[220 / 319] Conditionals
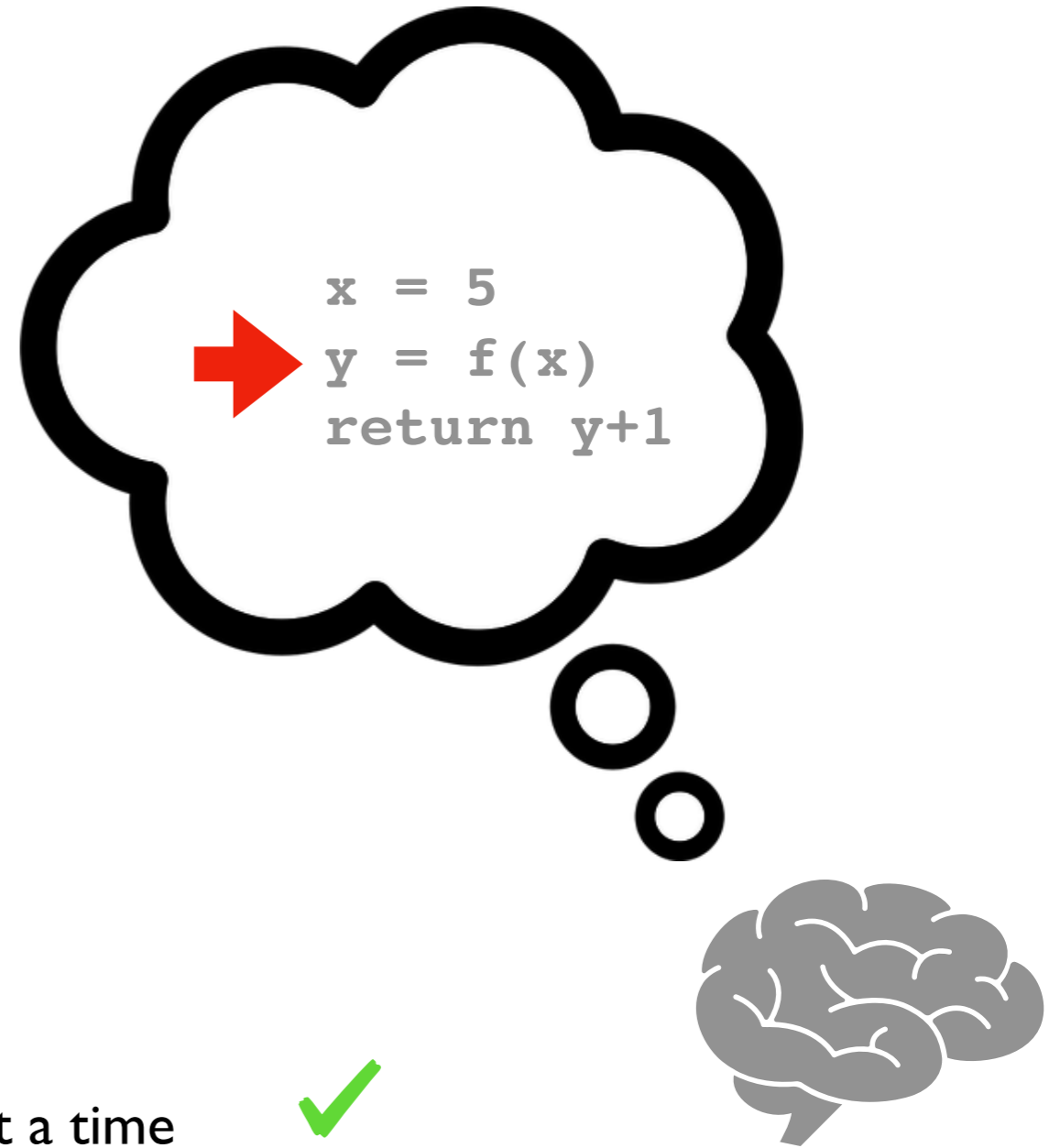
Meena Syamkumar
Andy Kuemmel

Exam 1 Conflict Form is available on the website

Mental Model of Control Flow

Code:

```
...
x = 5
y = f(x)
return y+1
...
```



```
x = 5
→ y = f(x)
  return y+1
```

1. do statements in order, one at a time ✓

three exceptions
2. **functions**: jump in and out of these ✓
3. **conditionals**: sometimes skip statements ← **TODAY**
4. **loops**: sometimes go back to previous

Learning Objectives Today

Reason about conditionals
- Conditional execution
- Alternate execution
- Chained conditionals
- Nested conditionals
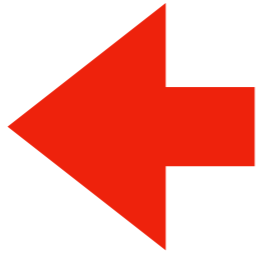
Understand code blocks
- Be able to identify the lines of code in the same block

Sanity checking
- Recognize errors
- Sanitize bad data automatically

Today's Outline

Review ⬅️

Control Flow Diagrams

Basic syntax for "if"

Identifying code blocks

Demos

Review 1: Indentation Example

```
print("A")
print("B")
```
not indented, so "outside" any function

```
def print_letters():
    print("C")
    print("D")
```
indented, so "inside" print_letters function

```
print("E")
print("F")
```
also not indented, so "outside" any function.
Runs BEFORE print_letters is called

```
print_letters()
```
printed last because print_letters is called last

what does it print?

A
B
E
F
C
D

Review 1: Indentation Example

```python
print("A")
print("B")
```
not indented, so
"outside" any function

```python
def print_letters():
    print("C")
    print("D")
```
indented, so "inside"
print_letters function

blank lines are **irrelevant**

```python
print("E")
print("F")
```
also not indented, so
"outside" any function.
Runs BEFORE
print_letters is called

```python
print_letters()
```

what does it print?

A
B
F
C
D

We use **indenting** to tell Python which code is **inside** or **outside**
of a function (or other things we'll learn about soon).

Review 1: Indentation Example

```
print("A")
print("B")

def print_letters():
    print("C")
    print("D")

print("E")
print("F")

print_letters()
```

we'll often call the lines of code **inside** something a **"block"** of code

A
B
E
F
C
D

Review 1: Indentation Example

```
print("A")
print("B")

def print_letters():
    print("C")



    print("D")
print("E")
print("F")

print_letters()
```

horizontal spaces
identify blocks
(not vertical space)

