2024 Workshop on Gravitational Waves and High-Performance Computing

Geoffrey Lovelace

August 12, 2024 – August 16, 2024

Welcome to the workshop!

- Please make sure you create a **free** account at
  https://cocalc.com

- Workshop supported by the
  National Science Foundation

- Website with useful materials:
  https://geoffrey-lovelace.com/Workshop/2024

Photos

- We would like to take some photos during the workshop
- The photos would appear on the Cal State Fullerton website, in news stories about the workshop
- If you would prefer to not have your picture taken, please let me know (message me or speak with me during the break) that you'd like to opt out

A commonly held inaccurate model of teaching and learning

Joe Reddish, 2001, AAPT, San Diego

Bill Watterson - Calvin and Hobbs

Results from cognitive science and education research

- Learning requires mental effort.
- New information must link with what you already know.
- Most people learn best when interacting with others.

Daily schedule

- Morning: 9:30 AM - 11:00 AM
- Afternoon I: 12:30 PM - 2:00 PM
- Afternoon II: 2:30 PM - 4:00 PM

Tentative schedule

- **Monday**: Powers of 10 & computing, programming with Python
- **Tuesday**: Programming with Python, Unix Command Line, using a supercomputer
- **Wednesday**: Simulating colliding black holes, black holes, gravitational waves
- **Thursday**: Gravitational-wave research, panel discussion, data center tour (if possible)
- **Friday**: visualizing colliding black holes, exit survey

About the pace…

- The pace is intense: you'l be learning a lot
- It's normal to feel confused…that's actually what learning feels like
- There is no such thing as a dumb question!!
- You will get the most out of this experience by participating! It's more like learning a sport or a musical instrument or a language or …

GW PAC GRAVITATIONAL WAVE
Physics and Astronomy Center

Icebreaker

- If you had to gain one superpower, which one would you choose?

A Ability to fly

B Power to be invisible

Powers of 10

- How many meters across is Earth?

A 10^6

B 10^7

C 10^8

D 10^9

Meters: $10^0=1$ $10^1=10$ $10^2=100$ $10^3 = 1000$ 10^4 10^5 10^6 10^7 10^8 10^9 10^{10}

10^{11} 10^{12} 10^{13} 10^{14} 10^{15} 10^{16} 10^{17} 10^{18} 10^{19} 10^{20} 10^{21} 10^{22} 10^{23} 10^{24} 10^{25} 10^{26}

Powers of 10

- How many meters across is Earth?

A	10^6
B	10^7
C	10^8
D	10^9

Powers of 10

- How many meters is a light year?

A	10^8
B	10^{12}
C	10^{16}
D	10^{20}

Powers of 10
& computers

Humans

- First entities called “computers” were teams of people
- Divide up the work into operations done in parallel, by hand (perhaps with mechanical aid)
- Redundant calculations to check accuracy
- Since 1700s
- 10^{-1} to 1 FLOPS / human (decimal operations / second / human)

1949 NACA High Speed Flight Station “Computer Room”)

Image courtesy wikipedia

Colossus (1942)

- First programmable, digital, electronic computer
- Break codes in World War II Britain
- 5 x 10^5 FLOPS

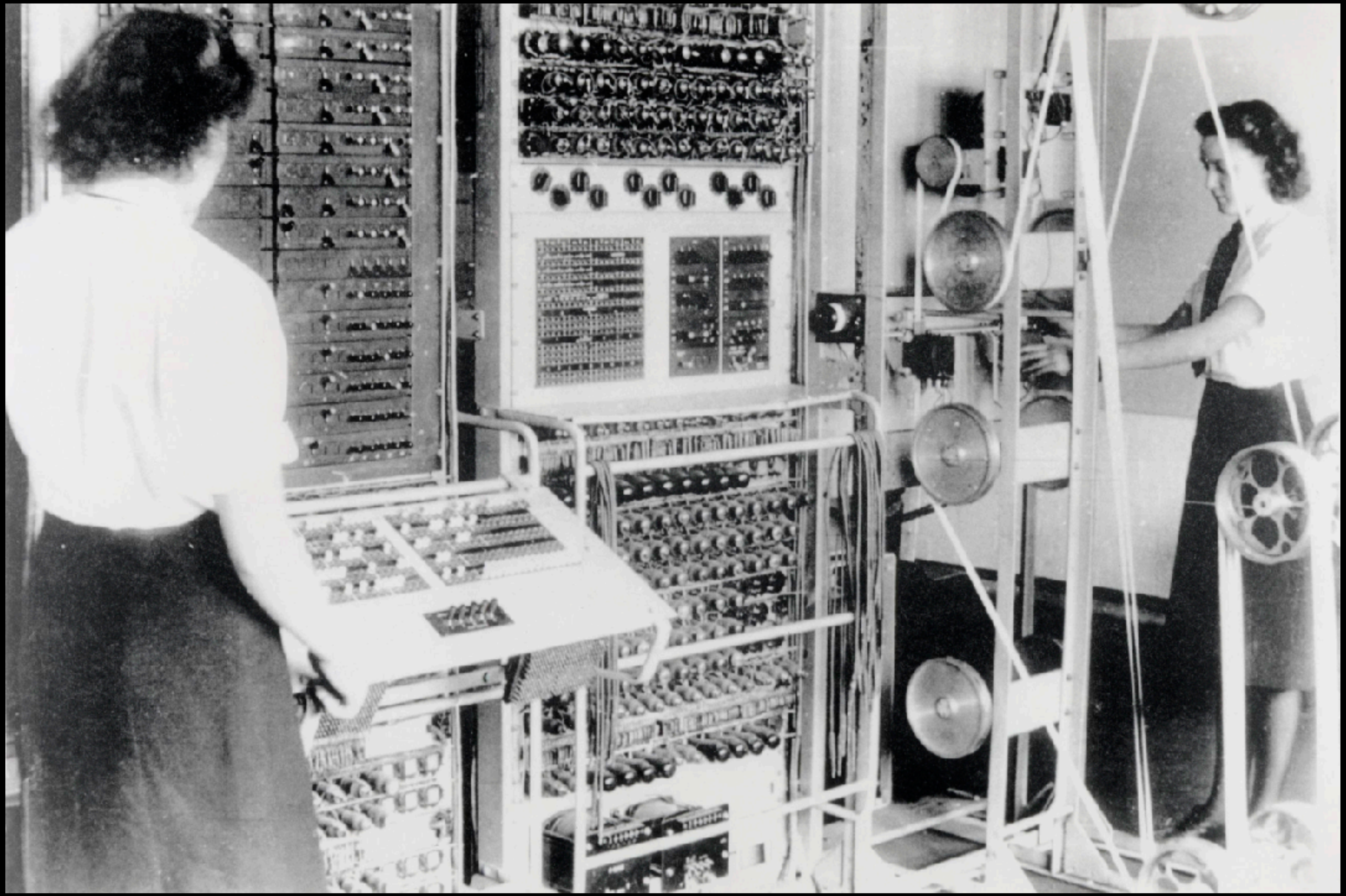


Image courtesy wikipedia

My first Mac (1984)

- First Macintosh
- 1×10^6 FLOPS

Image courtesy wikipedia

My Mac in 2003

- 2 cores
- 1-2 x 10^9 FLOPS

Image courtesy Apple

My current Mac

- Apple M1 Max (10 cores)
- 3 x 10^{11} FLOPS

Image courtesy Apple

My iPhone

- 6 cores
- 1×10^{11} FLOPS

Image courtesy Apple

Clicker question #1.4

- In 1 second, today's high-end smart phones can perform ________ calculations per second (FLOPS).

A 10^6 (1 million)

B 10^9 (1 billion)

C 10^{11} (100 billion)

D 10^{13} (10 trillion)

Images courtesy wikipedia, NASA

Clicker question #1.5

- In 1 second, today's high-end smart phones can perform as many calculations as _______ humans?

A 10^6 (1 million)

B 10^9 (1 billion)

C 10^{11} (100 billion)

D 10^{13} (10 trillion)

For comparison:

Humans alive in 2018: 7.6×10^9
Total humans who ever lived: 10^{11}
Sources: google.com, pro.org

Clicker question #1.3

- Today's most powerful computers are ________ times more powerful than *today's high-end personal computers*.

A	10 (ten)
B	10^4 (10 thousand)
C	10^7 (10 million)
D	10^{10} (10 billion)

Images courtesy wikipedia, NASA

Ocean supercomputer at Cal State Fullerton

- Supercomputer for Cal State Fullerton Gravitational-Wave Physics and Astronomy Center
- 828 cores
- $\approx 2 \times 10^{12}$ FLOPS

Frontera

- Most powerful computer
  I have accessed
- 470k cores
- 2.35×10^{16} FLOPS

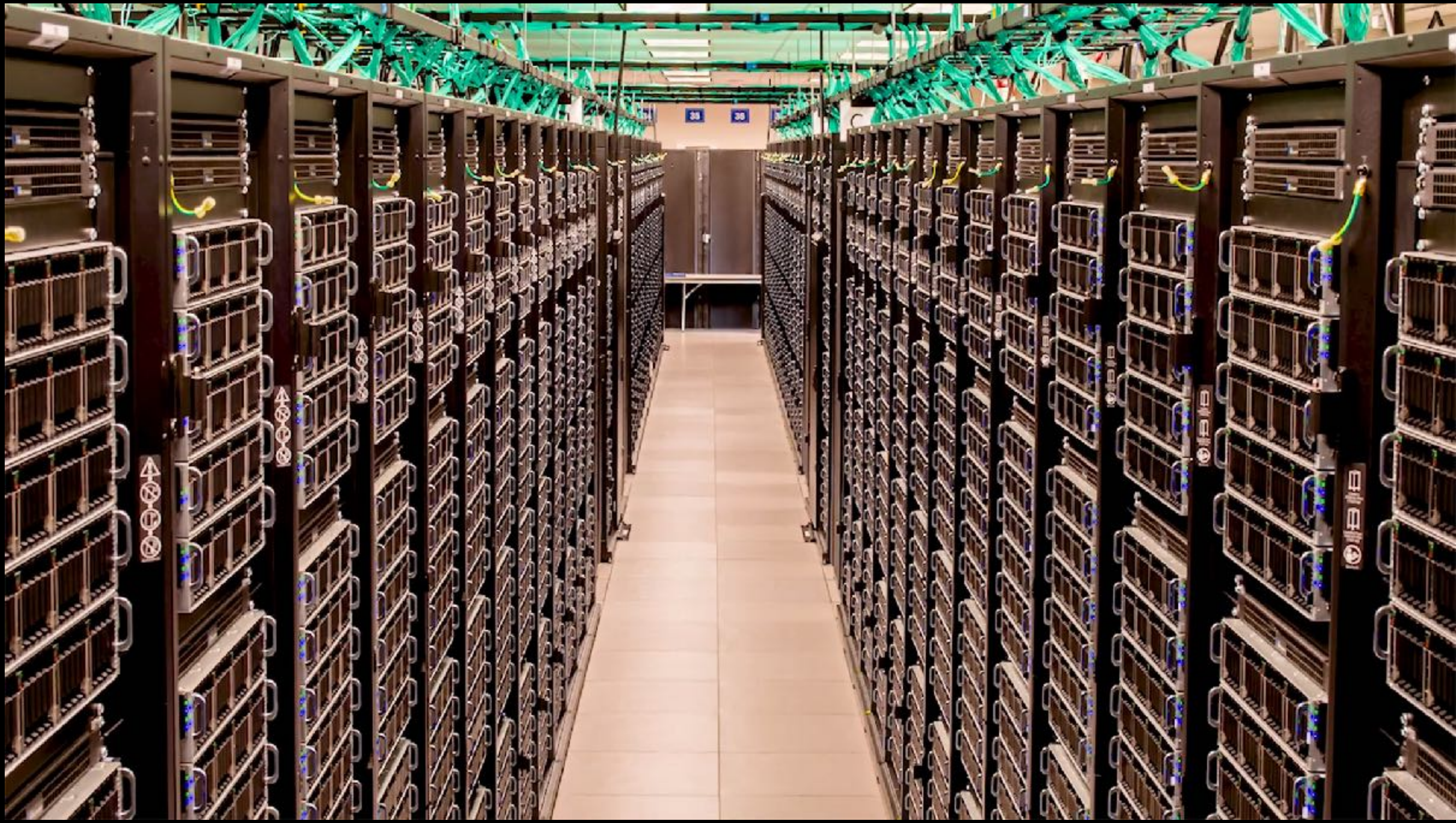


Image courtesy Frontera, Texas Advanced Computing Center

Frontier

- Most powerful computer in the world as of June 2022
- 8.7 million cores
- 1.7×10^{18} FLOPS

Image courtesy Frontier, Oak Ridge National Lab

High performance computing

- Computing beyond what personal devices can do
  - Many cores work together in parallel

FLOPS	Example	Computing Type
10^0	*Addition by human with pen & paper*	*Early*
10^5	*Room-sized computer in 1940s*	
10^6	Personal computers around year 1984	Personal
10^9	Personal computers around year 2000	
10^{11}	High-end PC/smartphone today	
10^{12}	**Small supercomputer today**	**High-Performance**
10^{16}	**Most powerful computer I ever used**	
10^{18}	**Most powerful computer in the world**	

Clicker question #1.3

- Today's most powerful computers are __________ times more powerful than *today's high-end personal computers*.

A. 10 (ten)

B. 10^4 (10 thousand)

C. 10^7 (10 million)

D. 10^{10} (10 billion)

Images courtesy wikipedia, NASA

Clicker question #1.6

- In 1 second, the most powerful computer in the world can perform as many calculations as _______ humans?

A 10^8 (100 million)

B 10^{11} (100 billion)

C 10^{14} (100 trillion)

D 10^{18} (1 quintillion)

For comparison:

Humans alive in 2024: 8.2 x 10^9
Total humans who ever lived: 10^{11}
Sources: google.com, pro.org

Clicker question #1.7

- In 1 second, a small supercomputer like Ocean can perform as many calculations as _______ humans?

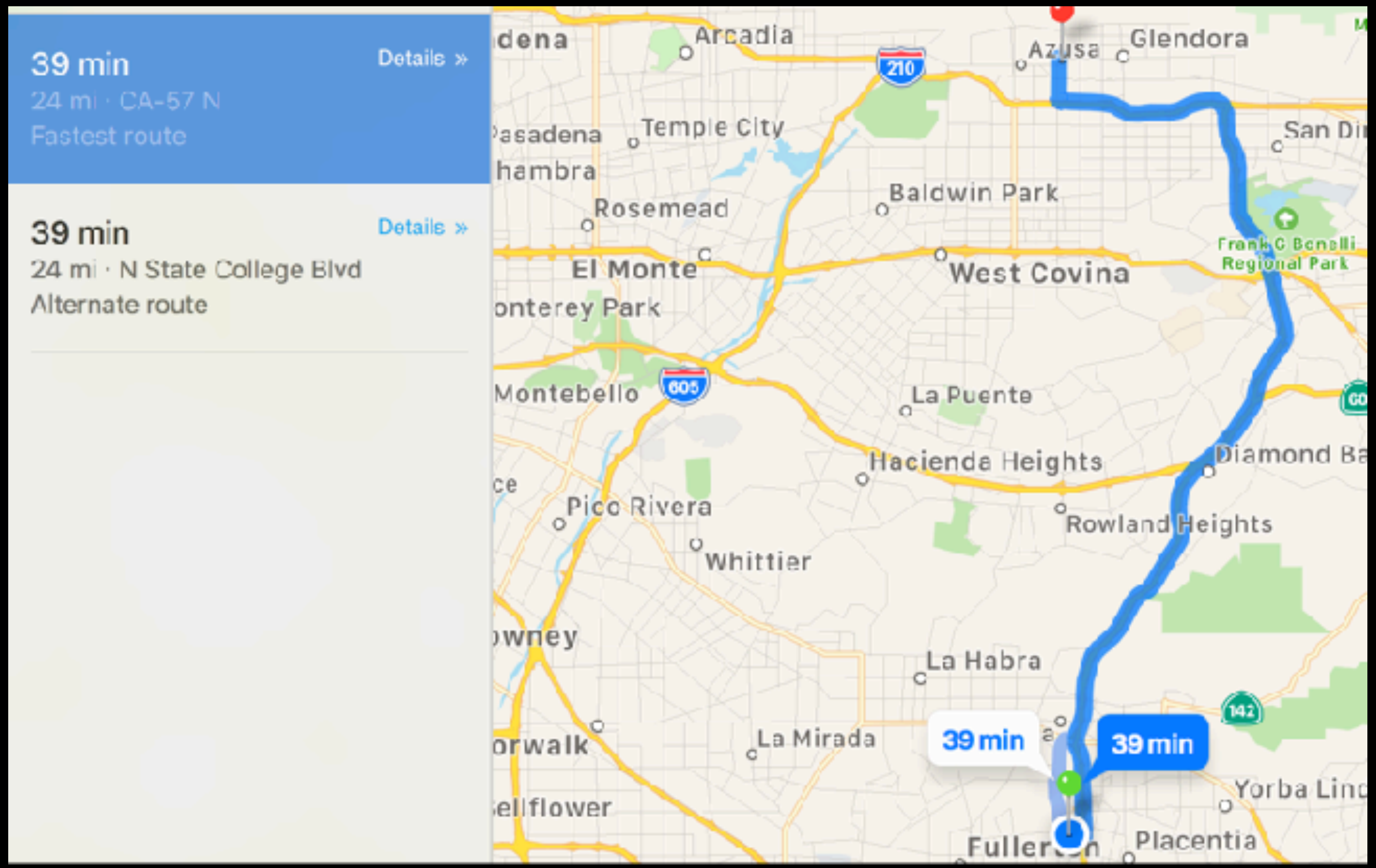
A	10^6 (1 million)
B	10^9 (1 billion)
C	10^{12} (1 trillion)
D	10^{15} (1 quadrilliion)

