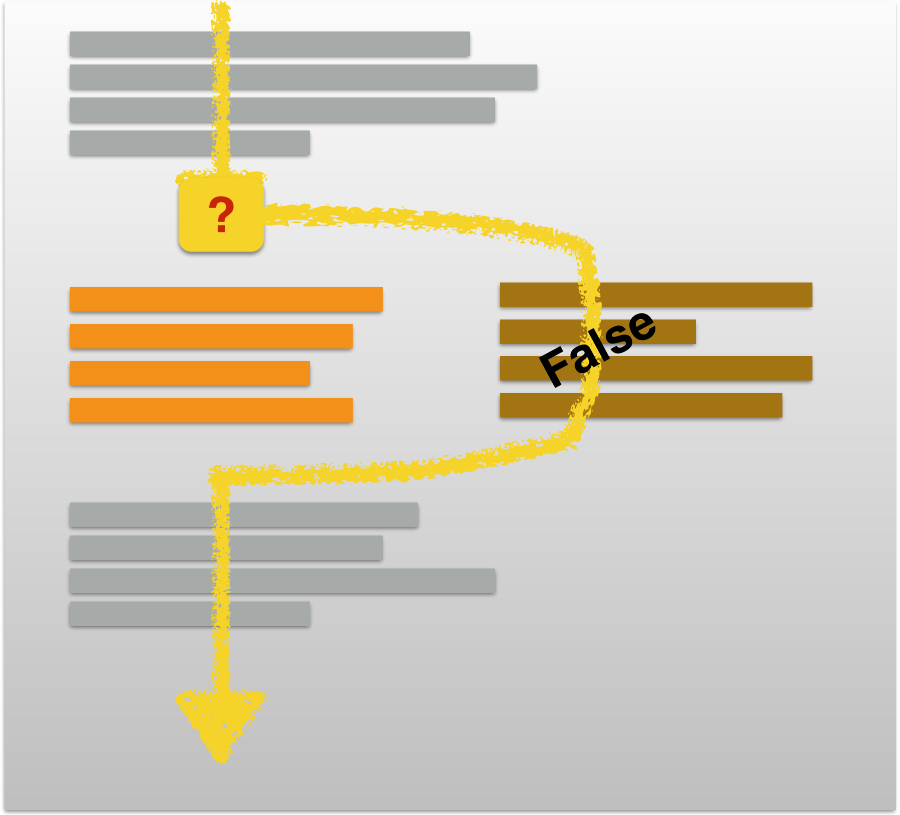
What if we need to make a **choice**?



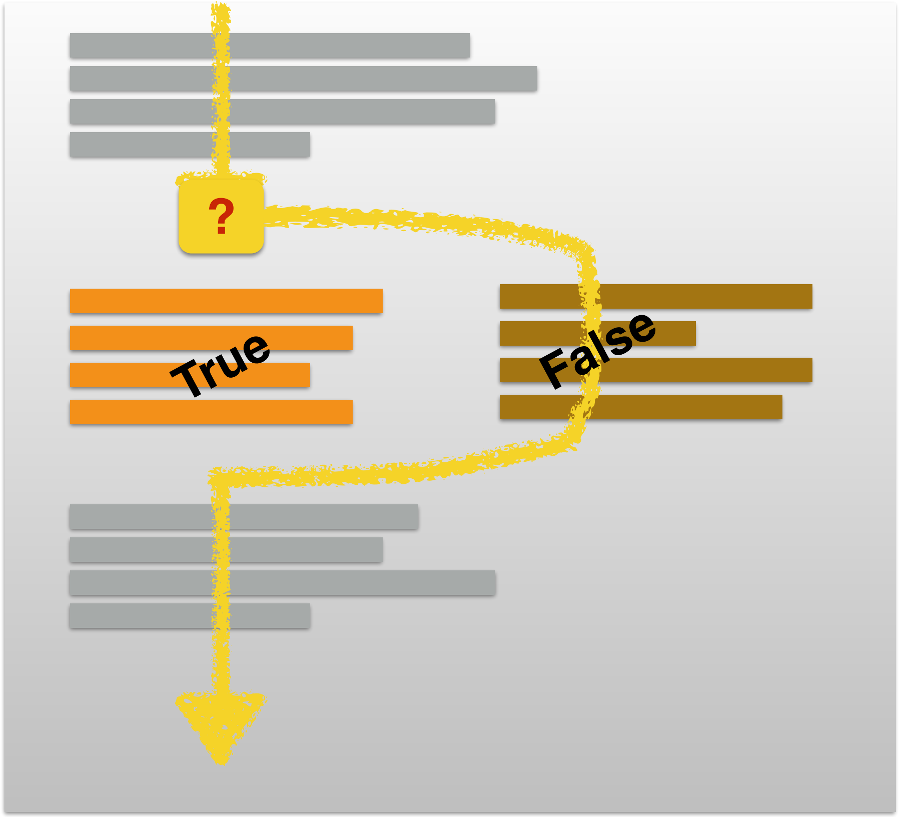
Booleans to the rescue!