A

B

E

F

C

D

Review 2: Argument Passing

```
def h(x=1, y=2):
    print(x, y)  # what is printed?

def g(x, y):
    print(x, y)  # what is printed?
    h(y)

def f(x, y):
    print(x, y)  # what is printed?
    g(x=x, y=y+1)

x = 10
y = 20
f(y, x)
```
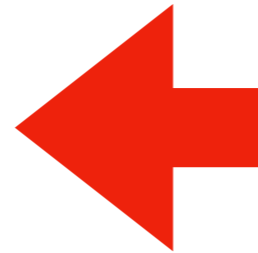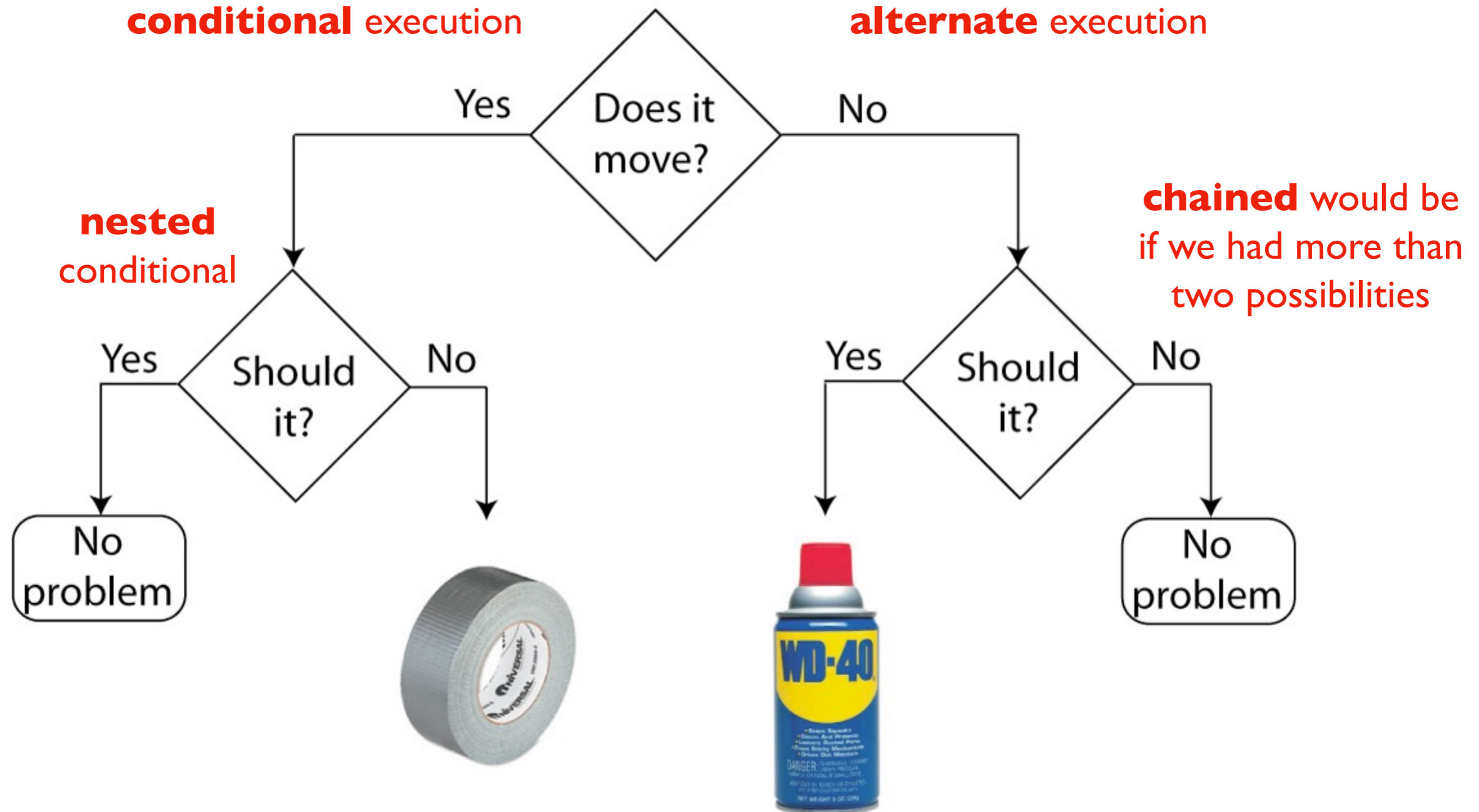
Today's Outline

Review

Control Flow Diagrams
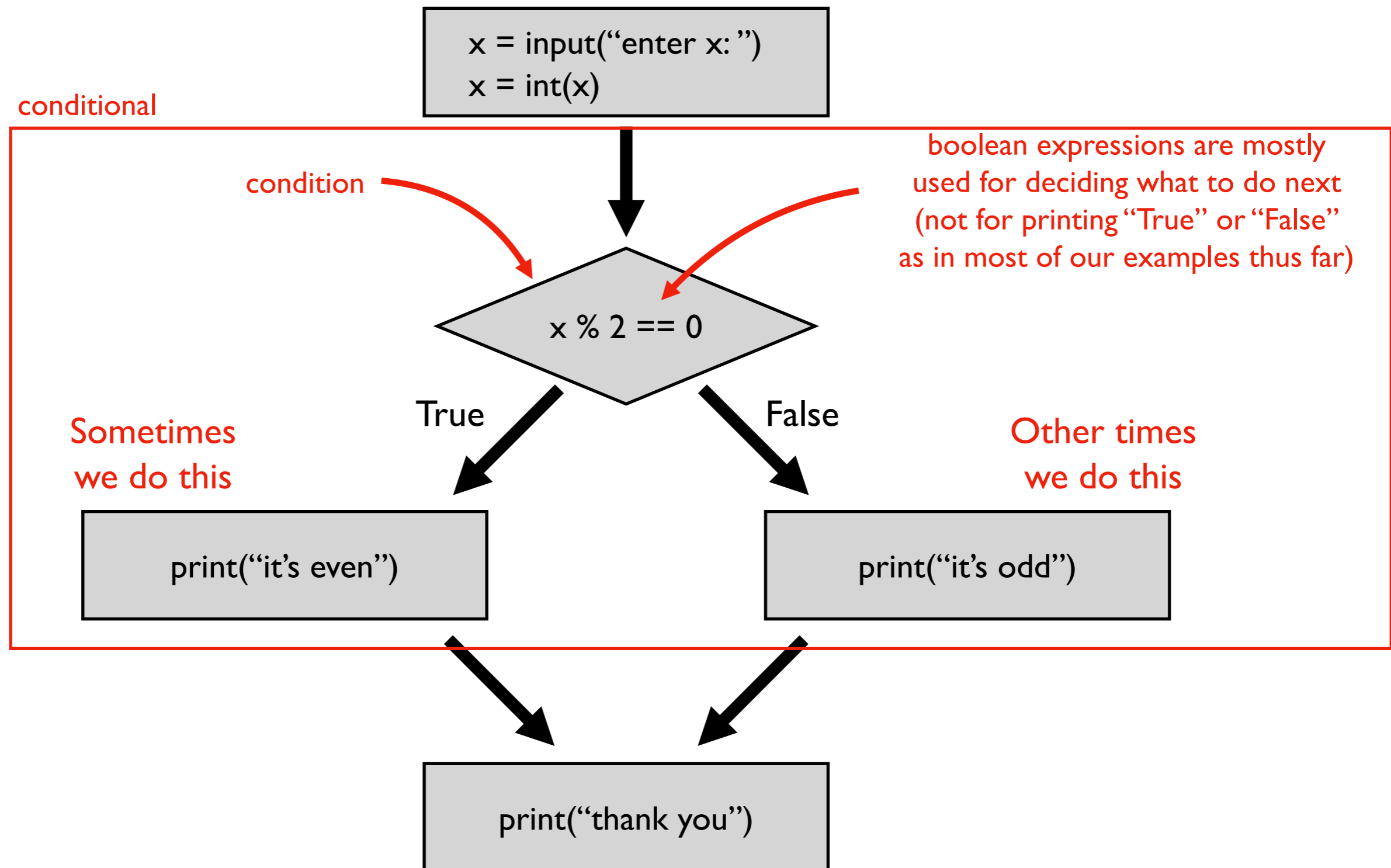
Basic syntax for "if"

Identifying code blocks

Demos

Laboratory Troubleshooting Flowchart

conditional execution

alternate execution

Yes — Does it move? — No

nested conditional

chained would be if we had more than two possibilities

Yes — Should it? — No

Yes — Should it? — No

No problem

No problem

in programming:
- questions are phrased as *boolean expressions*
- actions are *code/statements*

Control Flow Diagrams (Flowcharts for Code)

```
x = input("enter x: ")
x = int(x)
```

conditional

condition

boolean expressions are mostly used for deciding what to do next (not for printing "True" or "False" as in most of our examples thus far)

```
x % 2 == 0
```

True

False

Sometimes we do this

Other times we do this

```
print("it's even")
```

```
print("it's odd")
```

```
print("thank you")
```

Branches (aka "Paths of Execution")