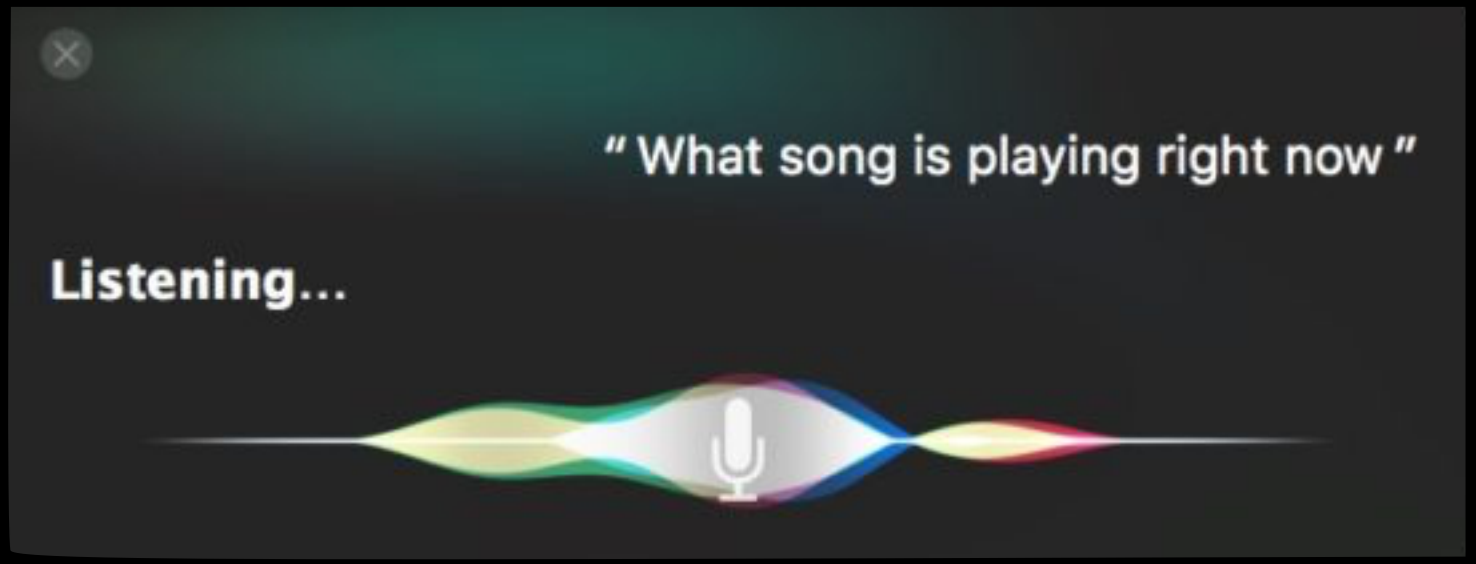
For comparison:

Humans alive in 2018: 7.6 x 10^9
Total humans who ever lived: 10^{11}
Sources: google.com, pro.org

High-performance computing in everyday life

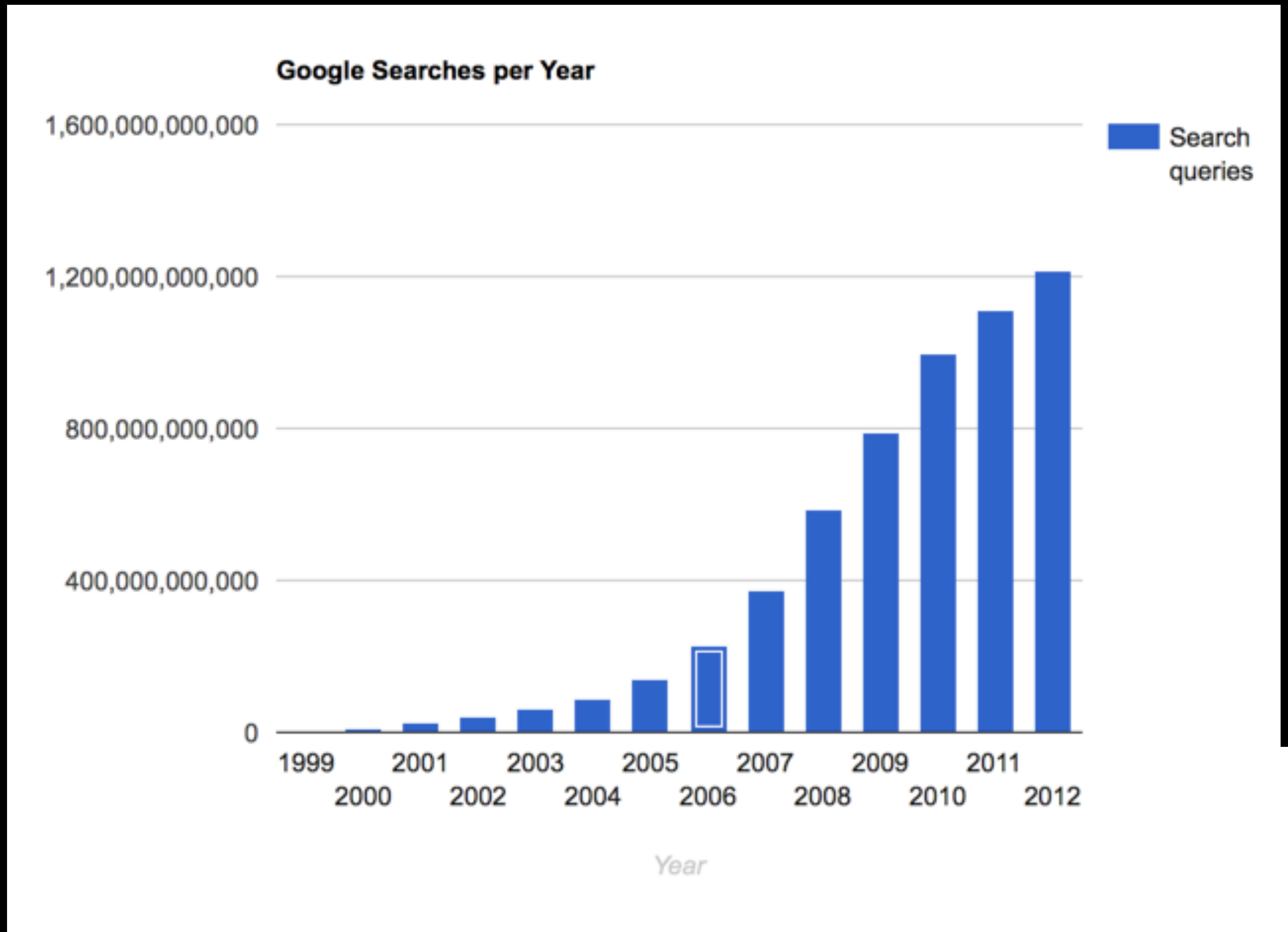
- Cloud computing
  - Search the web
  - Identify a song
  - Get directions
  - Voice assistants
  - Speech recognition
  - Cloud AI servers (ChatGPT & friends)





Example: Google search

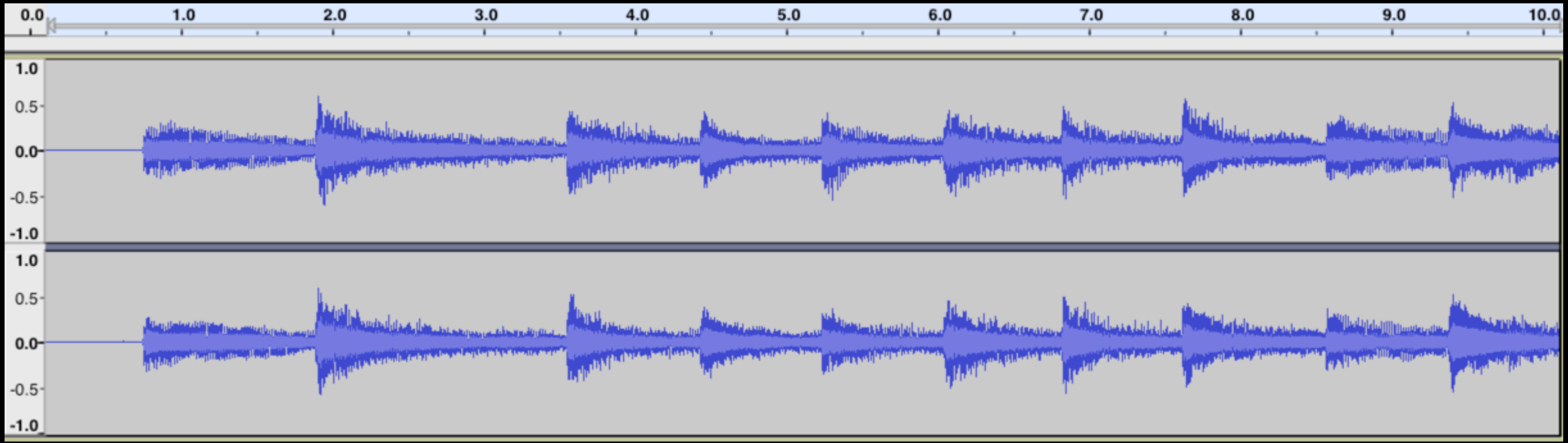
- Search ~ 10^{13} web pages
- 10^3 "servers" per query
- Each query takes about 0.2 seconds
- 1x 10^5 queries on average every second of every day
  - All google servers: ~10^{17} flops in 2008

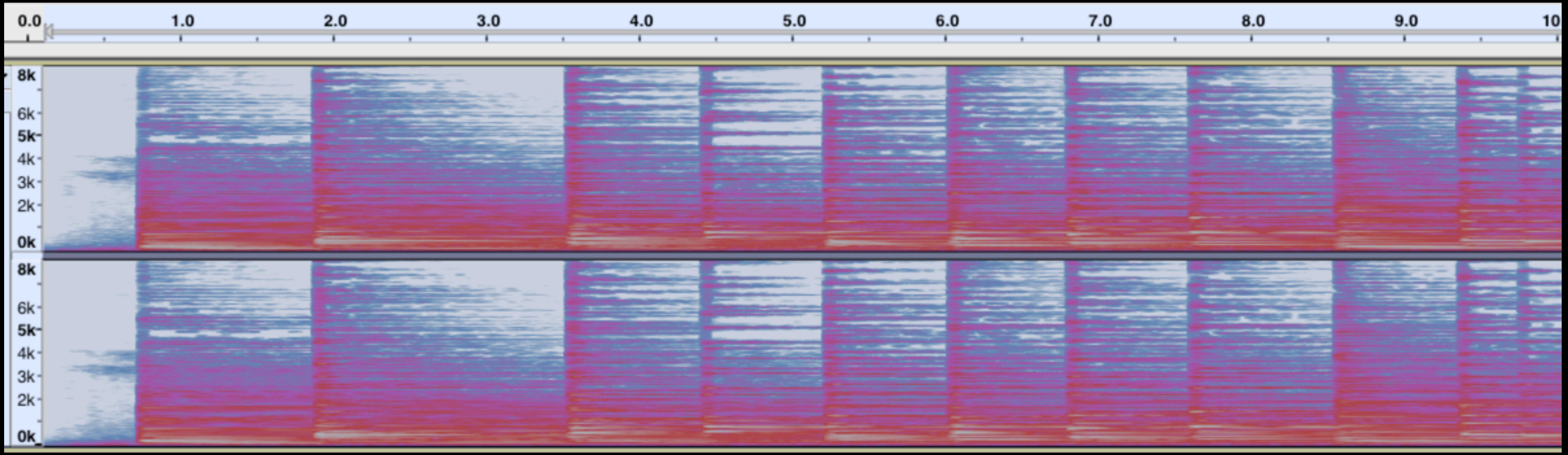


Images courtesy Google, internetlivestats.com

Example: Shazam

- 200 queries on average every second of every day
  - Convert sound into time-frequency plots, filter to keep only the loudest notes
  - Compare to a large library
  - Similar to how LIFO searches data for gravitational waves!
  - One query is a PC-sized calculation, roughly





Amazon web services data center
Courtesy amazon.com

Microsoft Azure data center
(courtesy sensorslab.co)