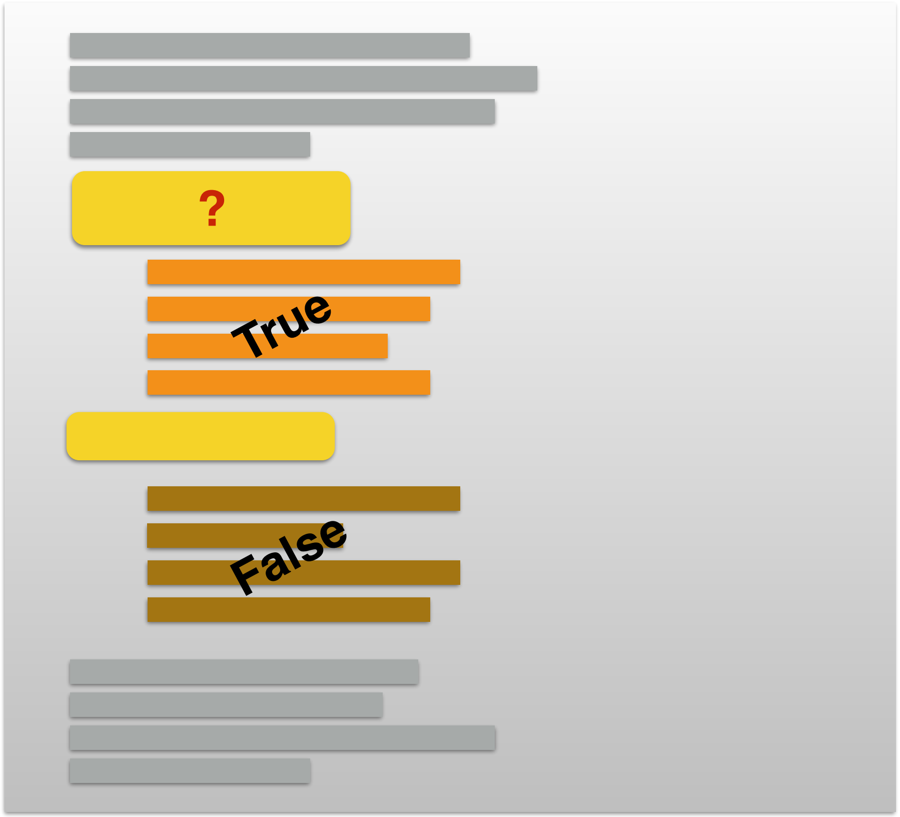
Booleans to the rescue!