Input/Output:

```
enter x: 8
it's even
thank you
```
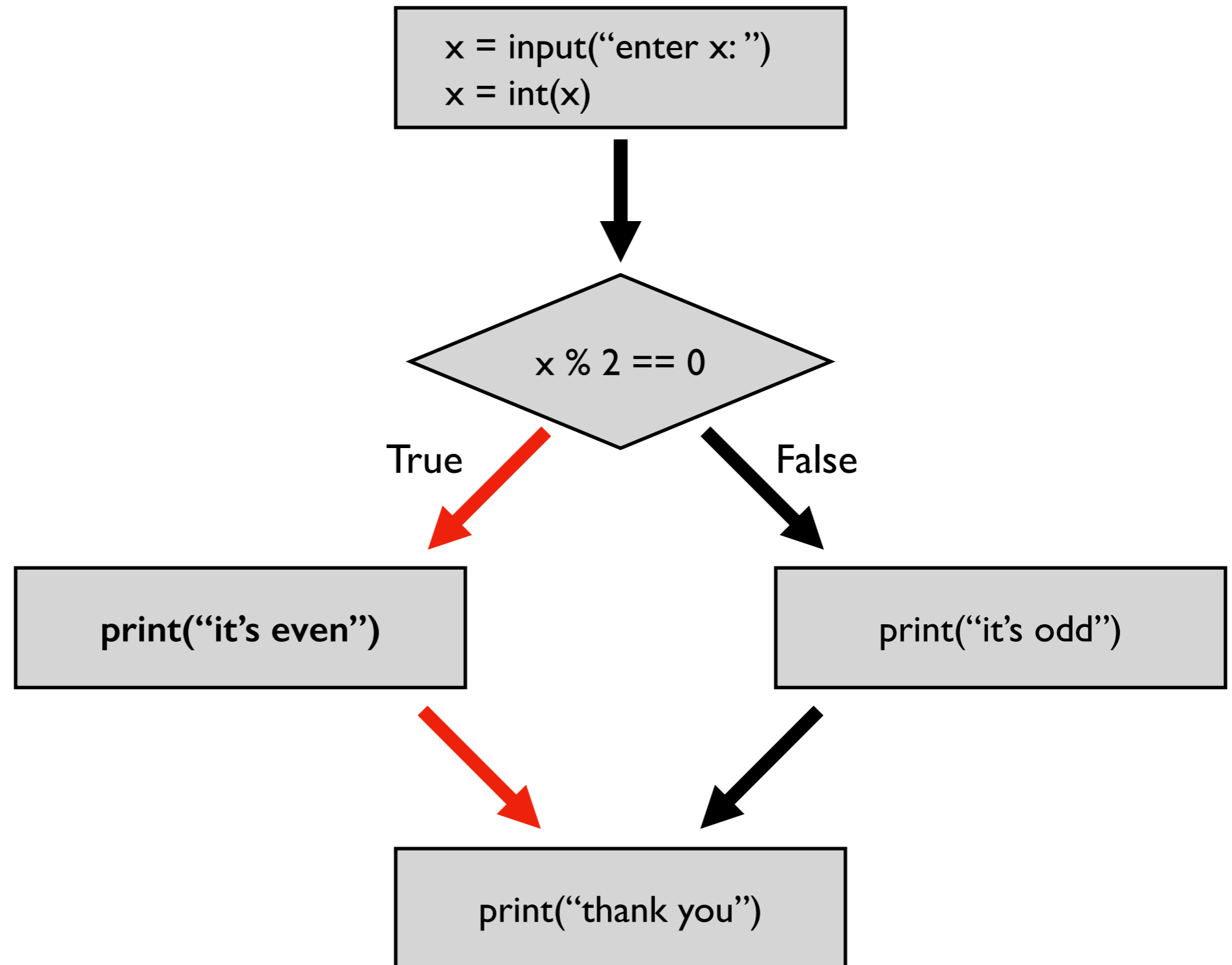
Branches (aka "Paths of Execution")

Input/Output:

```
enter x: 7
it's odd
thank you
```
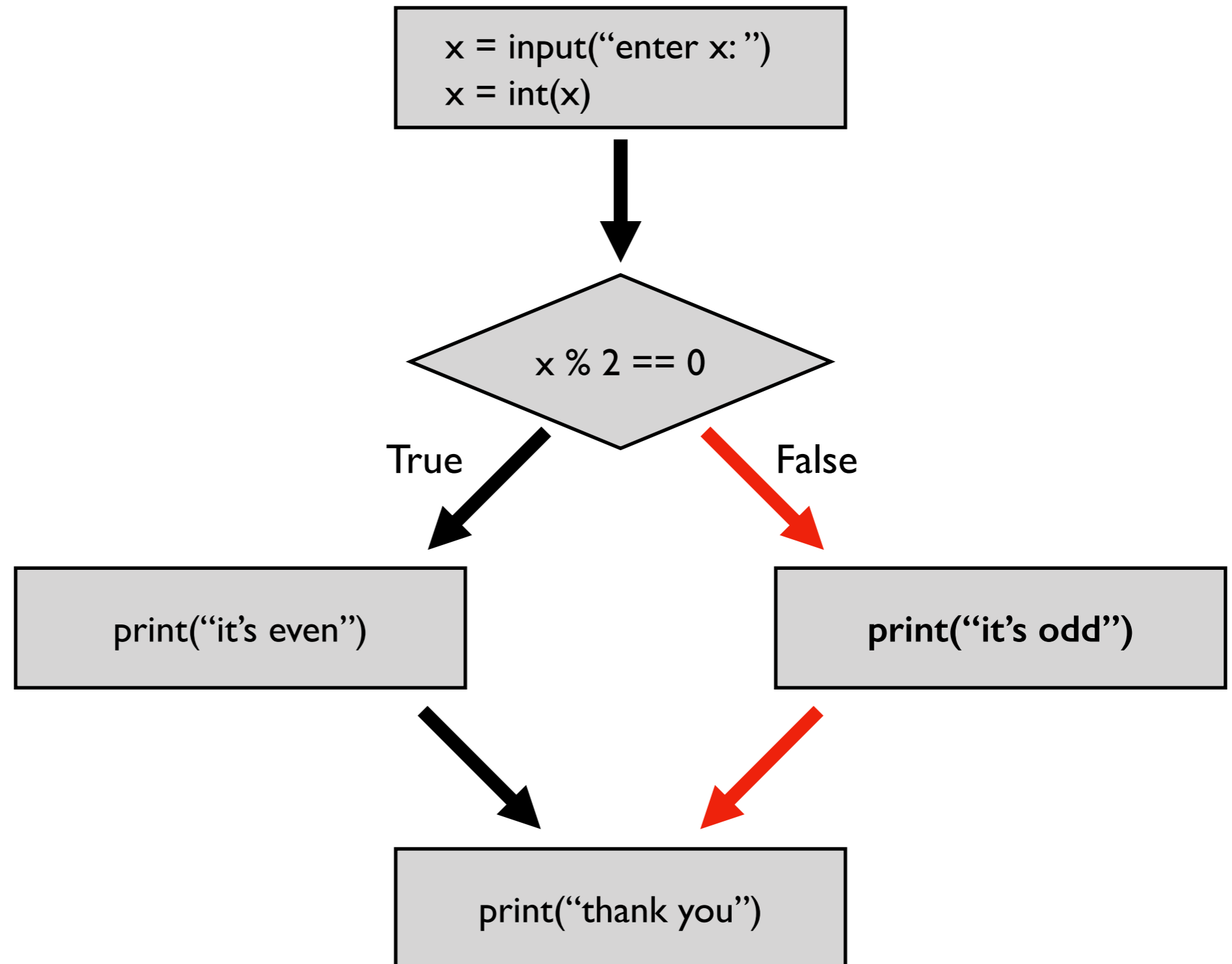

x = input("enter x: ")
x = int(x)

x % 2 == 0

True

False

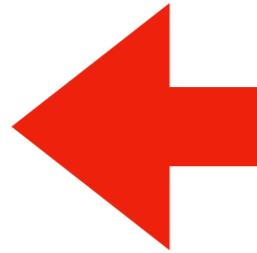print("it's even")

print("it's odd")

print("thank you")

Today's Outline

Review

Control Flow Diagrams

Basic syntax for "if"