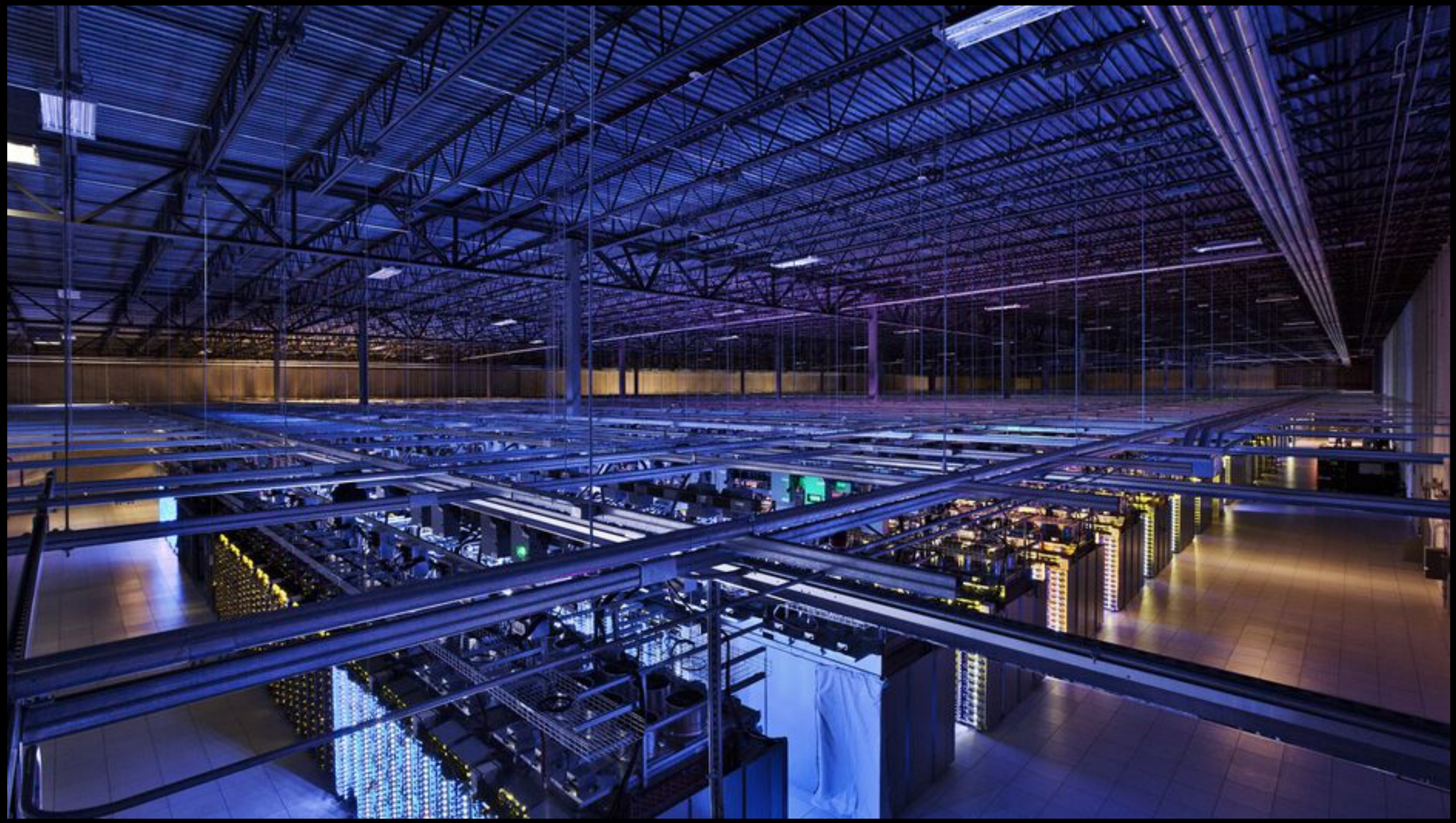


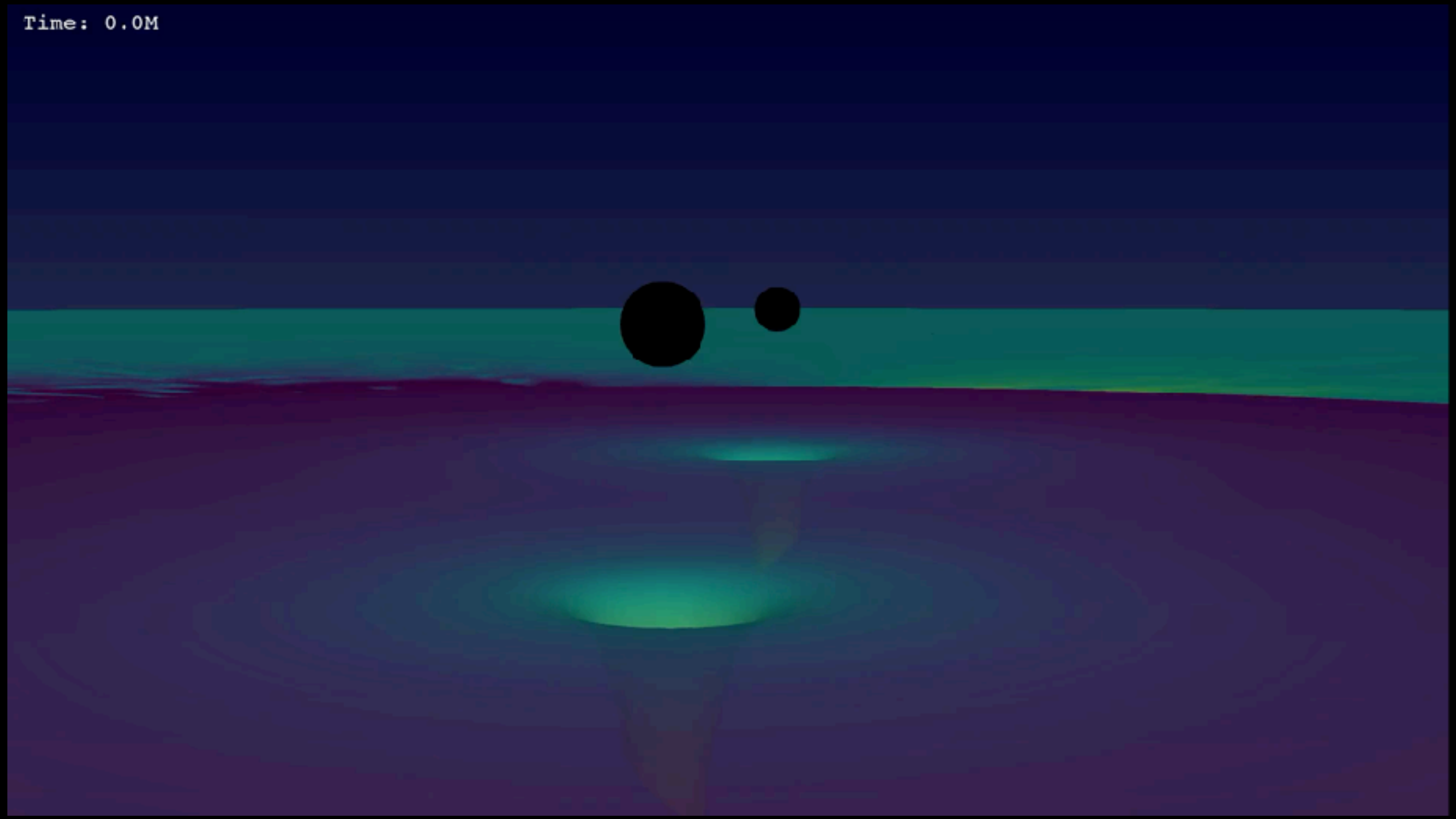
Image courtesy cnet: Google data center, Council Bluffs, Iowa

Google: 60,000 searches/second

Provide many 10^{15} FLOPS
of performance to customers

High-performance computing for science

- Solve otherwise unsolvable problems
- Insight into scientific data & results
  - Experimental measurements
  - Results of calculations
  - Complicated pencil & paper results



Movie & calculation by undergraduate
Alex Carpenter
Simulating eXtreme Spacetimes collaboration

Programming with Python

Programming is like magic

- Say the right cryptic words and something cool happens
- Mess up a word and the spell fizzles

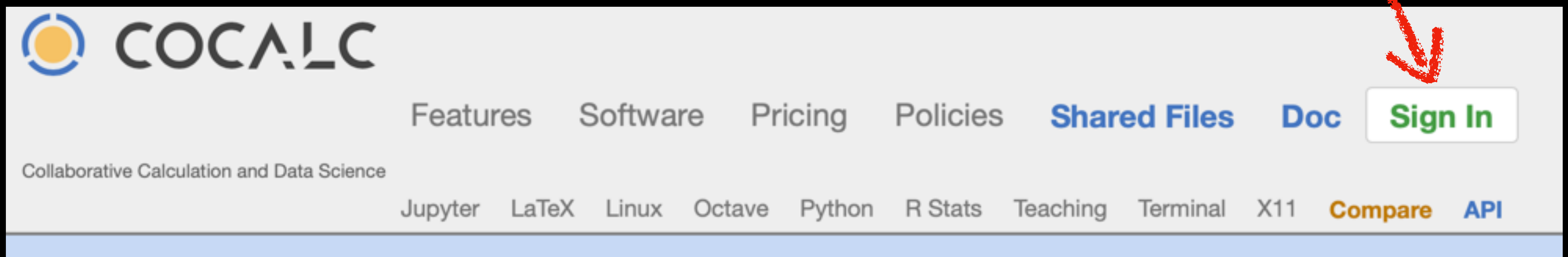
Images courtesy Warner Bros.

Cocalc

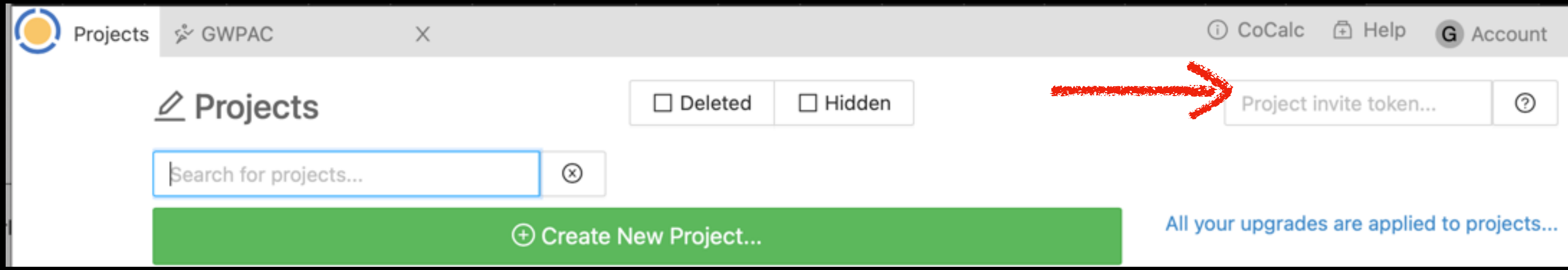
- https://cocalc.com
- Limited paid service
    - This course: ~$20/month paid plan (I paid, don't worry!)

Image courtesy amazon, Tech Vision

- Open https://cocalc.com and sign in



- See slack chat for the "token": enter it in the "token" box and press enter



https://cocalc.com/app?project-invite=a8VwaQcqDPAMekcc

- Click "Day1.ipynb"

- Scroll to your name, and click in the box saying "`# Insert code here`" labeled below your name

  - Enter this code: VBBNN1tpwckqE5mw

Output

- Your program needs to tell you the result
- Tradition since 1974: first program prints "Hello world"
- Python (language commonly used in scientific computing) makes this easy

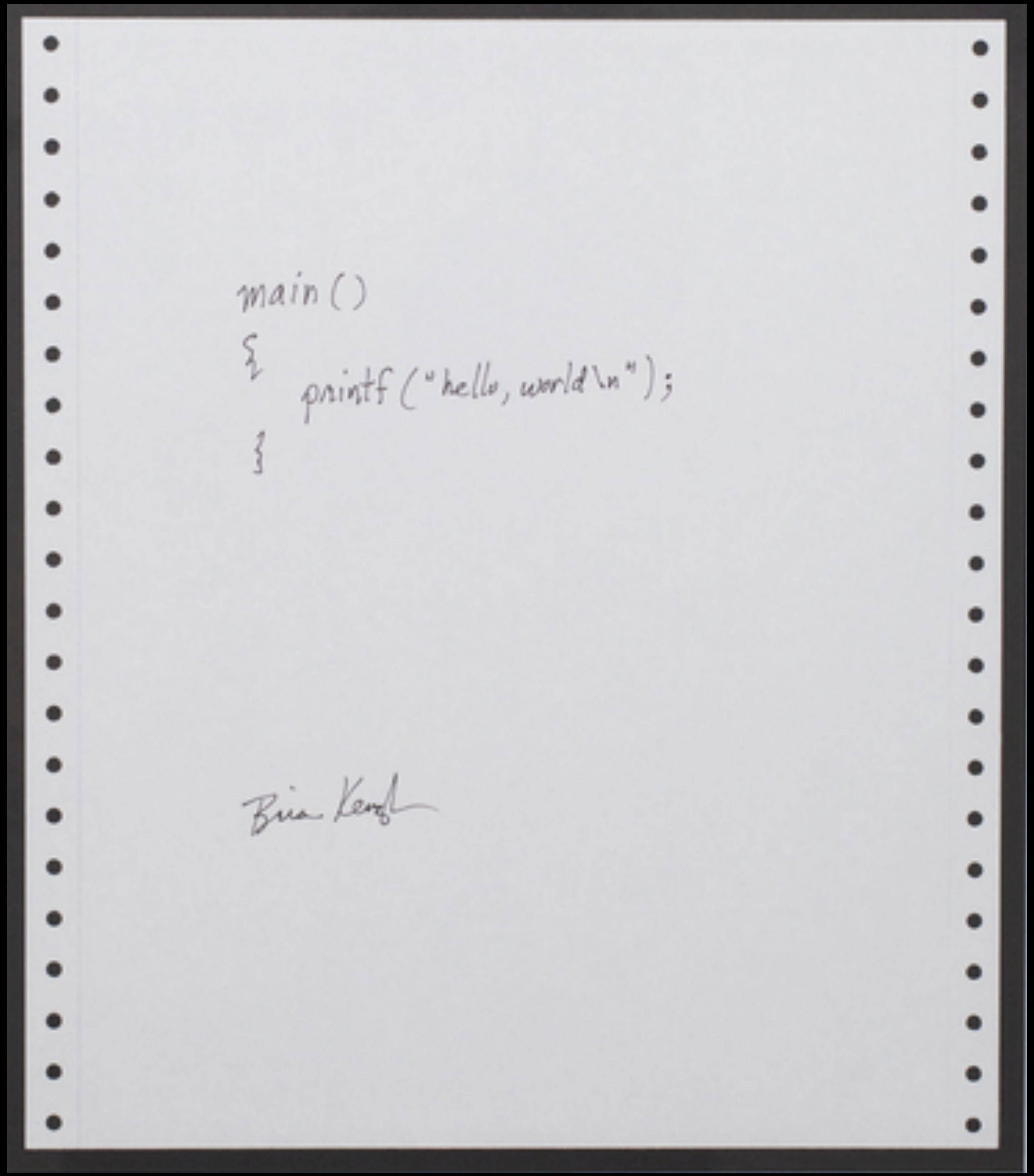
Try:

```
print("Hello, world!")
```

- Print basically anything

Try:

```
print(4*4+4-4)
```



Brian Kernighan
(early UNIX developer), 1978

Libraries

- Don't reinvent the wheel when you want to hit the road
  - (But OK if you want to learn how to make wheels)
- Python has *many* libraries for numerical computing & everything else
- By "Libraries", I mean any pre-written code that you can use in your programs

Try in tutor:

```
import math
print(math.pi)
```

Math

Try in tutor (only type the left hand side of the ==):

- Arithmetic operations built in

```
(4 + 4) * 4 / 4 - 4
```

- Exponents with **

```
4 ** 4 == 256
```

- Scientific notation

```
4e4 == 40000
```

- The rest in the math library

```
math.sin(4)
math.sqrt(4)
```

Expressions

- Value = piece of data of a particular type
- Type = kind of data
- Operator = combine values to get a new value
  - Behavior depends on type
- Expression = group of values and operators
- Python evaluates expressions, like a calculator

`4.444`	`“Hello world”`
`float`	`string`
`+ - * /`	`+`
`4.0 * 3.0 - 2.0`	`"Hello" + " world"`

Clicker question #2.0

- What does Python get when it evaluates this expression? `4.0 * 3.0 - 2.0`

A 4.0

B 10.0

C Some other number

D An error

Try out some expressions

```
4.0 * 3.0 - 2.0
```

```
"Hello" + " world"
```

Try out some expressions

```
print(4.0 * 3.0 - 2.0)
```

```
print("Hello" + " world")
```

```
#make up your own
```

Some types we will need

- Float
- Int
- String
- Boolean

Type: float

- **Values**: real numbers ("numbers with decimal points")
  - Examples `4.1234` `4.0` `4.4e2` `-5.2e-3`
  - If you don't include a decimal point, it is an integer!
- **Operators:** `+ - * / **`

Try in tutor:

```
print(22.0 / 7.0)
```

```
print(-3.0e-3 * 10.0)
```

```
print(8.0**2.0)
```

```
print(1.0/3.0)
```

```
print(type(4))
print(type(4.0))
```

Type: int

- **Values**: integers (whole numbers, positive, negative, zero)
  - Examples `-4` `742352046` `7` `-33`
    - Don't use commas when typing an int or float
- **Operators**: `+ - * **` `/` `//` `%`

Try in tutor:

```
print(2**8)

print(4 * 3 - 2)
```

```
print(7 / 3) #float in Python3,
             #int in Python2 (avoid!)

print(7 // 3) # quotient
print(7 % 3)  # remainder
```

Clicker question #2.1

- In Python 3, what is the value of this expression?
`10 // 3 + 1`

A	4
B	4.3333333333
C	Some other number
D	An error

Type: boolean

- **Values**: true or false
  - Examples `True` `False`
- **Operators**: `and` `or` `not`
  - `a and b` is true if both are true, false otherwise
  - `a or b` is true if a is true, b is true, or both are true
    is false if both a and b are false
  - `not a` is true if a is false, false if a is true

= and ==

- = stores results in a named object ("variable")

```
myNumber = 4
print(myNumber * myNumber)
```

```
print(myNumber * myNumber == 16)
True
```

- == tests whether two objects are equal

```
print(2 + 2 == 5)
False
```

Try some of these

- = stores results in a named object ("variable")
- == tests whether two objects are equal

```python
print(2 + 2 == 4 and 3 + 3 == 6)
print(2 + 2 == 4 and 3 + 3 == 7)
print(2 + 2 == 4 or 3 + 3 == 7)
print(not 3 + 3 == 7)
```

```python
a = True
b = True
c = False
d = False
```

```python
# Pick a few of these
print(a)
print(not c)
print(not a)
print(a or b)
print(a or c)
print(c or d)
print(a and b)
print(a and c)
print(c and d)
```

Converting types

Try in tutor:

```
q = 4
print("The number is "+q)
```

```
q = 4
print("The number is "+str(q))
```

```
print(type(4))
print(type(str(4)))
print(type(float(4)))
```

Clicker question #2.2

- What does this line print?

```
import math
print("The value of pi is "+math.pi)
```

A The value of pi is 3.141592653589793

B The value of pi is math.pi

C Something else but not an error

D An error

Clicker question #2.2

- What does this line print?

```
import math
print("The value of pi is "+str(math.pi))
```

A The value of pi is 3.141592653589793

B The value of pi is math.pi

C Something else but not an error

D An error

Comments

- Comments explain what you're doing
- Use comments to explain your code
- Use names that help explain, even without comments

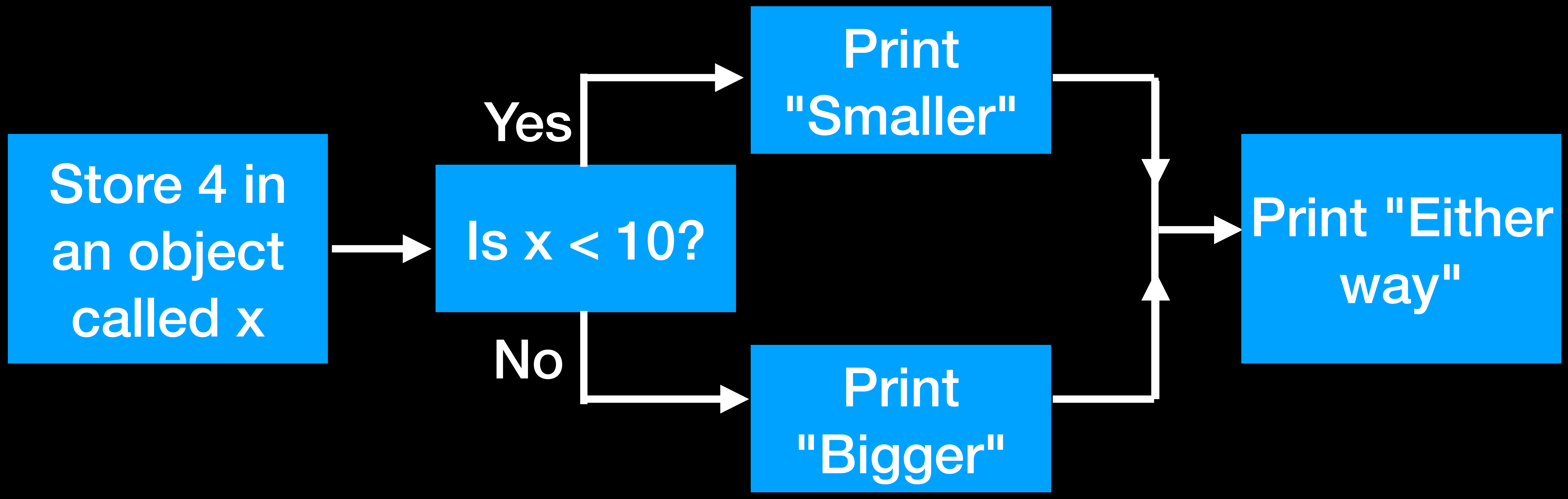
```python
# Say hello to someone by name
personName = "Geoffrey"
print("Hello " + personName)
```

If/else

- If does the first indented thing if the stuff in () is True
- Otherwise it does the indented stuff under "else"

```
x = 4
if(x < 10):
    print("Smaller")
else:
    print("Bigger")
print("Either way.")
```

Try in tutor!