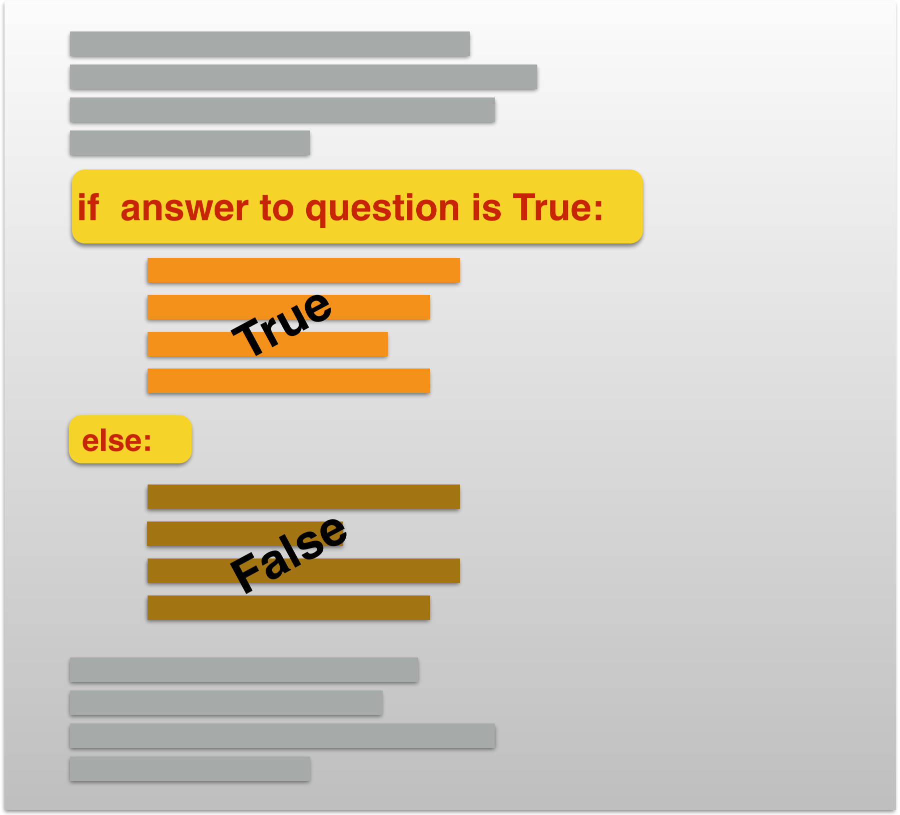
Just one **problem**: how do we write it?



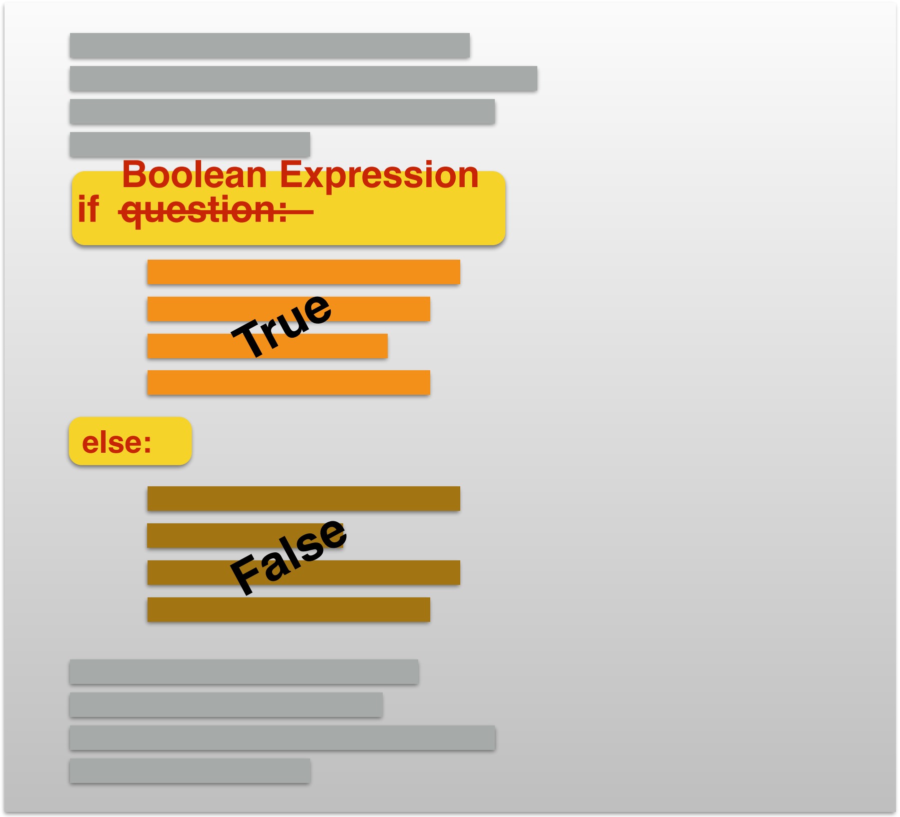
We can only type **one line** at a time...