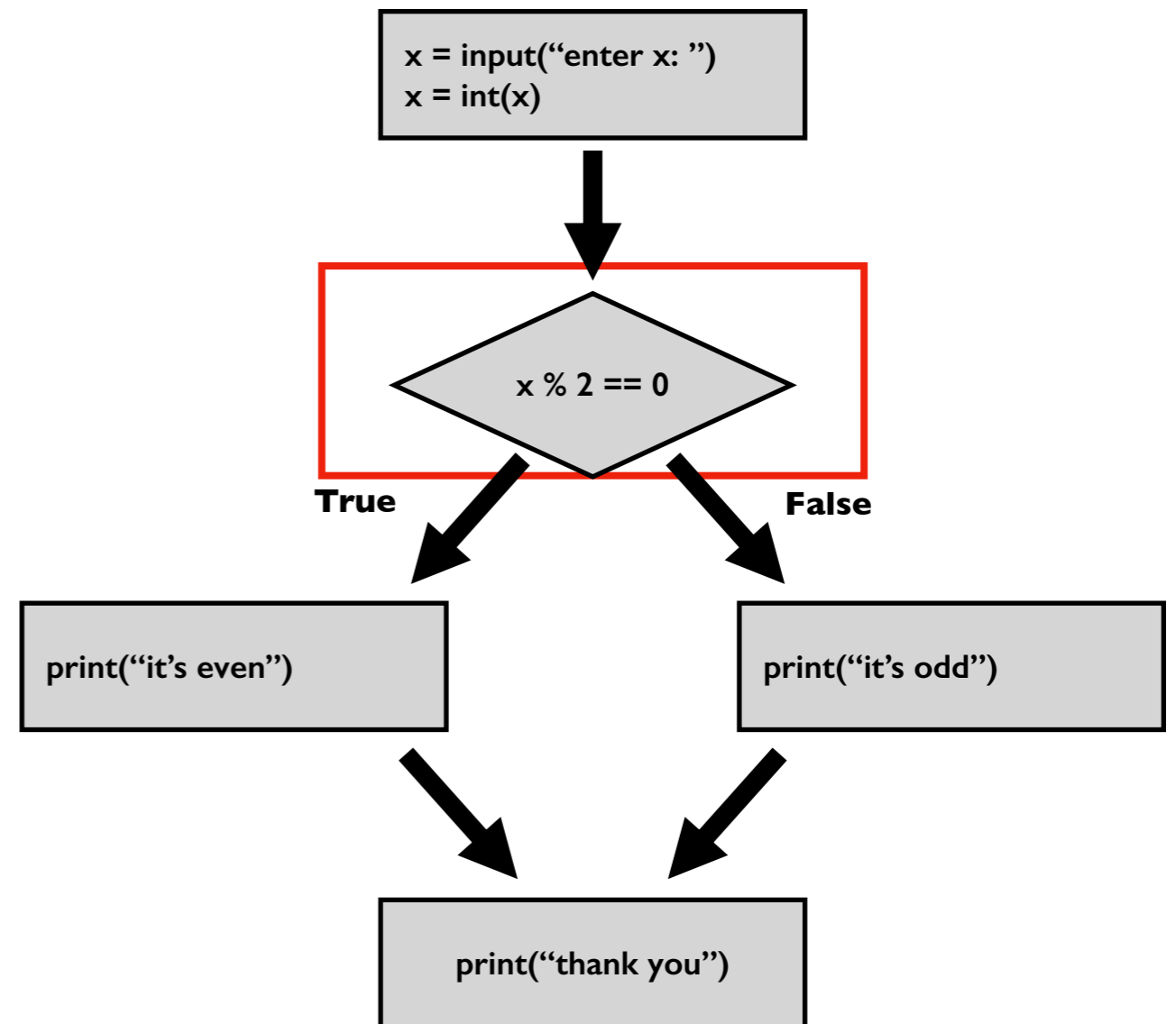Identifying code blocks

Demos

Writing conditions in Python

Code:

```
x = input("enter x: ")
x = int(x)

if x % 2 == 0:
```
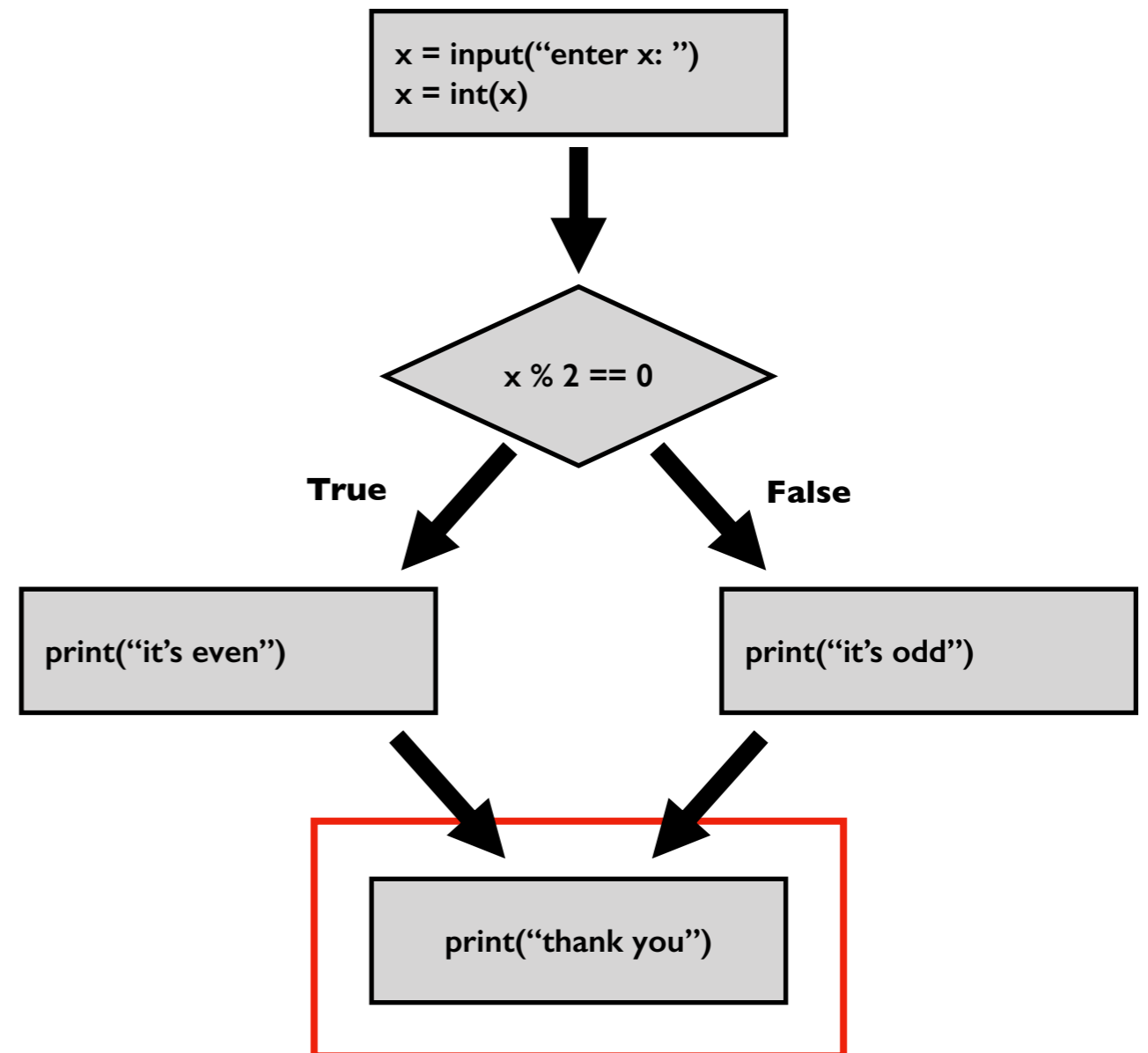
boolean expression

Writing conditions in Python

Code:

```python
x = input("enter x: ")
x = int(x)

if x % 2 == 0:
    print("it's even")
else:
    print("it's odd")

print("thank you")
```