Clicker question #2.2b

- What does this program print?

```
x = 4
if x==10 or x==11:
    print('yes')
else:
    print('no')
```

A Yes

B No

C The code gives an error

Clicker question #2.2

- What does this program print?

```
x = 4
if x==10 or 11:
    print('yes')
else:
    print('no')
```

A Yes

B No

C The code gives an error

Clicker question #2.2

- What does this program print?

```
x = 4
if x==10 or 11:
    print('yes')
else:
    print('no')
```

```
x = 4
if false or true:
    print('yes')
else:
    print('no')
```

A Yes

B No

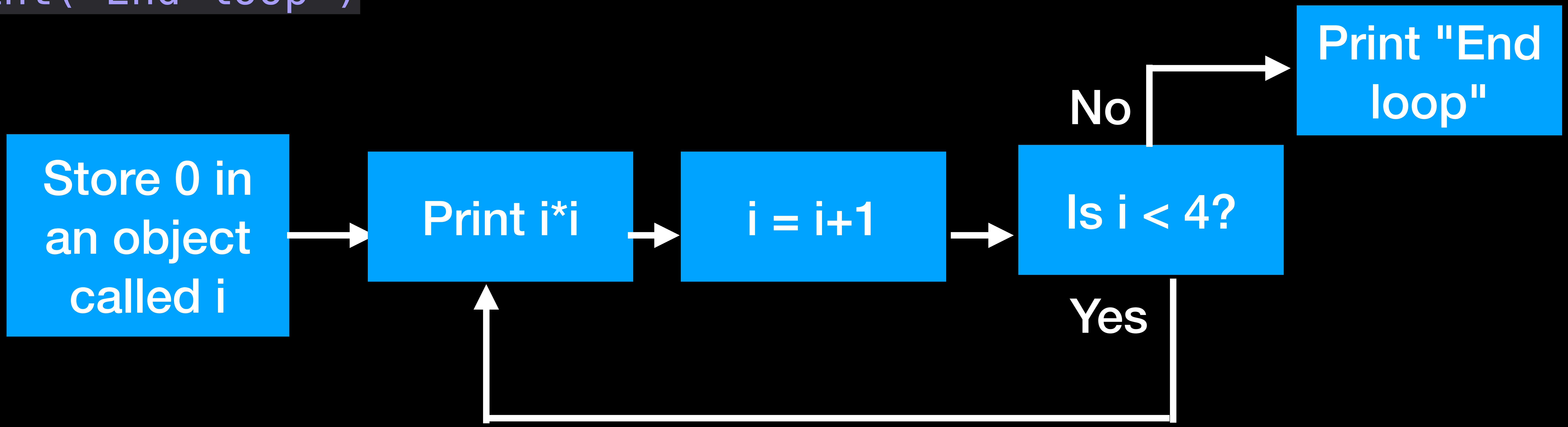
C The code gives an error

Try in tutor!

Loops

```
i = 0
while i < 4:
  print(i*i)
  i = i + 1
print("End loop")
```

```
0
1
4
9
```



Loops

```
for i in [0,1,2,3]:
  print(i*i)
```

```
0
1
4
9
```

```
i = 0
while i < 4:
  print(i*i)
  i = i + 1
```

```
0
1
4
9
```

So far, our programs just run & stop...

How do programs with a user interface work?

Clicker question #2.3

- What does this program print?

```
j = 1
while j < 3:
    j = j + 1
print(j)
```

A	1
B	2
C	3
D	4

Clicker question #2.4

- What does this program print?

```
product = 1
j = 1
while j < 3:
    product = product * j
    j = j + 1
print(product)
```

A	1
B	2
C	6
D	24

Clicker question #2.4b

- What does this program print?

```
product = 1
j = 1
while j < 4:
    product = product * j
    j = j + 1
print(product)
```

A	1
B	2
C	6
D	24

Clicker question #2.4c

- What value of x makes the program print 24?

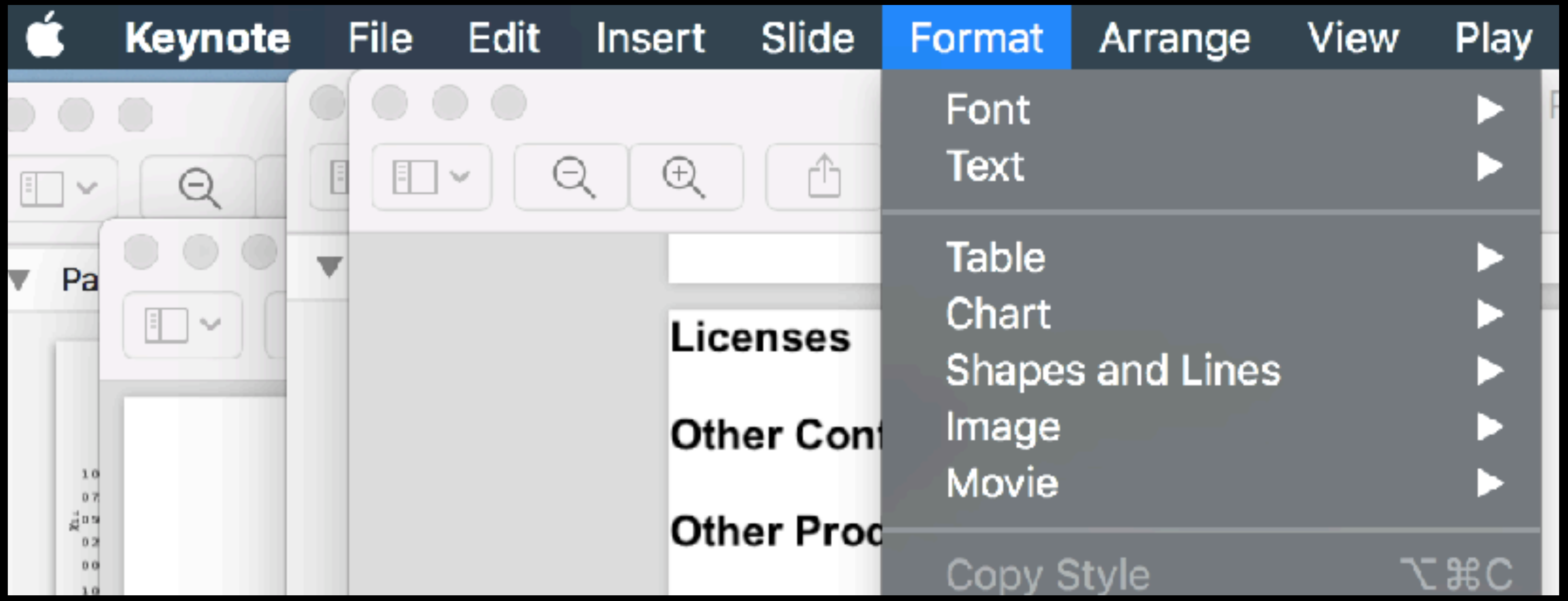
```
product = 1
j = 1
while j < x:
    product = product * j
    j = j + 1
print(product)
```

- A 3
- B 4
- C 5
- D 6

Loops

- Real life:
  event loop
- Event = key press,
  mouse/trackpad
  click,
  …

```
while message != quit:
  message = get_next_message()
  process_message(message)
```



My first program

- Basic, 1987

```
10 PRINT "GEOFFREY"
20 GOTO 10
```

- Python equivalent

```
done = False
while not done:
  print("Geoffrey")
```

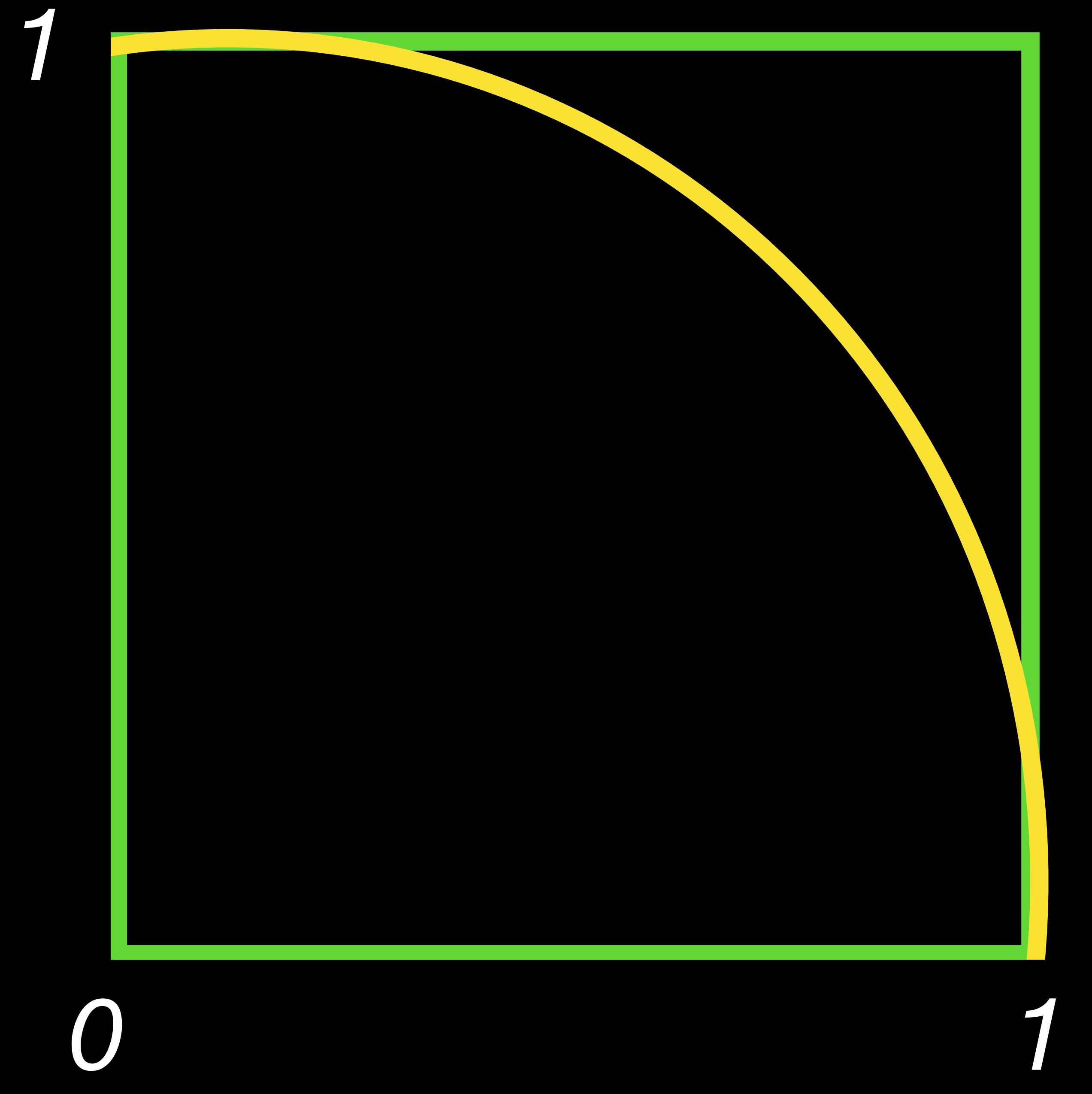
```
while true:
  print("Geoffrey")
```

Course web page

- https://geoffrey-lovelace.com/Workshop/2024
  - Cheat sheets for python & unix
  - Links to places where you can run python notebooks for free
  - Slides from the workshop

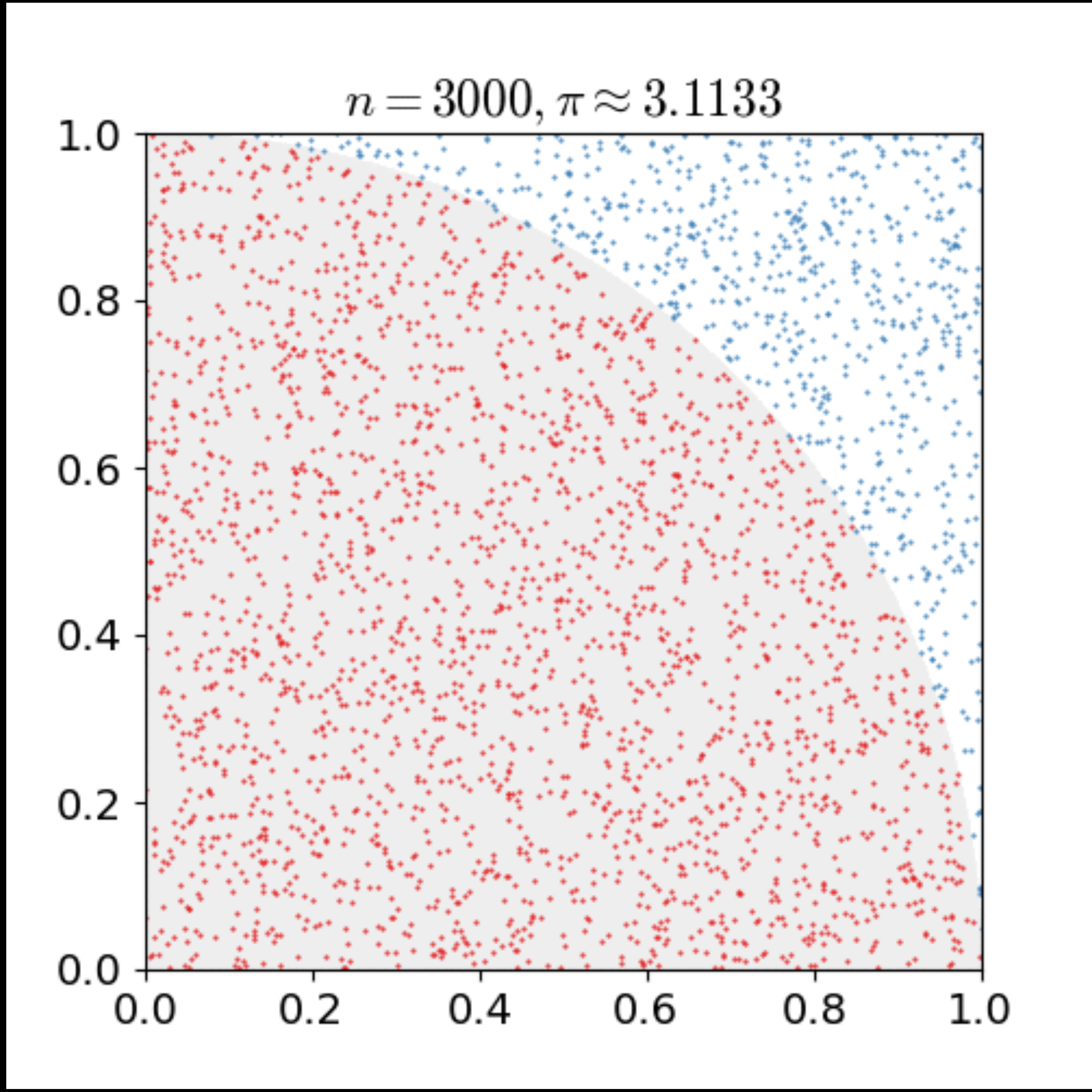
A silly way to compute π

- Area of circle?
- Area of square?
- Idea: throw darts in square
  - (circle area) ÷ (square area) ≈ darts in circle ÷ darts in square = “hits” / (“hits” + “misses”)