What we **want** to say



What we have to **work with**



Real life examples (**pseudocode**)

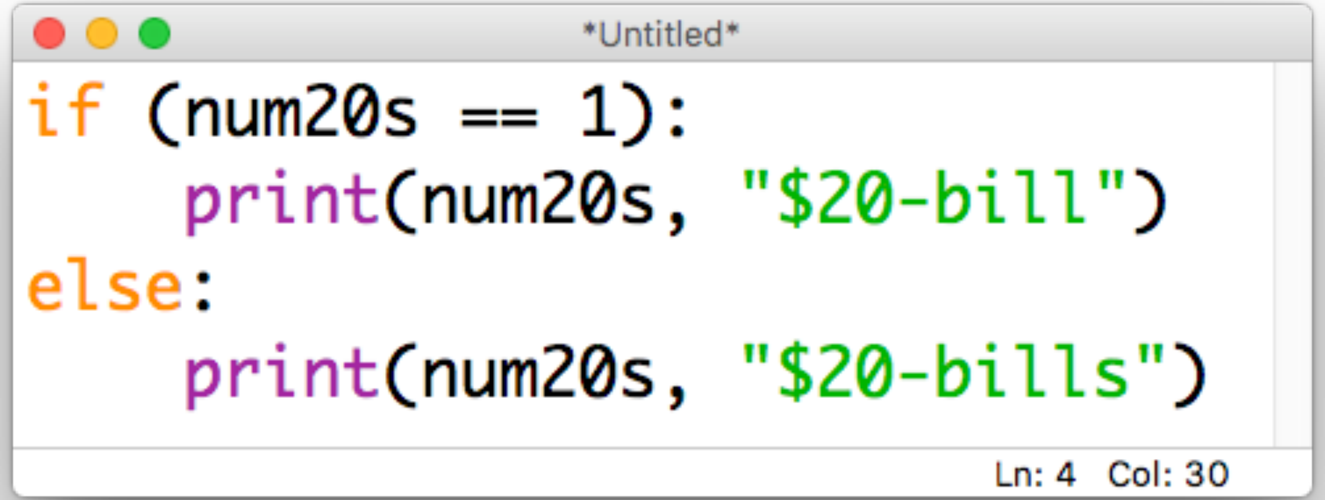
```
if (today is a weekday):
     go to class
else: # (today is a weekend)
     sleep in

if (food at Tyler looks good):
     eat at Tyler
else: # food at Tyler doesn’t look good
     order Domino’s
```

Real life example (our change machine)

```
Amount? 71

3 $20-bill(s)

1 $10-bill(s)

0 $5-bill(s)

1 $1-bill(s)
```

print the ”s”
only if necessary

Real life example (our change machine)



```
if (num20s == 1):
    print(num20s, "$20-bill")
else:
    print(num20s, "$20-bills")
```

Relational operators

Operator	Meaning
==	equal to
!=	not equal to
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

these come in handy when constructing **boolean statements**

Discussion: multiple conditions

```
if (it is sunny):
     go to the beach
if (it is snowy):
     go skiing
else:
     stay home
```

DEMO
TIME

Sequential **`if`** statements are **independent**

```
if (it is sunny):
     go to the beach
if (it is snowy):
     go skiing
else:
     stay home
```

The `else` refers only to the **nearest `if`**

```
if (it is sunny):
    go to the beach
if (it is snowy):
    go skiing
else:
    stay home
```

To chain multiple “checks” together: **`elif`**

evaluated in order

```
if (it is sunny):
     go to the beach
elif (it is snowy):
     go skiing
else: # it is neither sunny nor snowy
     stay home
```

To chain multiple “checks” together: **`elif`**

evaluated in order

```
if (it is sunny):
     go to the beach
elif (it is snowy):
     go skiing
else: # it is neither sunny nor snowy
     stay home
```

Remember: order matters!

evaluated in order

```
if (it is sunny): # regardless of snow
     go to the beach
elif (it is snowy): # but not sunny
     go skiing
else: # it is neither sunny nor snowy
     stay home
```

Remember: order matters!

evaluated in order

```
if (it is snowy): # regardless of sun
      go skiing
elif (it is sunny): # but not snowy
      go to the beach
else: # it is neither sunny nor snowy
      stay home
```