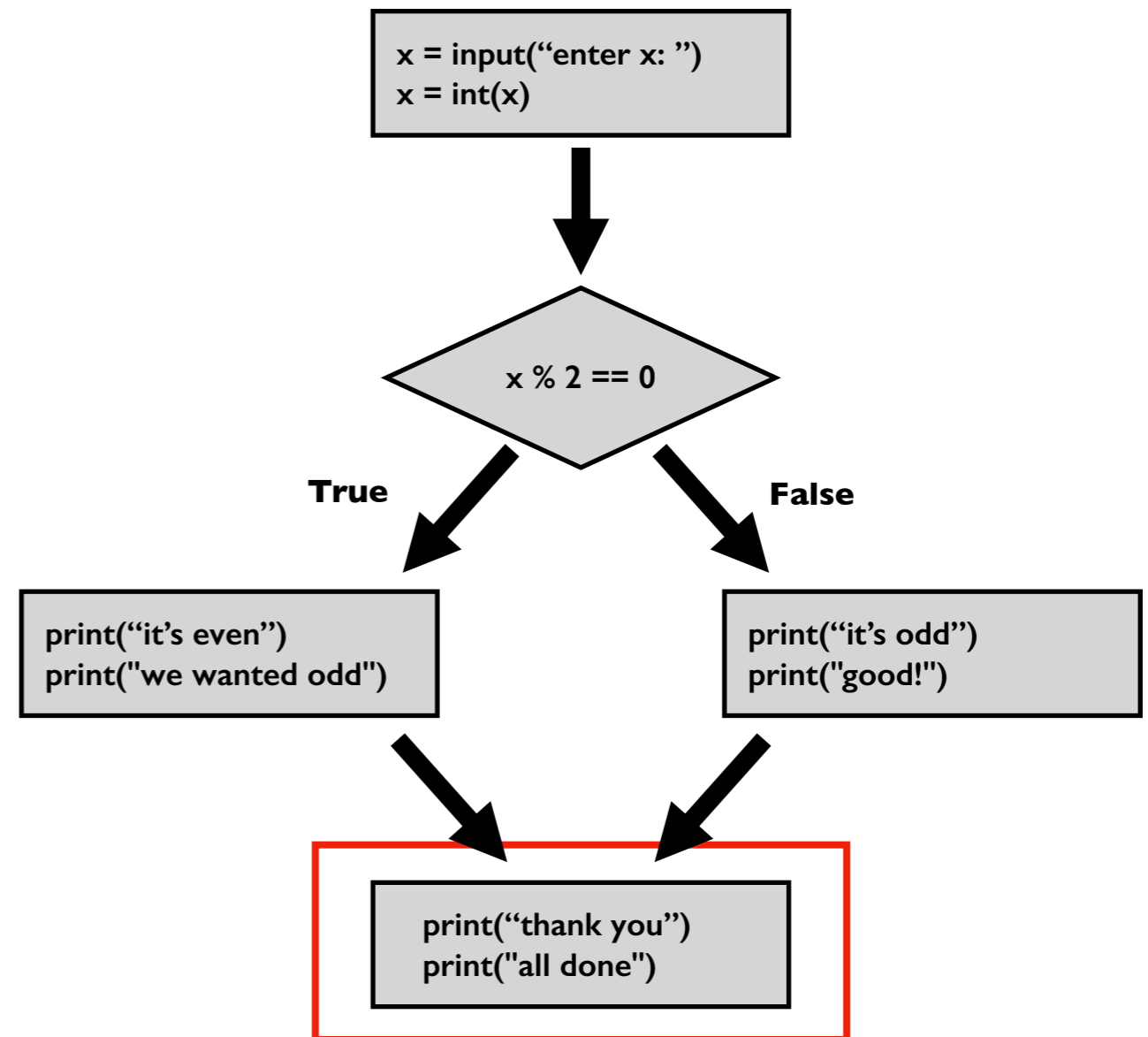colons will *almost* always be followed by a tabbed new line

Writing conditions in Python

Code:

```python
x = input("enter x: ")
x = int(x)

if x % 2 == 0:
    print("it's even")
    print("we wanted odd")
else:
    print("it's odd")
    print("good!")

print("thank you")
print("all done")
```
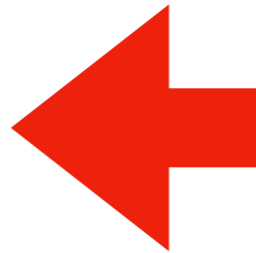
Today's Outline

Review

Control Flow Diagrams

Basic syntax for "if"

Identifying code blocks ⬅

Demos

Code Blocks

Code:

```
x = input("enter x: ")
x = int(x)

if x % 2 == 0:
    print("it's even")
    print("we wanted odd")
else:
    print("it's odd")
    print("good!")

print("thank you")
print("all done")
```

block of code inside "if"

block of code inside "else"

What if all this were inside a function?

Code Blocks

Code:

```python
def check_oddness():
    x = input("enter x: ")
    x = int(x)

    if x % 2 == 0:
        print("it's even")
        print("we wanted odd")
    else:
        print("it's odd")
        print("good!")

    print("thank you")
    print("all done")

check_oddness()
```
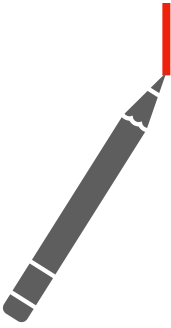
block of code inside "if"

block of code inside "else"

block of code in check_oddness

Identifying Code Blocks

Code:

```python
def check_oddness():
    x = input("enter x: ")
    x = int(x)

    if x % 2 == 0:
        print("it's even")
        print("we wanted odd")
    else:
        print("it's odd")
        print("good!")

    print("thank you")
    print("all done")

check_oddness()
```
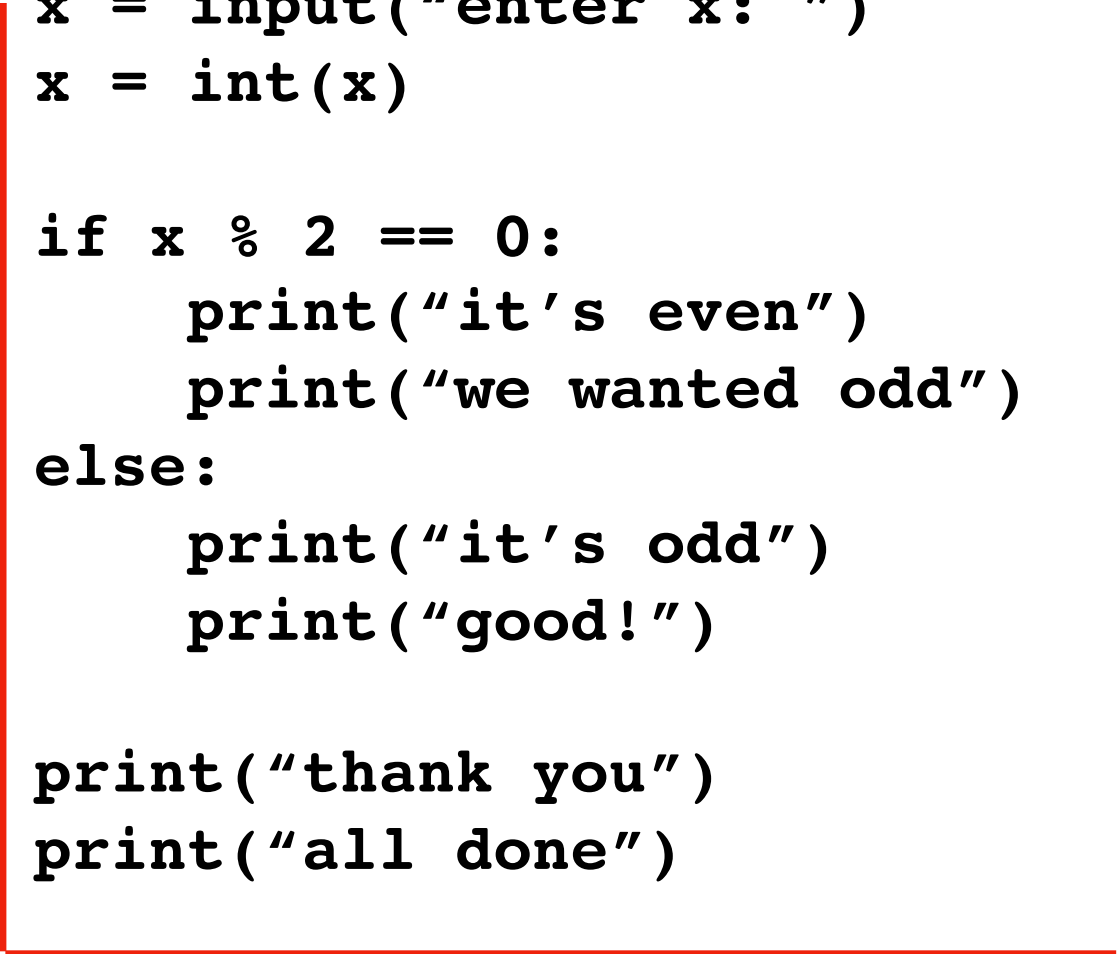
Step 1: look for a colon at end of a line