A silly way to compute π

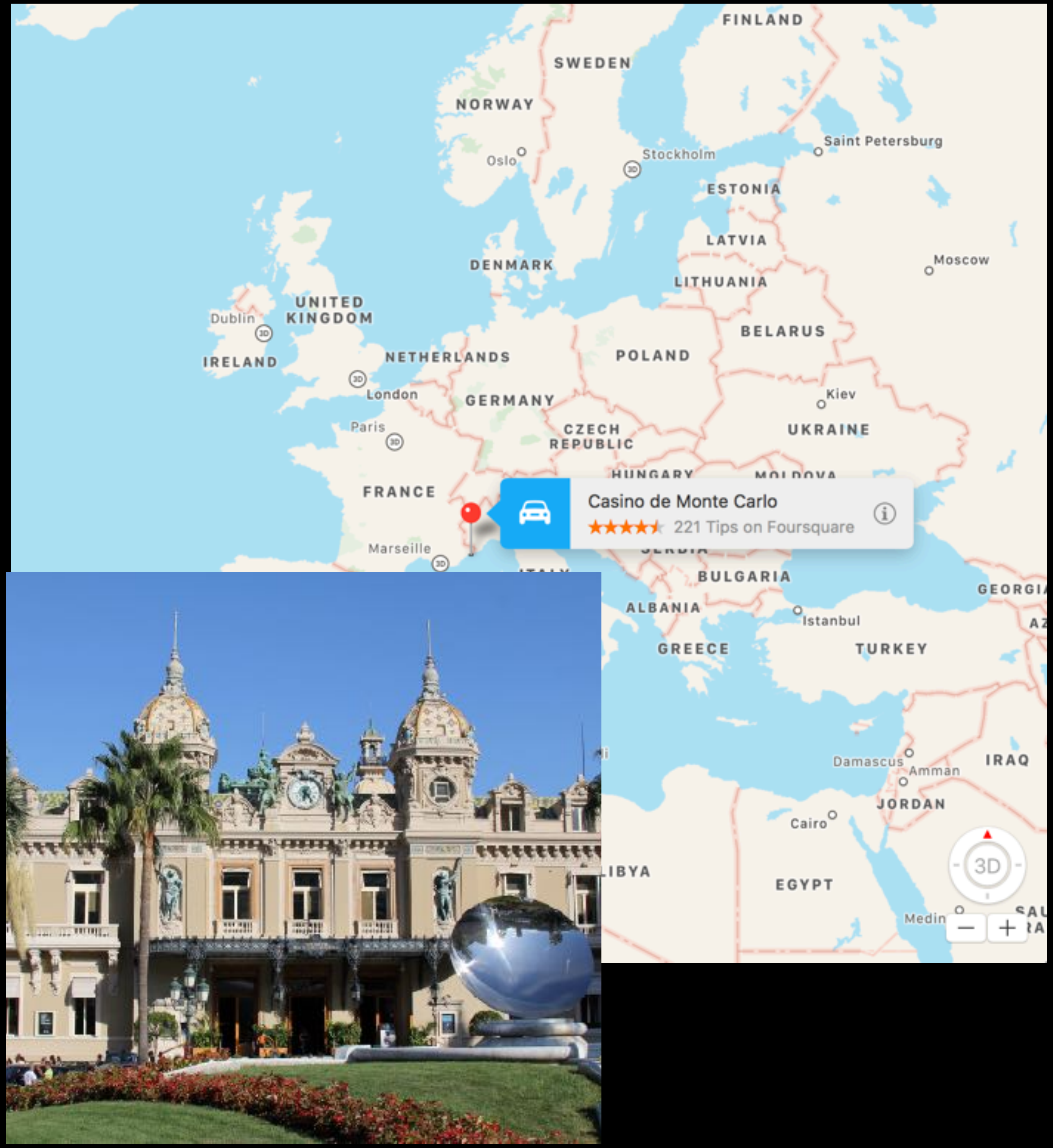
- Throw darts in square
  - (circle area) ÷ (square area) ≈ darts in circle ÷ darts in square = π/4



Courtesy wikipedia

Monte Carlo methods

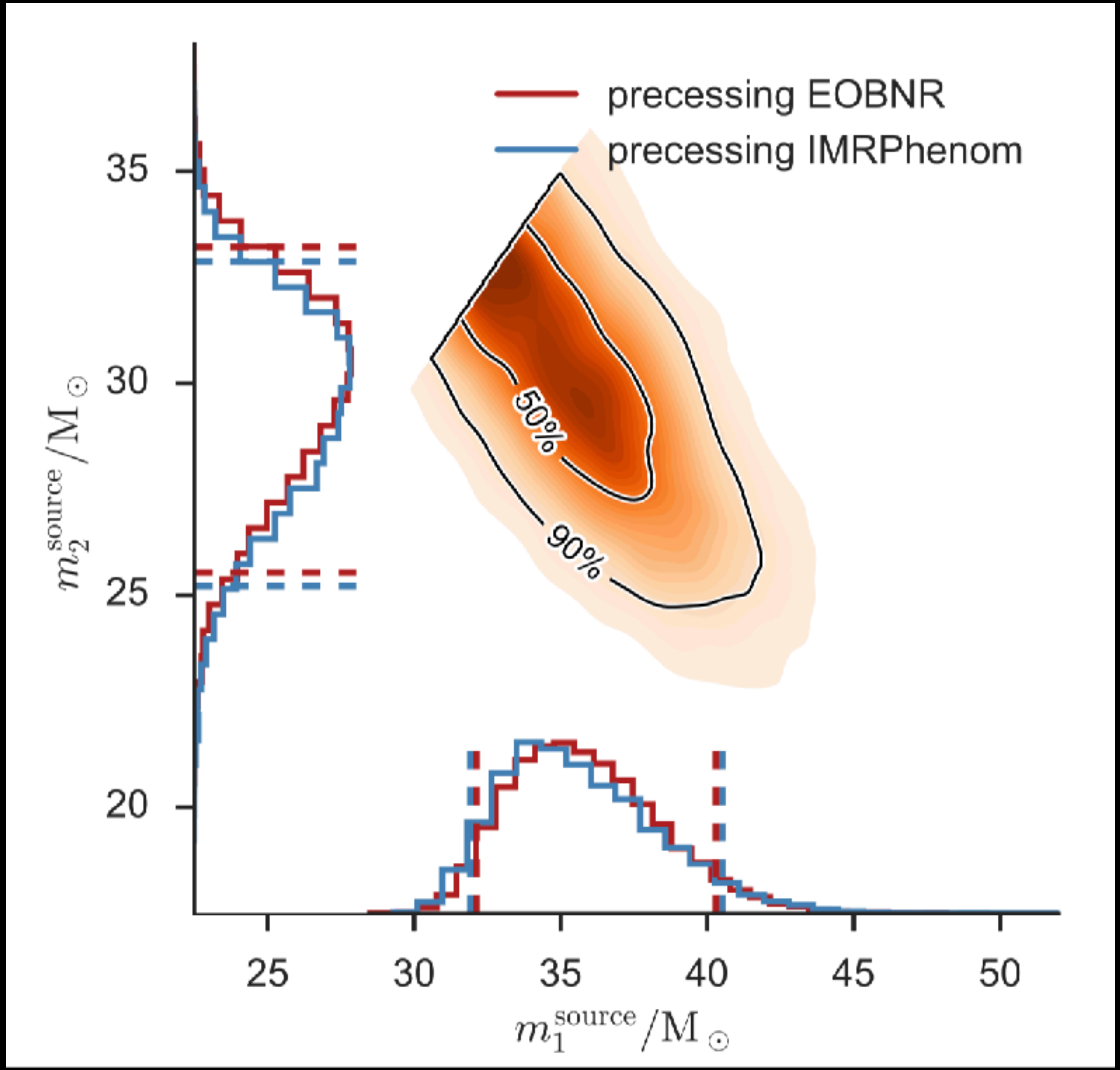
- This idea might seem silly, but it actually has a lot of uses in physics
- **Monte Carlo methods: use repeated random numbers to get results**
  - Min/max of functions
    especially functions of many variables
  - Integrals
    especially high dimensional
  - Explore probability distributions



Images courtesy Wikipedia, Apple Maps

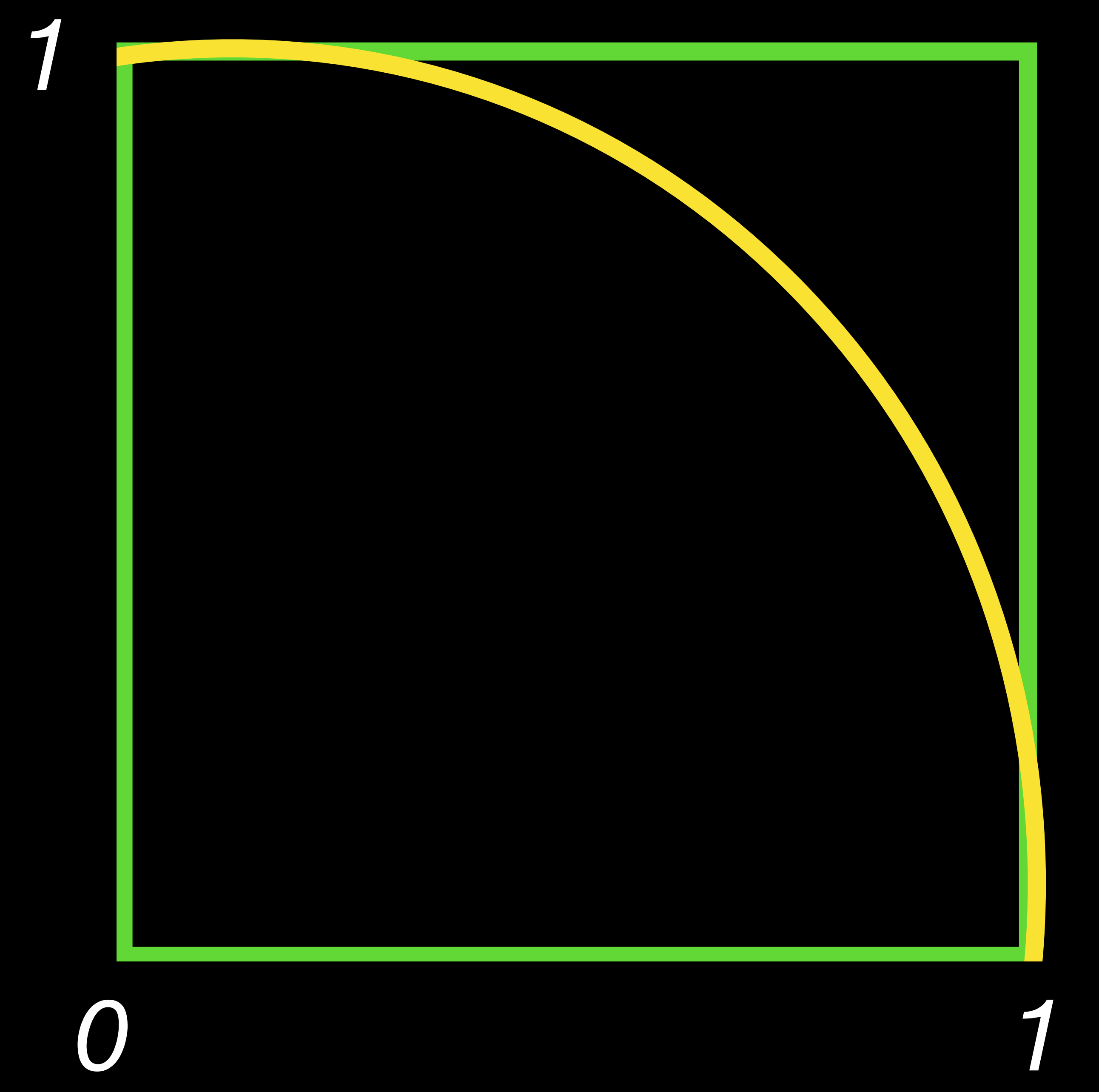
Monte Carlo methods

- This idea might seem silly, but it actually has a lot of uses in physics
- When we observe a gravitational wave from merging black holes…
  - What kinds of black holes made the waves?
    - Choose random parameters (masses, spins, …)
    - Compute the corresponding grav. wave
    - More likely to call the wave a “hit” the better it matches—vs. the last wave “hit”



GW150914: Abbott+ (2016)

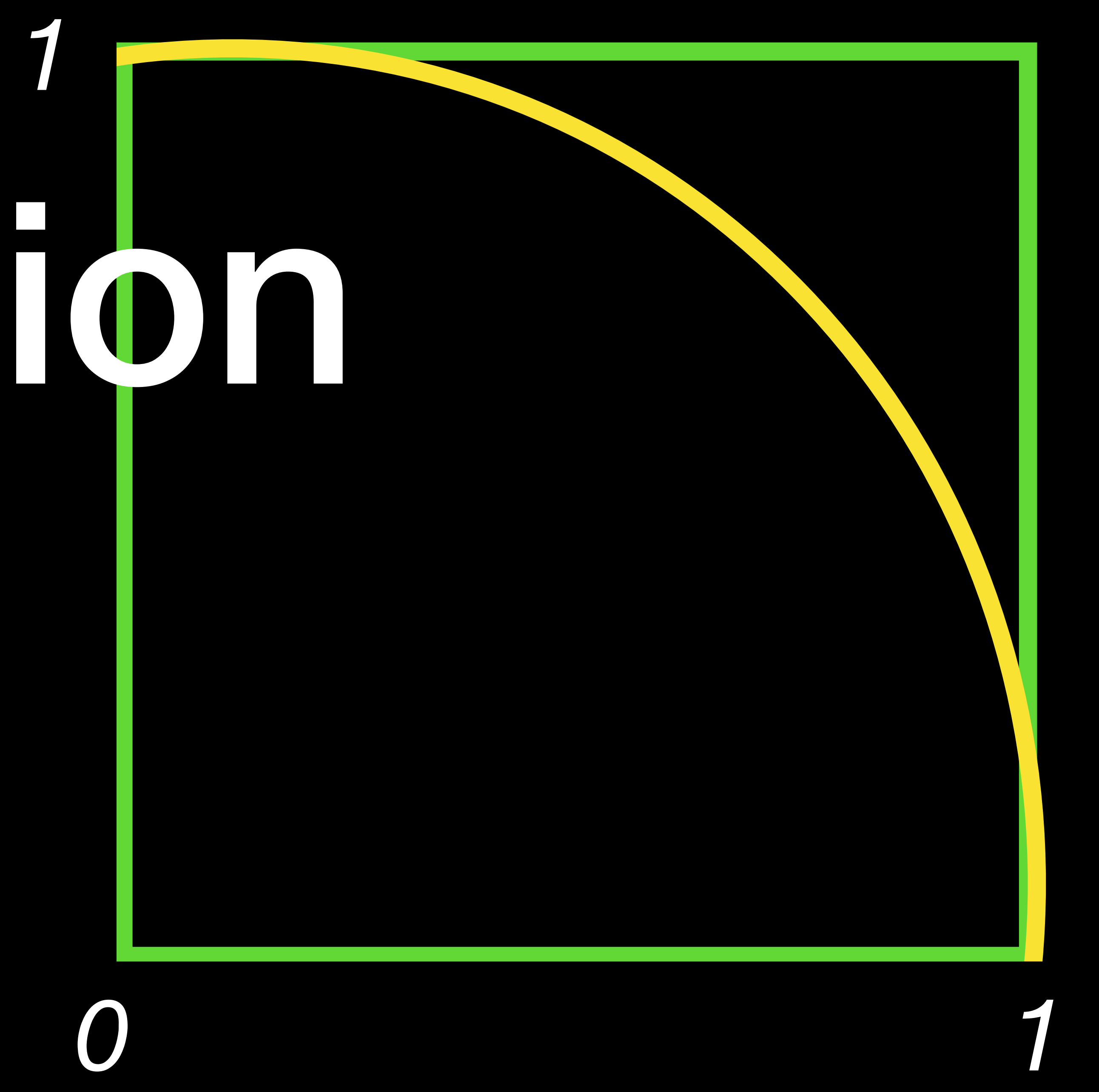
Pi Dartboard 1



- Write a program that prints one random number between 0 and 1

```python
import math
import random

print(random.random())
```