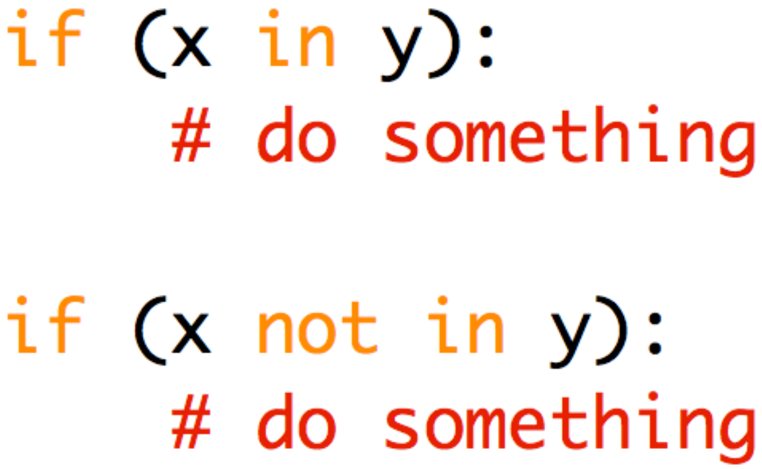
Nested conditions

```
if (class is cancelled):
    if (you have homework):
        work on homework
    else: # class cancelled, no HW
        binge-watch Netflix
```

Simultaneous conditions

```
if (it's Friday and it's 4pm):
      go to tea


if (you're hungry or you're bored):
      go to the CC
```

Boolean statements about sets

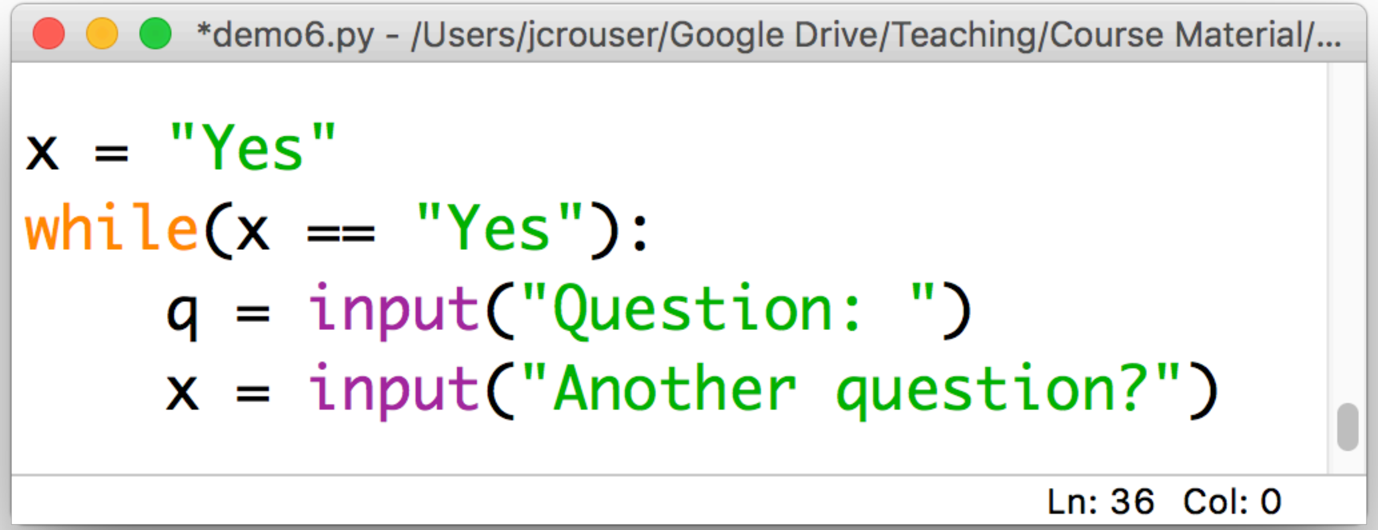


```
if (x in y):
    # do something

if (x not in y):
    # do something
```

DEMO
TIME

Side note: `while()` loops

- Boolean statements will also come in handy for controlling a new kind of loop, e.g.



*demo6.py - /Users/jcrouser/Google Drive/Teaching/Course Material/...

```python
x = "Yes"
while(x == "Yes"):
    q = input("Question: ")
    x = input("Another question?")
```

Ln: 36 Col: 0

Overview of the week

- ✓More about variables
- ✓Numeric values and basic operations
- ✓More mathematical operators
  - ✓Revisiting ints and floats
  - ✓The `math` module
  - ✓Formatting
- ✓Lab: Cash Machine
- ✓Conditional (“if”) statements

Coming up next

- *A2: Clunky Calculator* is due **Sunday 11:55pm**
- **Mon 9/24:** strings and string methods
- **Weds 9/26**: the **`main()`** method
- **Lab**: Pretty Printing
- **Fri 9/28**: Life skill #2: Debugging