Step 2: start drawing a line on next code line, indented in

Identifying Code Blocks

Code:

```
def check_oddness():
    x = input("enter x: ")
    x = int(x)

    if x % 2 == 0:
        print("it's even")
        print("we wanted odd")
    else:
        print("it's odd")
        print("good!")

    print("thank you")
    print("all done")

check_oddness()
```

Step 3: continue down until you hit code that is less indented

Step 4: box off the code

Identifying Code Blocks

Code:

```
def check_oddness():
    x = input("enter x: ")
    x = int(x)

    if x % 2 == 0:
        print("it's even")
        print("we wanted odd")
    else:
        print("it's odd")
        print("good!")

    print("thank you")
    print("all done")

check_oddness()
```

Step 4: box off the code

**to find more boxes,
look for the next colon
and repeat**

Identifying Code Blocks

Code:

```
def check_oddness():
    x = input("enter x: ")
    x = int(x)

    if x % 2 == 0:
        print("it's even")
        print("we wanted odd")
    else:
        print("it's odd")
        print("good!")

    print("thank you")
    print("all done")

check_oddness()
```

to find more boxes,
look for the next colon
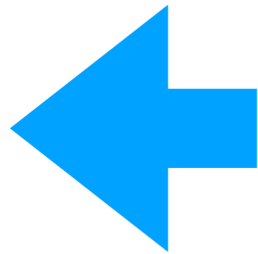and repeat