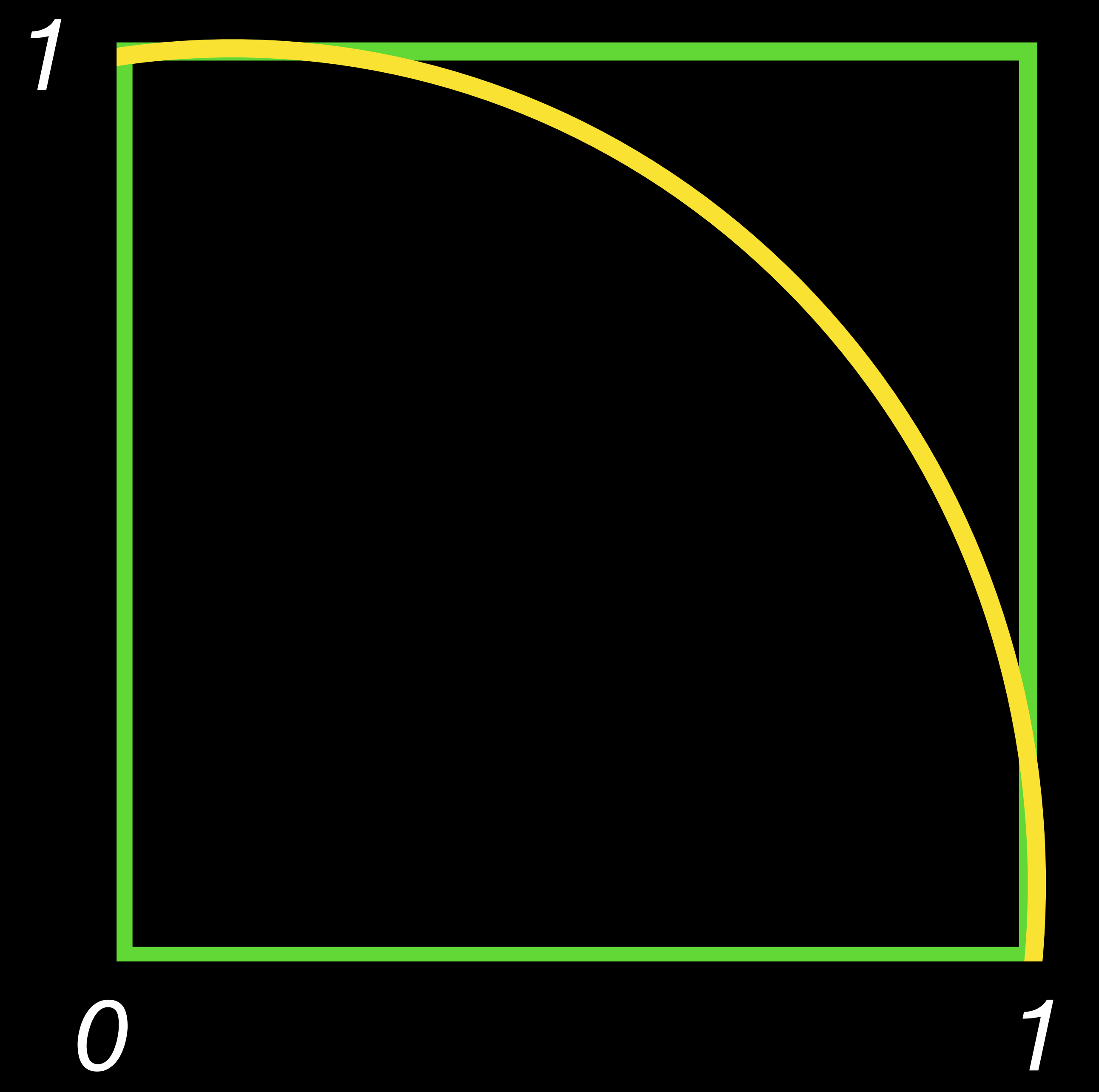
Pi Dartboard 2 Solution



- **Challenge:** Modify your program
  - Store the random number in a variable x
  - Store a second random number in a variable y
  - Print x and y

```
import math
import random

x = random.random()
y = random.random()

print(x)
print(y)
```

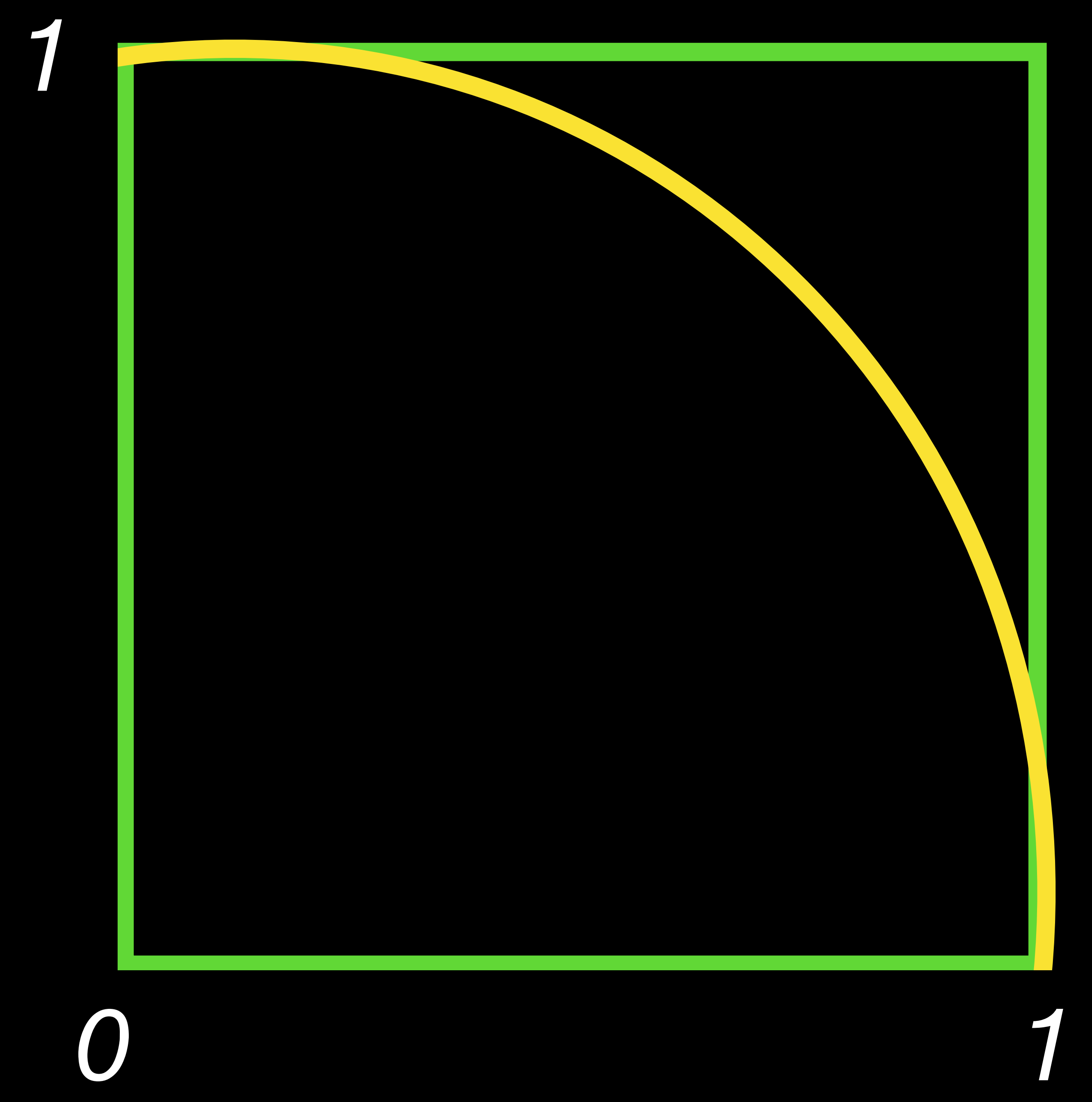
Pi Dartboard 3



- **Challenge:** Modify your program
  - Print $x^2 + y^2$ instead of just x and y

```
import math
import random

x = random.random()
y = random.random()

print(x)
print(y)
```

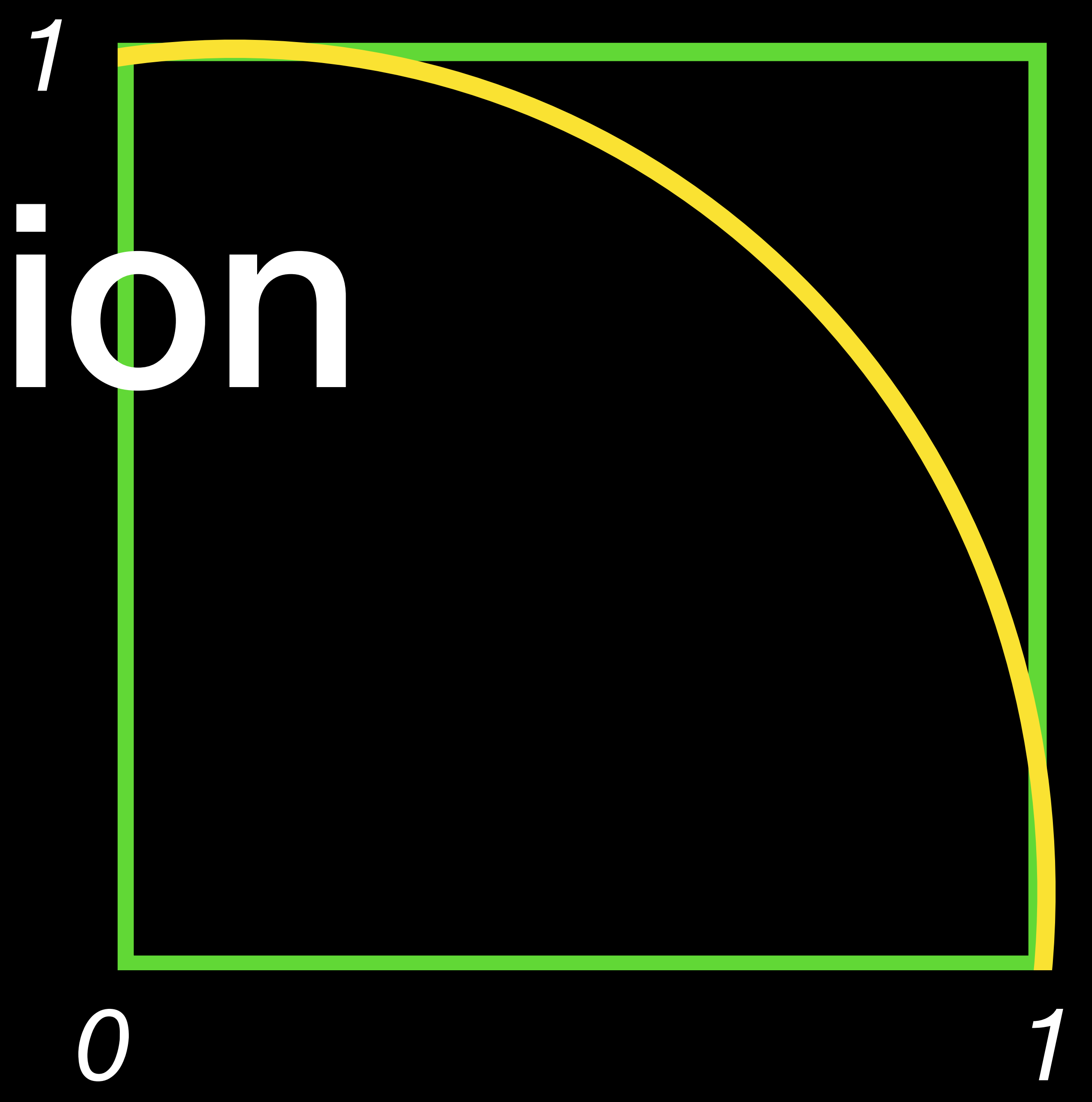
Pi Dartboard 3



- **Challenge:** Modify your program
  - Compute $x^2 + y^2$ and store it in a variable rSquared
  - Print rSquared instead of just x and y

```
import math
import random

x = random.random()
y = random.random()

print(x)
print(y)
```

Pi Dartboard 3 Solution



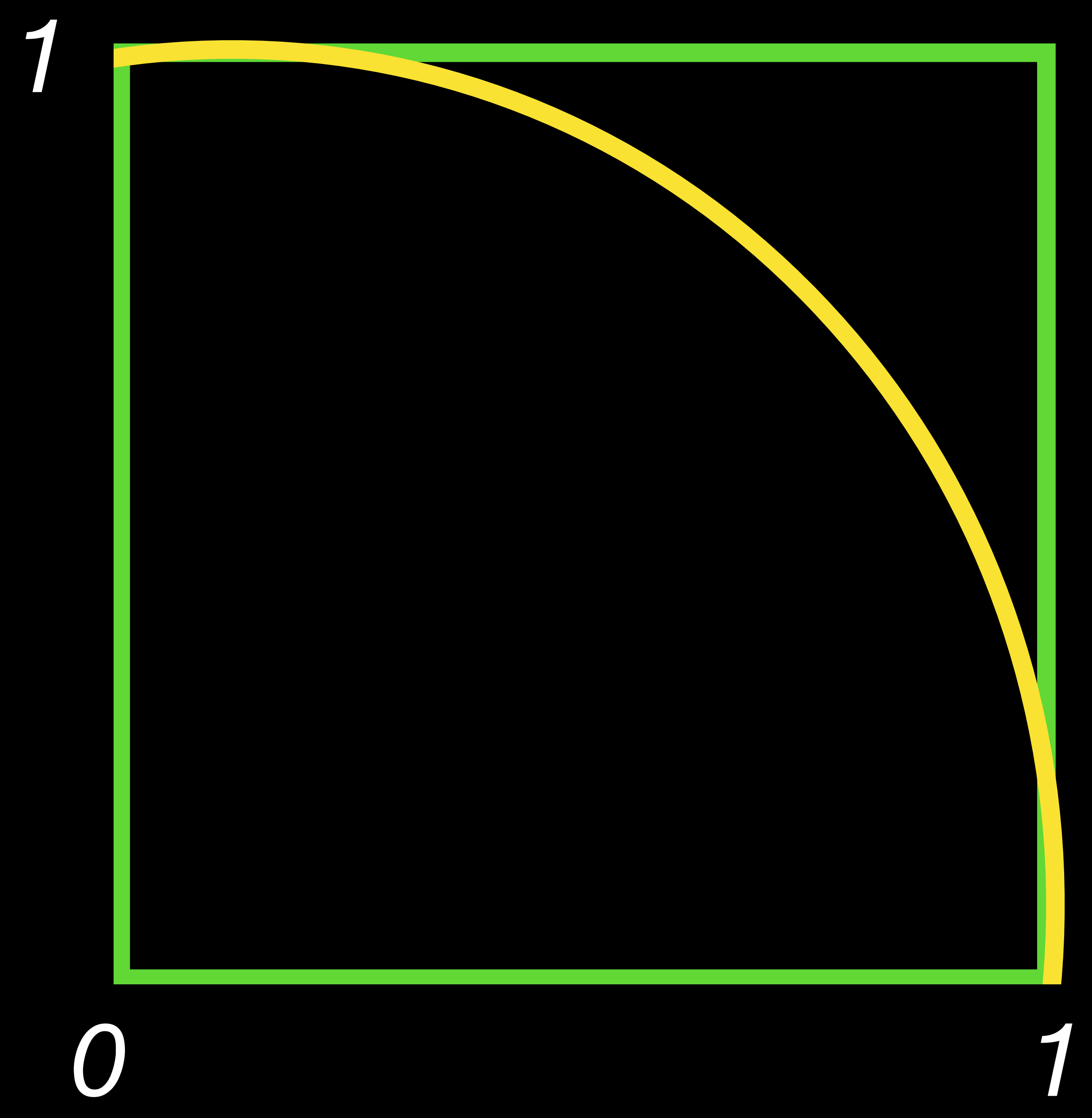
- **Challenge:** Modify your program
  - Compute $x^2 + y^2$ and store it in a variable rSquared
  - Print rSquared instead of just x and y

```
import math
import random

x = random.random()
y = random.random()

rSquared = x**2 + y**2
print(rSquared)
```

Clicker question #2.5

- Which could be a number the program prints?

```
import math
import random

x = random.random()
y = random.random()

rSquared = x**2 + y**2
print(rSquared)
```



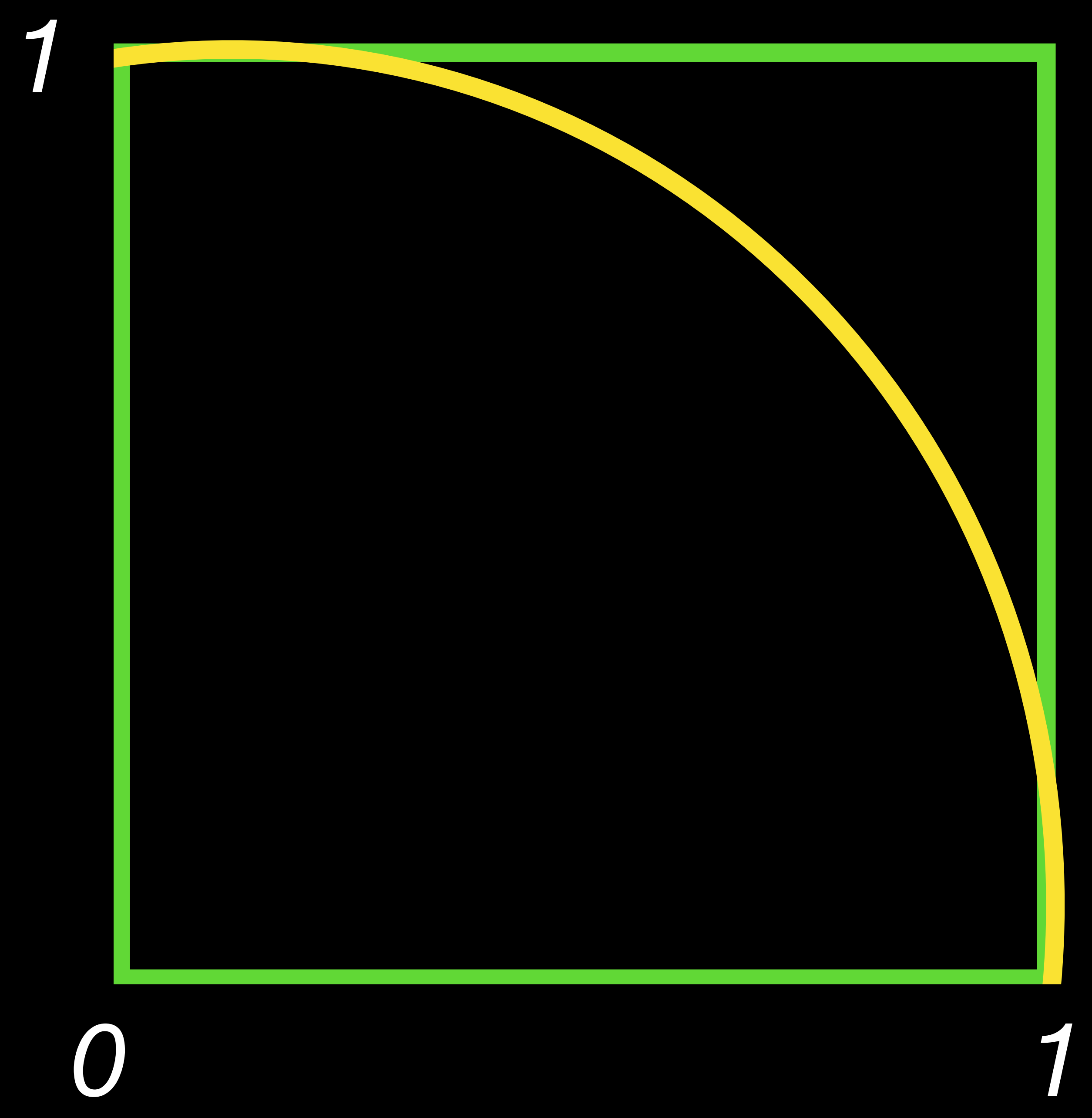
A -1.51

B 2.43

C -0.32

D 1.01

Clicker question #2.5



- If the dart is inside the **circle**, which could be the number printed by the program?

```
import math
import random

x = random.random()
y = random.random()

rSquared = x**2 + y**2
print(rSquared)
```

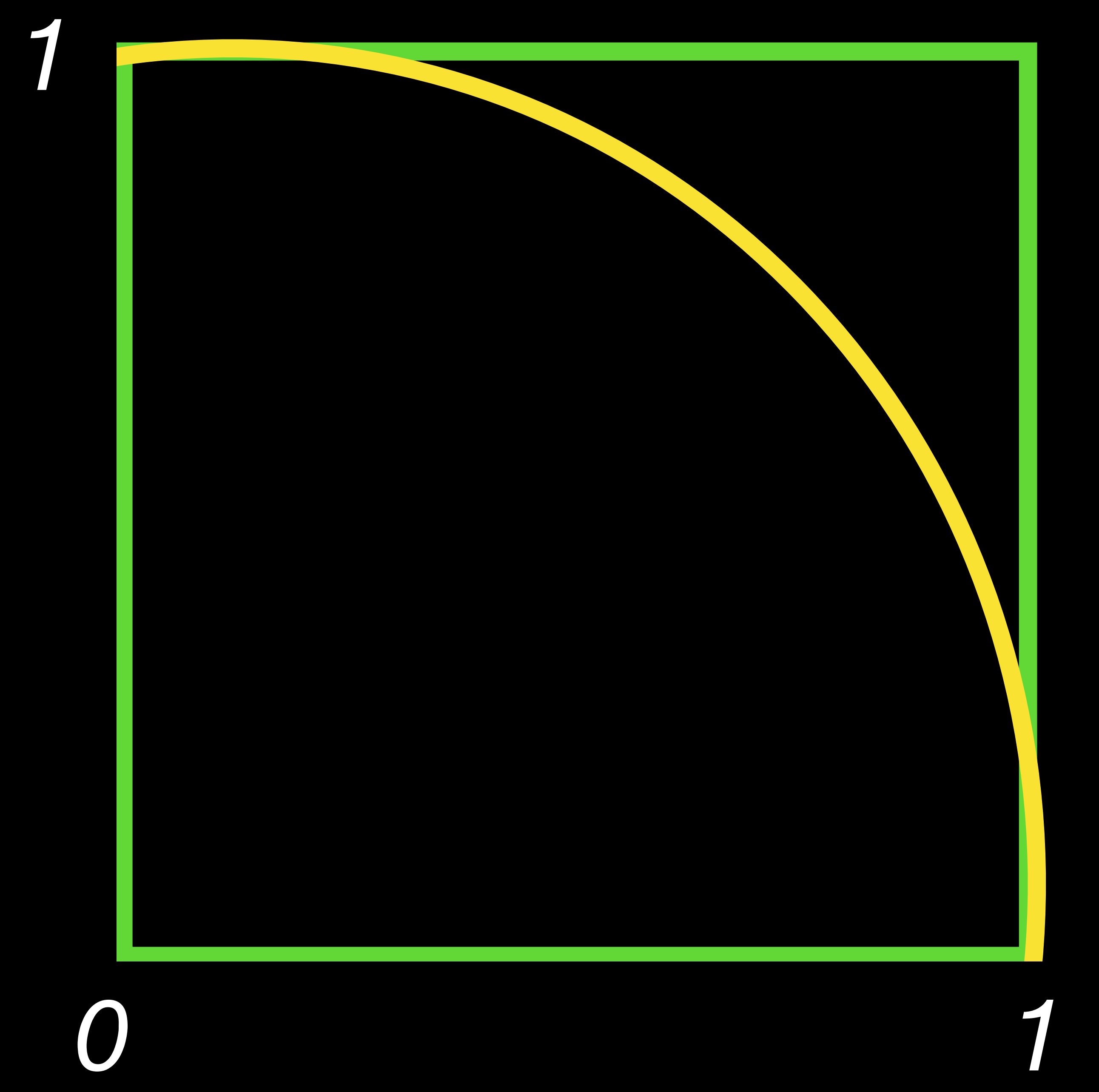
A 1.43

B 0.99

C 1.01

D More than one of ABC

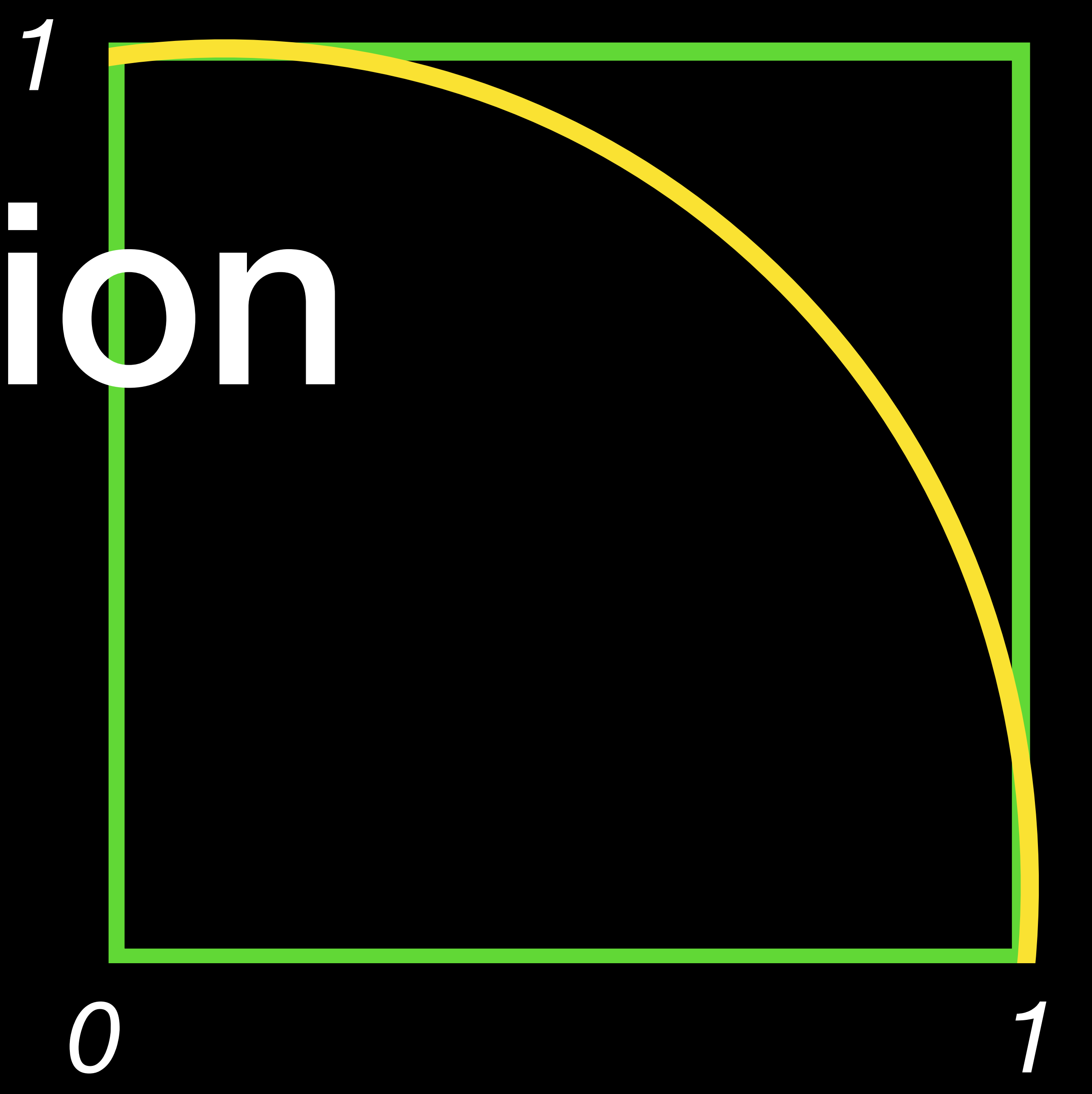
Pi Dartboard 4



- **Challenge:** Modify your program
  - Just below import random, make a new variable called “hits”, set to 0
  - If rSquared < 1, add 1 to hits
  - Print hits instead of rSquared

```
import math
import random

x = random.random()
y = random.random()

rSquared = x**2 + y**2
print(rSquared)
```

Pi Dartboard 4 Solution



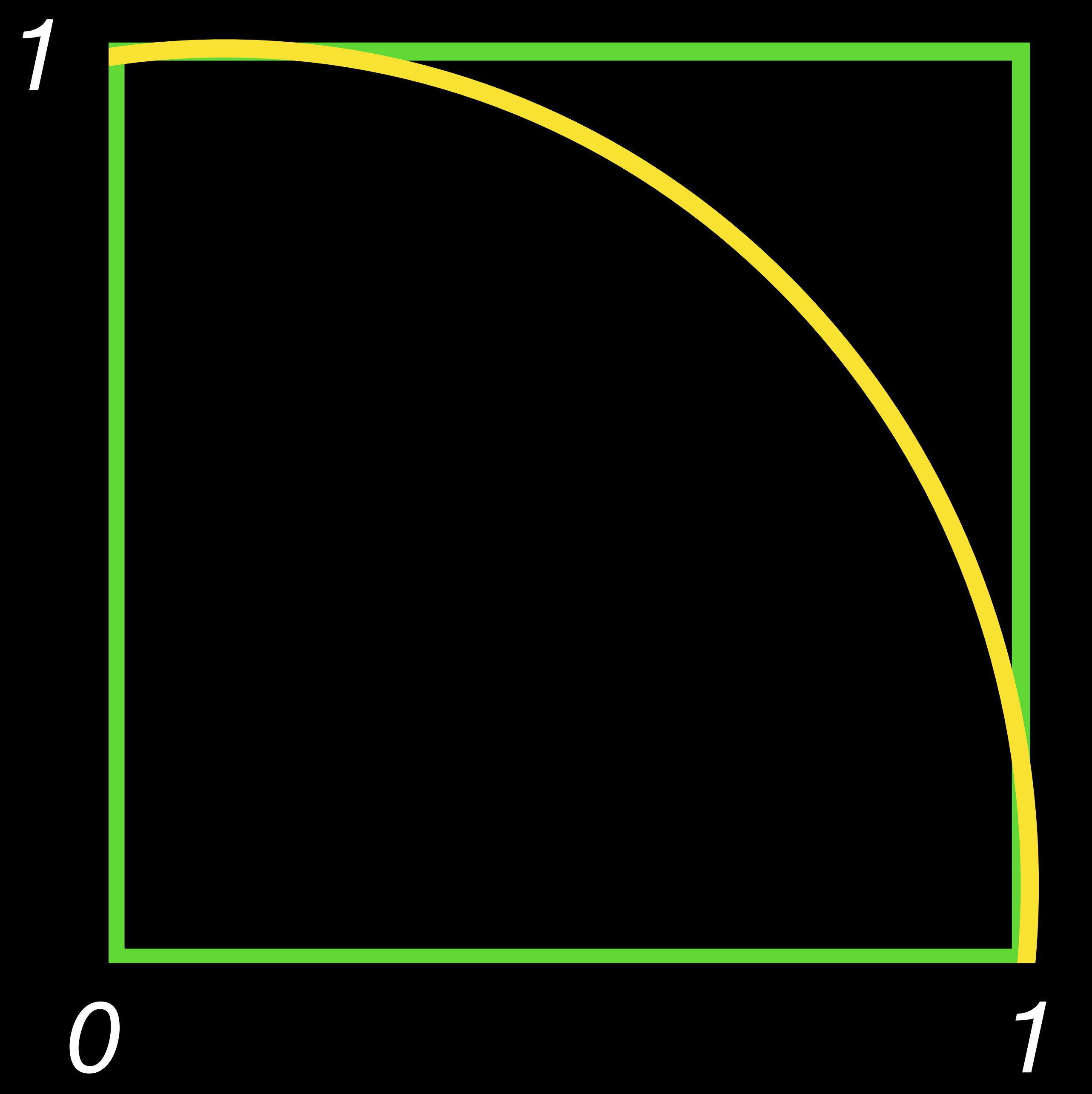
- **Challenge:** Modify your program
  - Just below import random, make a new variable called “hits”, set to 0
  - If rSquared < 1, add 1 to hits
  - Print hits instead of rSquared

```
import math
import random
hits = 0
x = random.random()
y = random.random()
rSquared = x**2 + y**2
if rSquared < 1:
    hits = hits + 1
print(hits)
```