Today's Outline
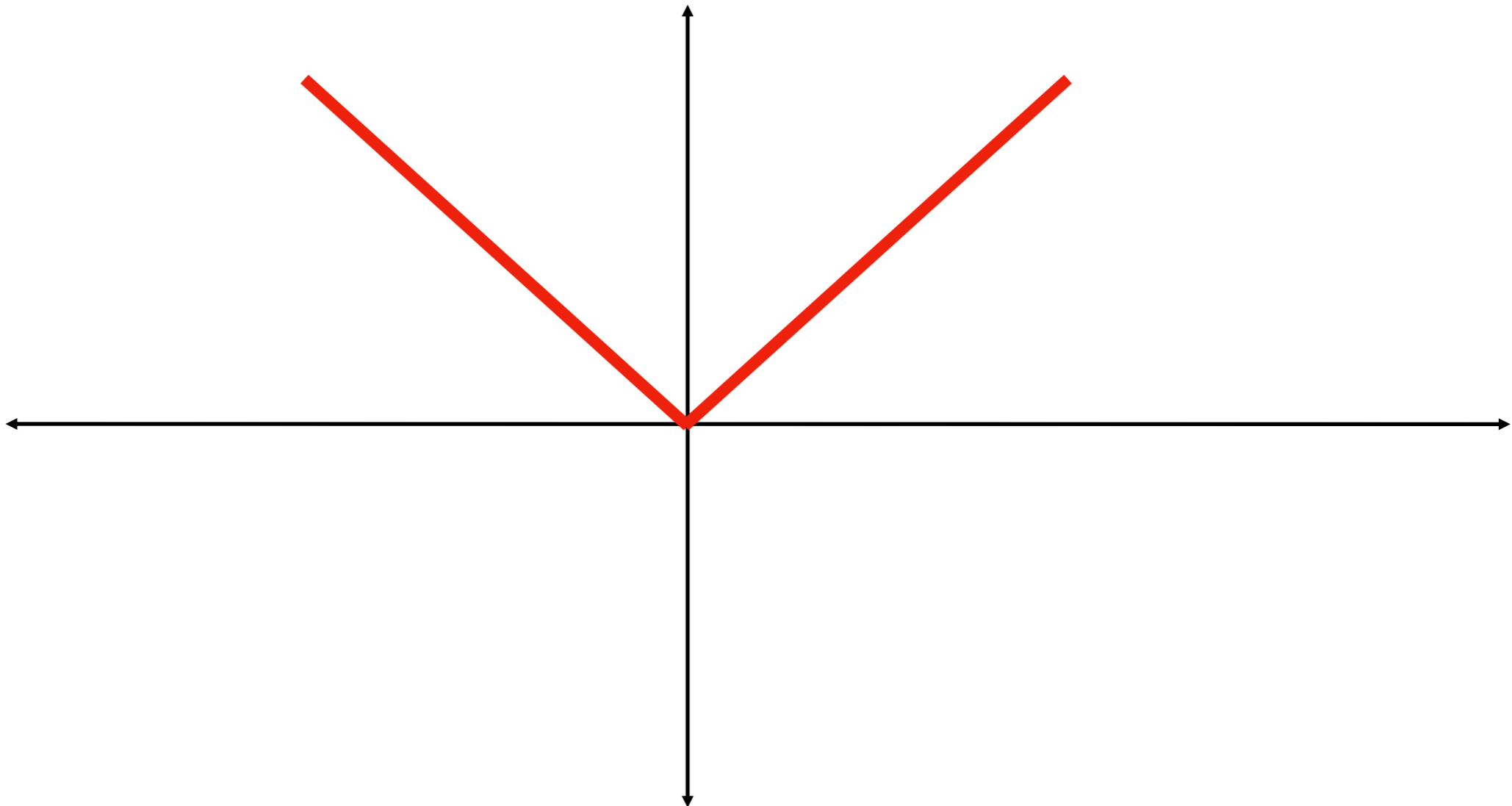
Review

Control Flow Diagrams

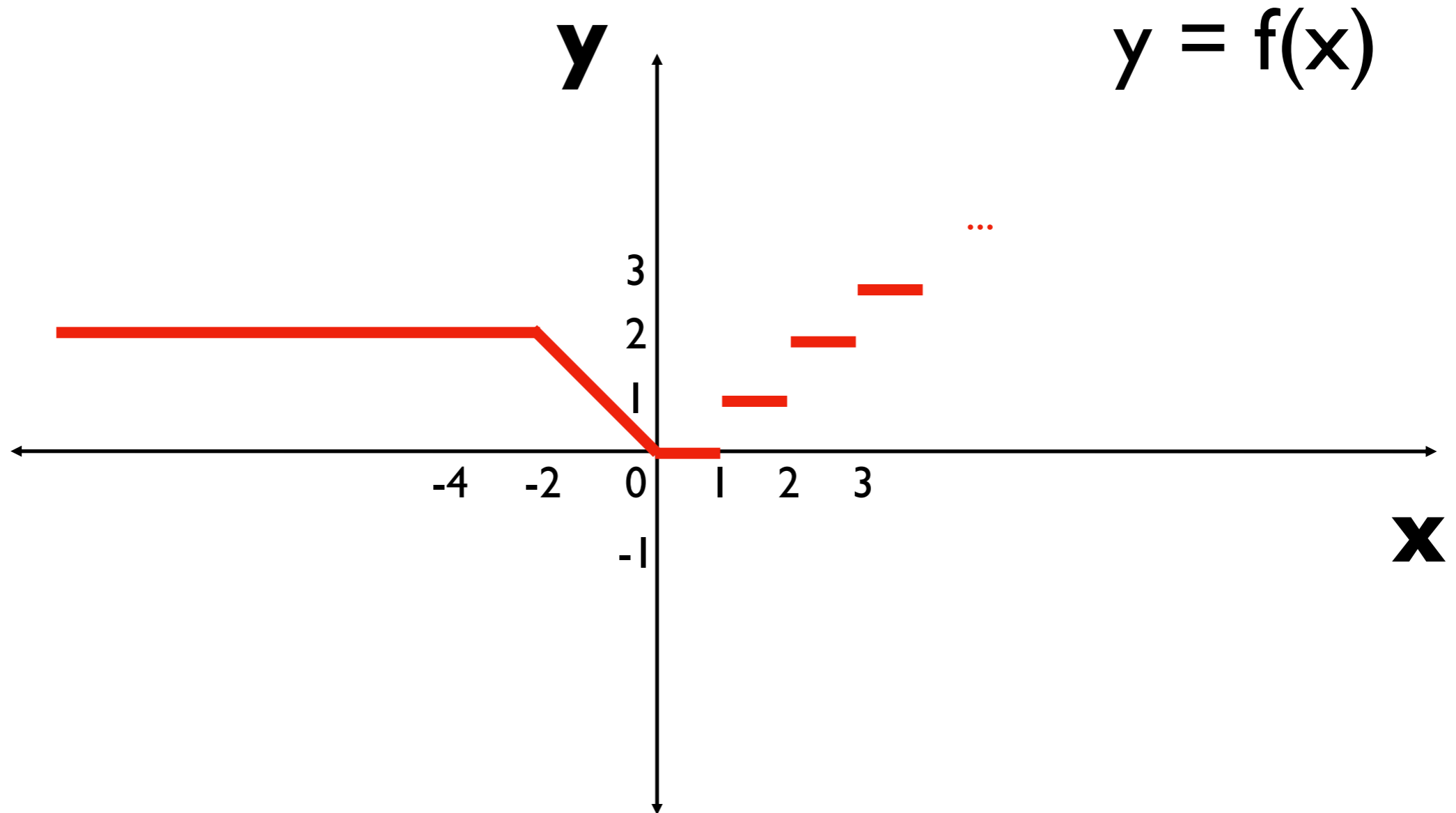Basic syntax for "if"

Identifying code blocks

Demos

Example: Absolute

compare 4 ways to compute the absolute of a number
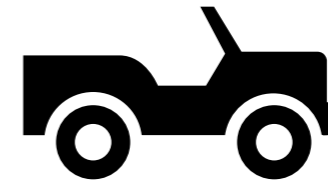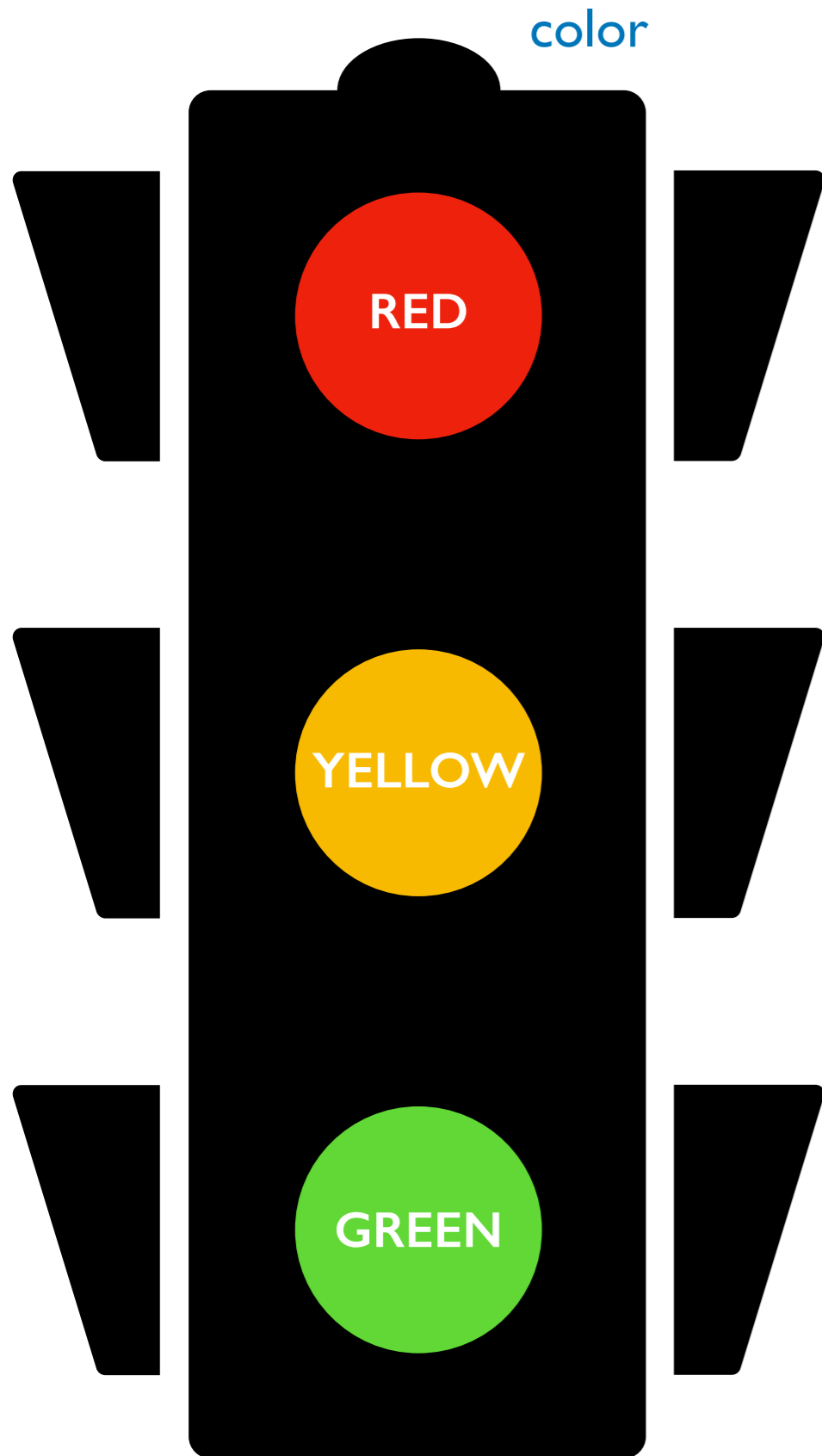**(step through in Interactive Exercises)**
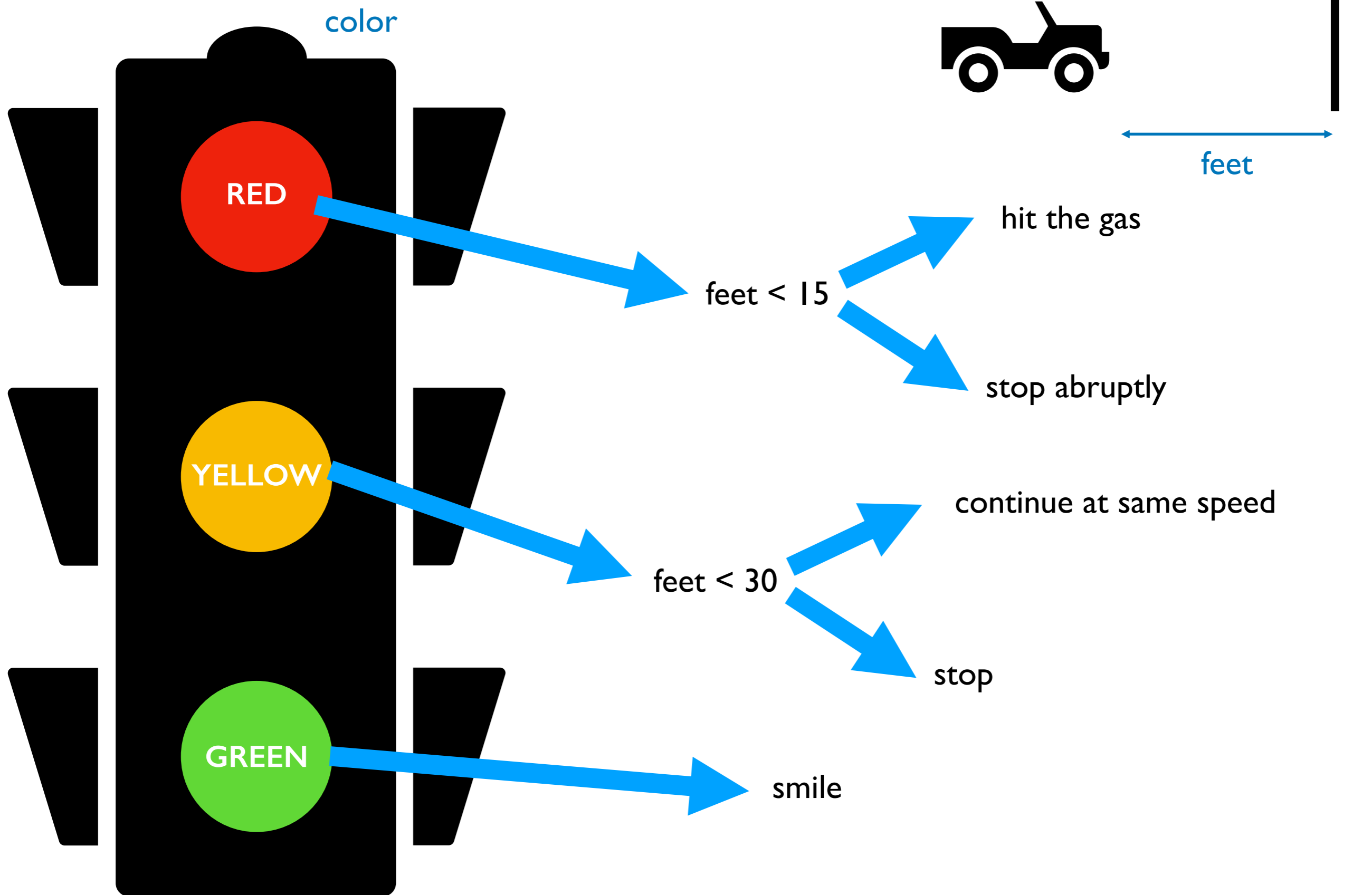
Example: Piecewise Function

Implement the f function in Python

Example: Stoplight – Next lecture …

color

RED

YELLOW

GREEN

feet

what should the driver do?

Example: Stoplight – Next lecture …

color

RED → feet < 15 → hit the gas
feet < 15 → stop abruptly

YELLOW → feet < 30 → continue at same speed
feet < 30 → stop

GREEN → smile

feet

Example: Date Printer – Next lecture …

```
please enter a year: (YYYY): 2021
please enter a month (1-12): 9
please enter a day (1-31): 29
the date is: Sep 29th of '21
```
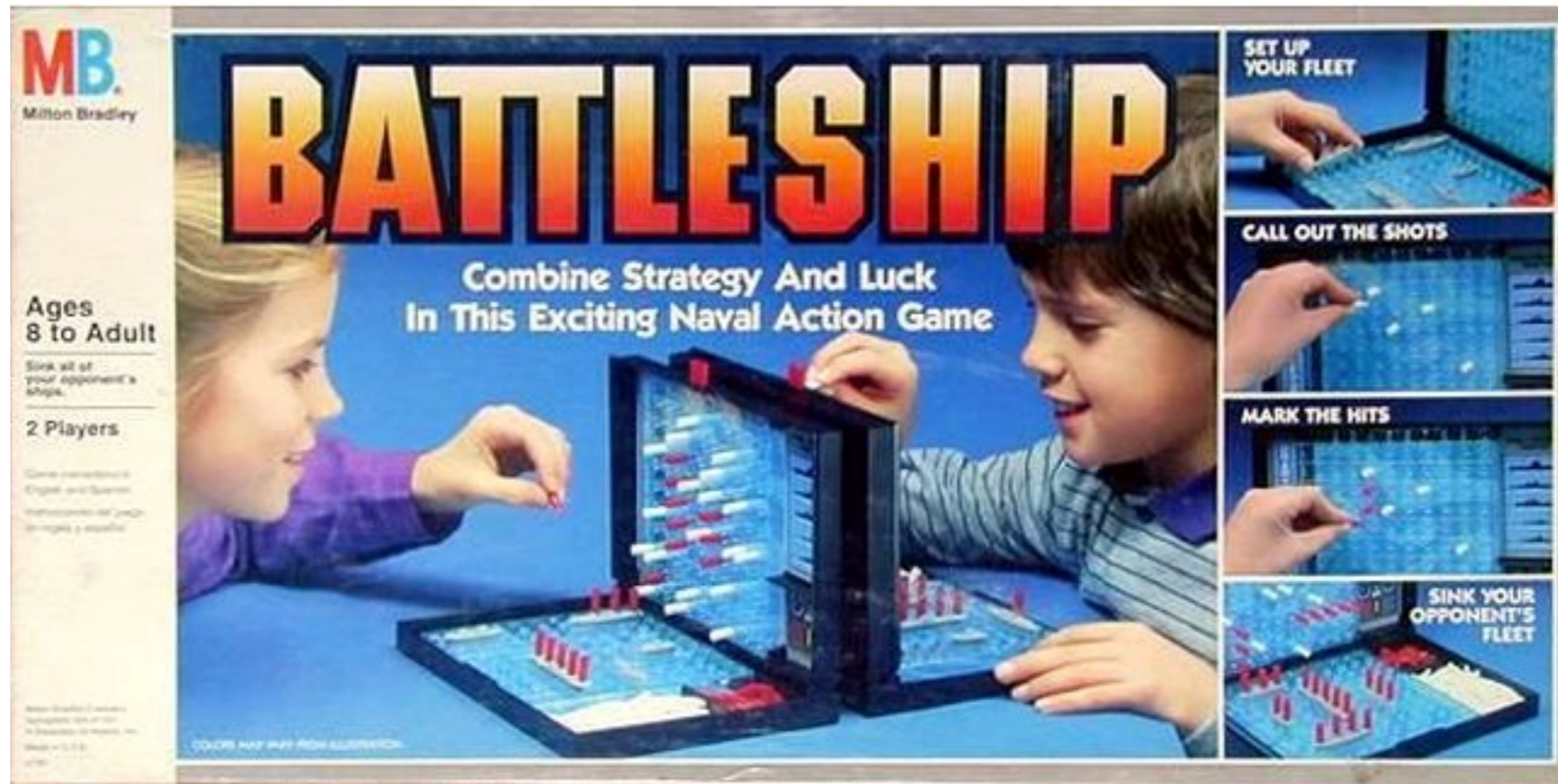
convert month num to name

'2-digit year

e.g., 1st, 2nd, 3rd, etc

Challenge: Better Battleship

Improvements
- give more meaningful feedback (not "True" or "False")
- check that user guessed in a reasonable range
- choose random placement for two ships, not overlapping
- show different symbols depending on which ship was hit
- give user up to 3 guesses (or until they get a hit)

Challenge: Addition Tester

1 point

```
what is 3+5? 5
what is 9+2? 10
what is 9+5? 2
```

0.5 points (close answer)

0 points

```
...

Your score is 6.5 of 10
```

We can get random number by using the random module:

```
random.randint(1, 10)
```