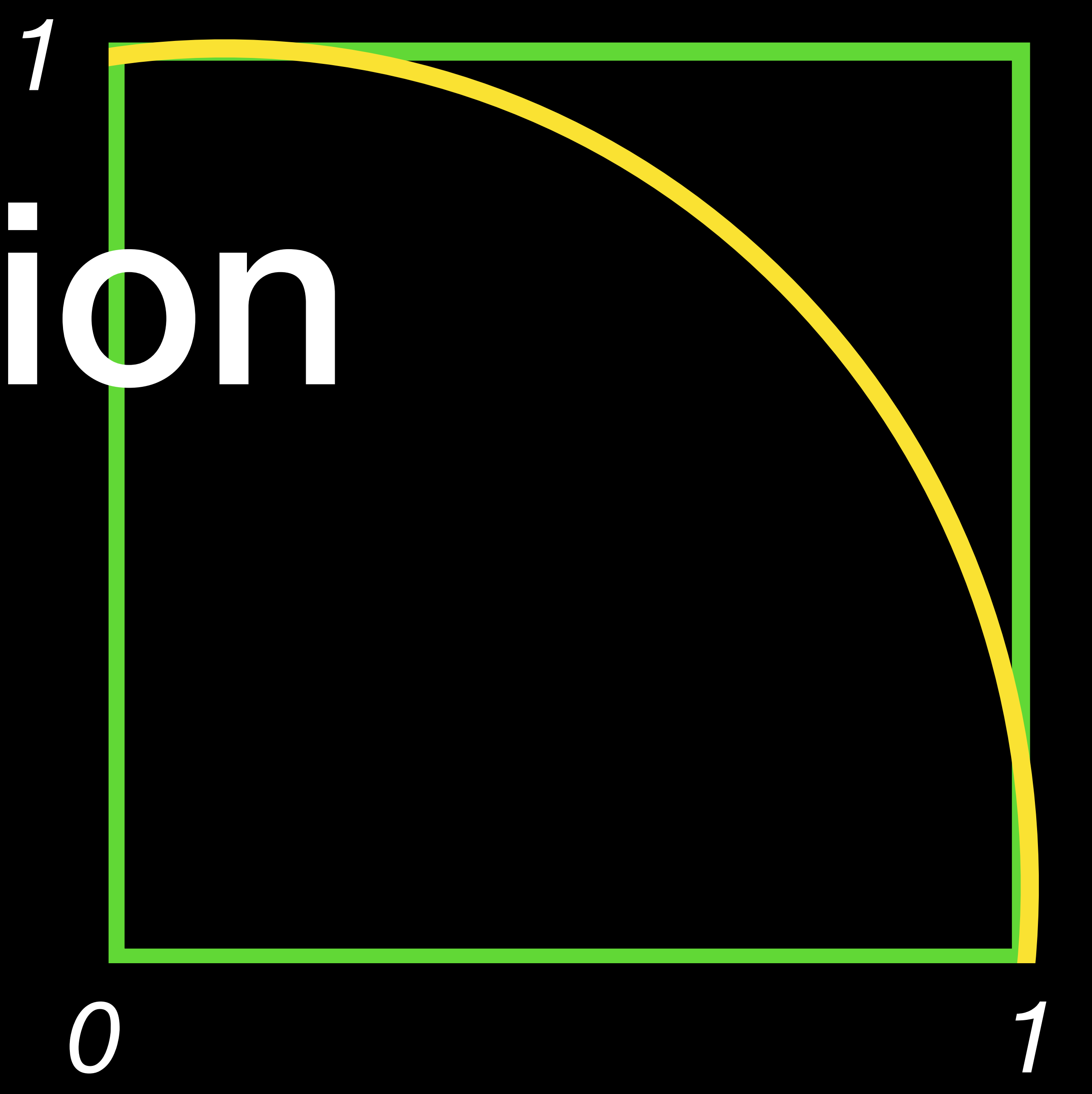
Pi Dartboard 5

- **Challenge:** Modify your program
  - Add a new variable, just below hits, called throws. Set it equal to 10.
  - Put the code that throws the dart and sees if it hit inside a while loop, so that you throw 10 darts instead of 1 dart
  - Don’t forget to increment your while loop counter variable (i or j or whatever)

```
import math
import random
hits = 0
x = random.random()
y = random.random()
rSquared = x**2 + y**2
if rSquared < 1:
    hits = hits + 1
print(hits)
```



Pi Dartboard 5 Solution

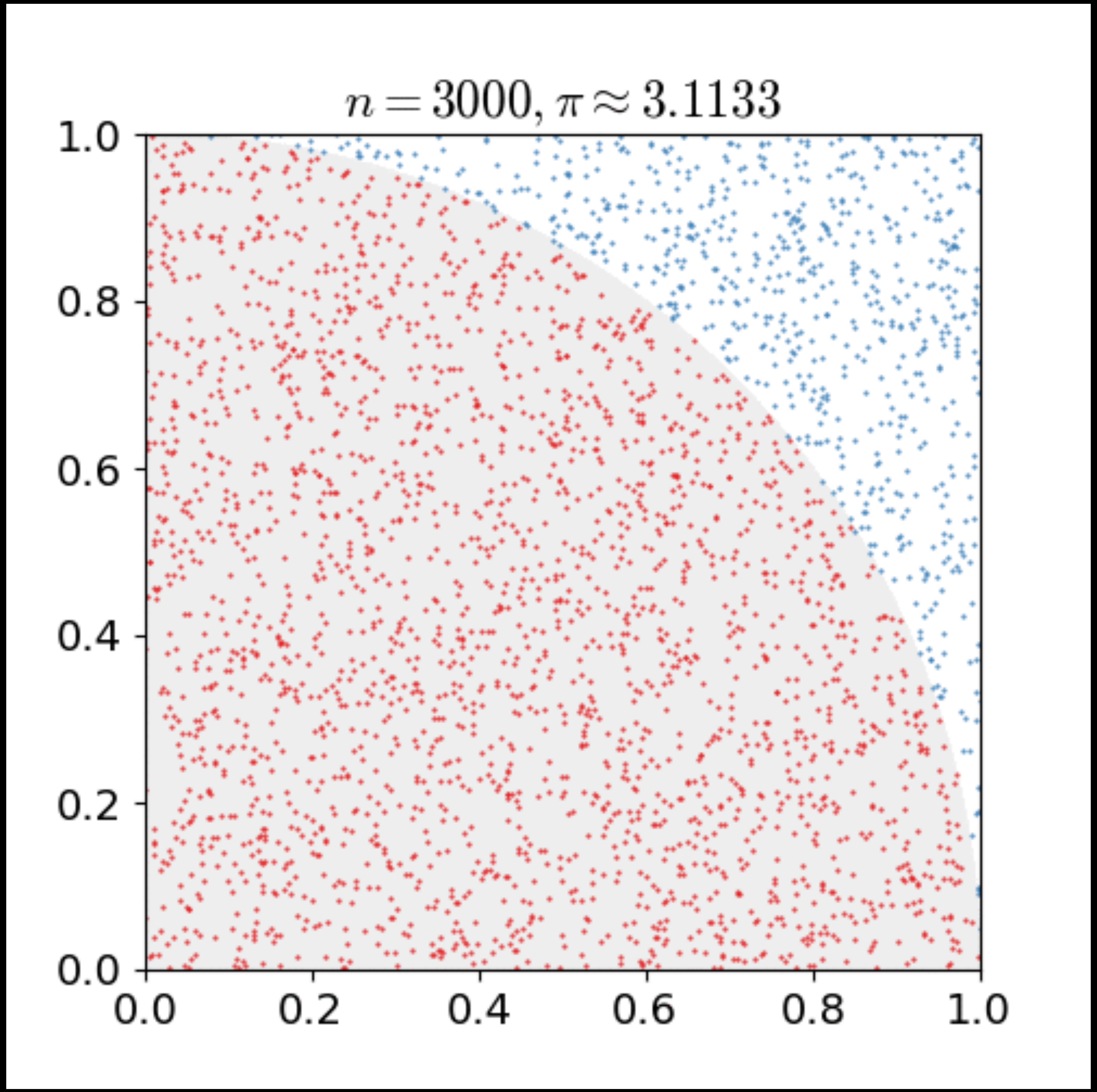


- **Challenge:** Modify your program
  - Add a new variable, just below hits, called throws. Set it equal to 10.
  - Put the code that throws the dart and sees if it hit inside a while loop, so that you throw 10 darts instead of 1 dart

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1

print(hits)
```

A silly way to compute π

- Throw darts in square
  - (circle area) ÷ (square area) ≈ hits ÷ throws = π/4
  - So π ≈ 4 * (hits ÷ throws)

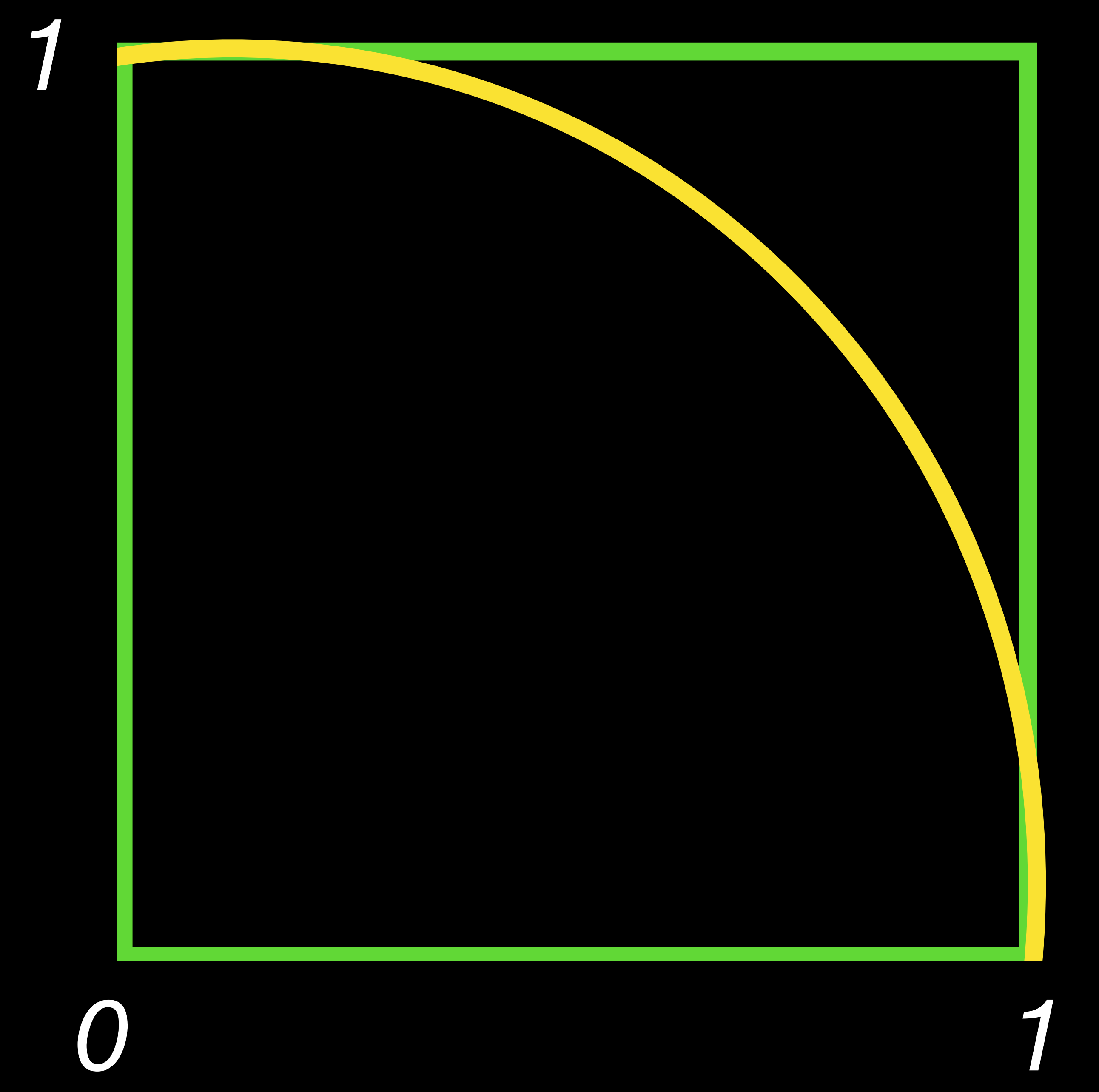


Courtesy wikipedia

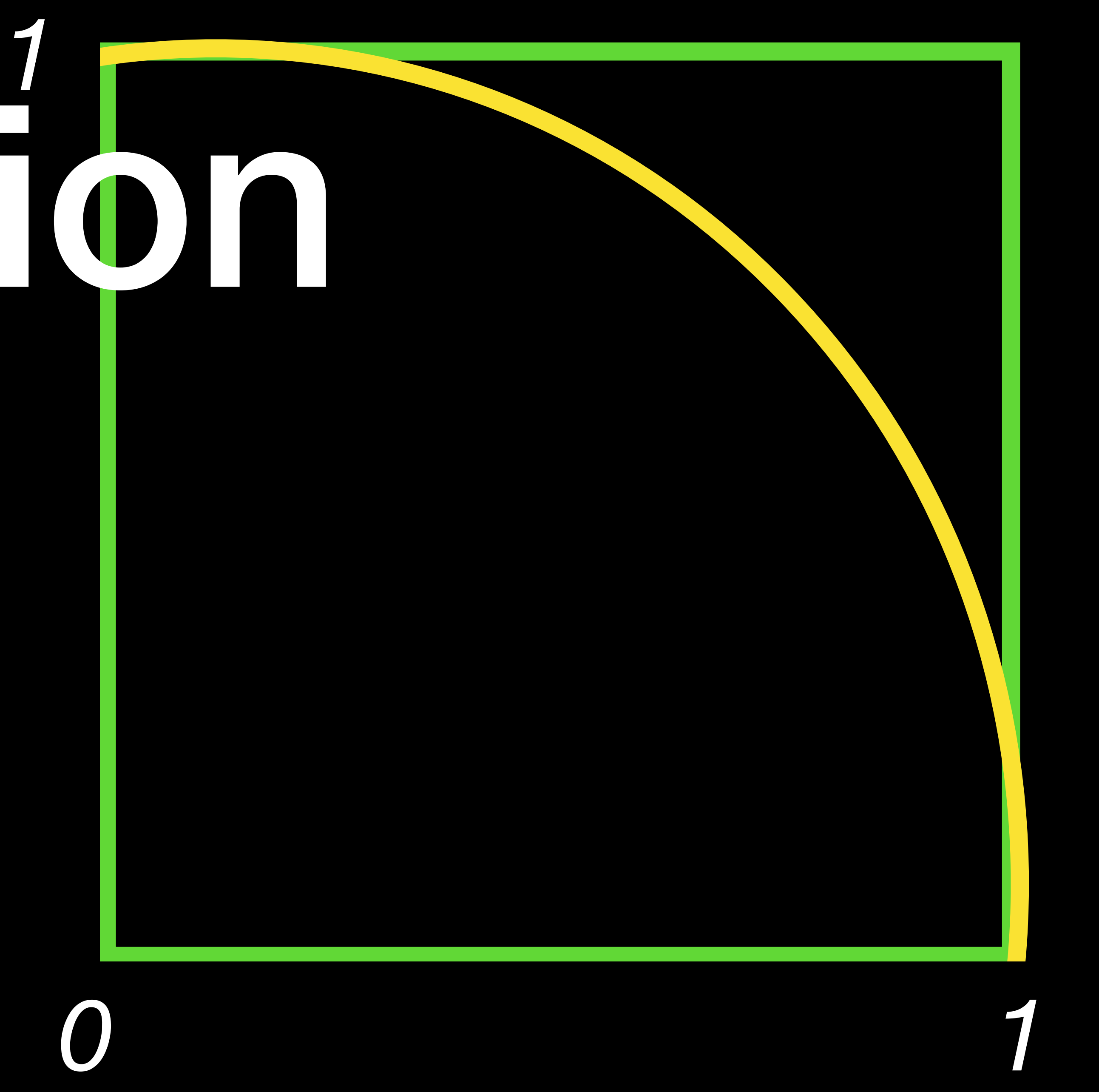
Pi Dartboard 6

- Finish the dartboard
  - Compute pi as 4.0 * float(hits) / float(throws)
  - Print your pi estimate

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1

print(hits)
```



Pi Dartboard 6 Solution



- Finish the dartboard
  - Compute pi as 4.0 * float(hits) / float(throws)
  - Print your pi estimate

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1

pi = 4.0 * float(hits) / float(throws)
print(pi)
```

Pi Dartboard 7

- See what happens as you make throws 10**n, n=1,2,3,4,5,6,7
- For n=7, how does speed compare if you do

```
rSquared = x*x + y*y
```

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1

pi = 4.0 * float(hits) / float(throws)
print(pi)
```