



# Beyond CS Bridge

# Plan For Today

- Where We've Been
- Where We Can Go Next
  - Learning Resources
  - Python
- Thank you!

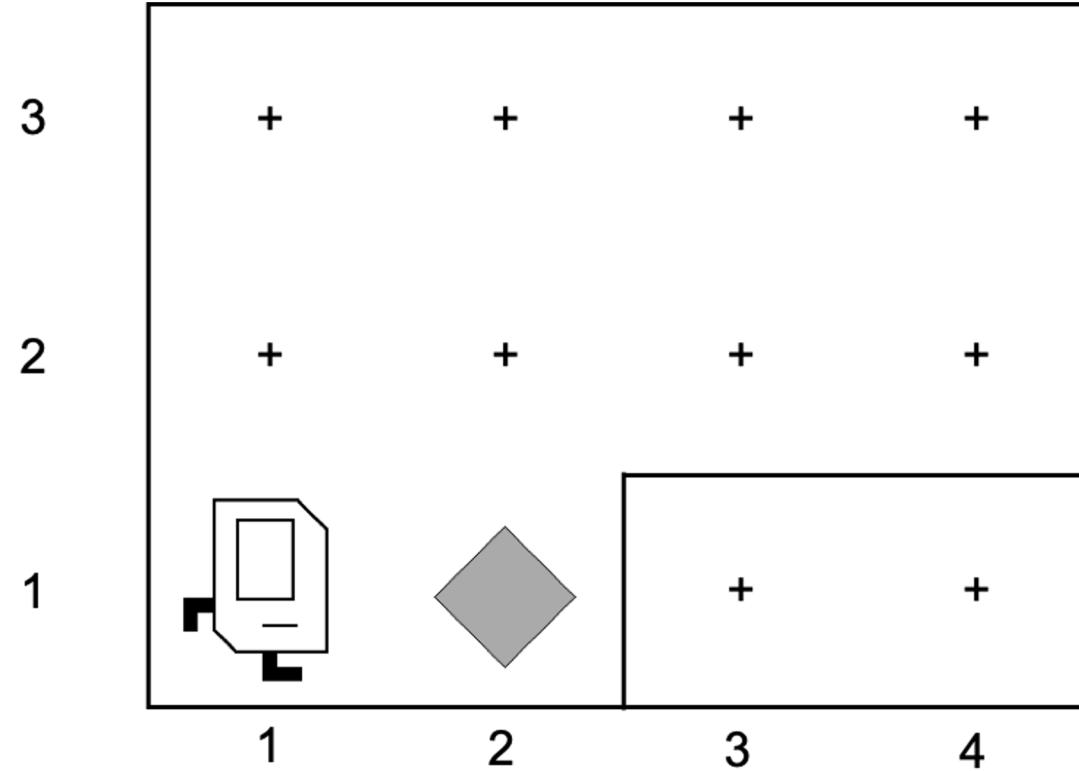
# Plan For Today

- **Where We've Been**
- Where We Can Go Next
  - Learning Resources
  - Python
- Thank you!

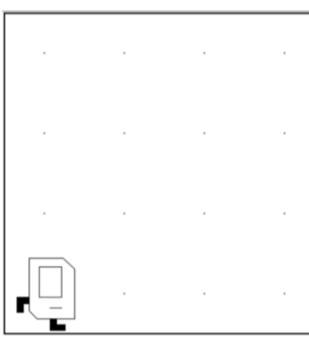
# Quick Recap

1. Karel
2. Control Flow
3. Variables
4. Control Flow Revisited
5. Graphics
6. Functions
7. Nested Loops
8. Animation
9. Lists
10. Mouse
11. Breakout
12. Keyboard
13. Dictionaries
14. Interactors
15. Artificial Intelligence
16. Machine Learning and Computer Vision
17. Music Information Retrieval

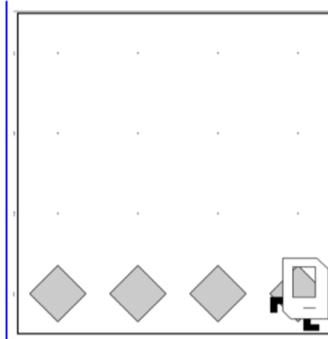
# First Day



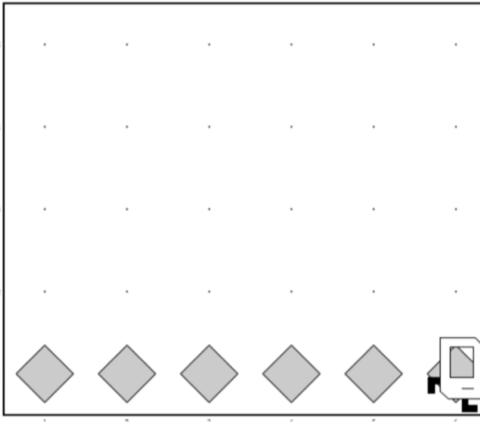
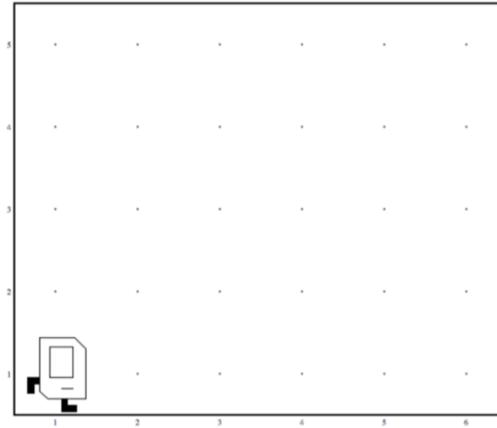
# Generalization



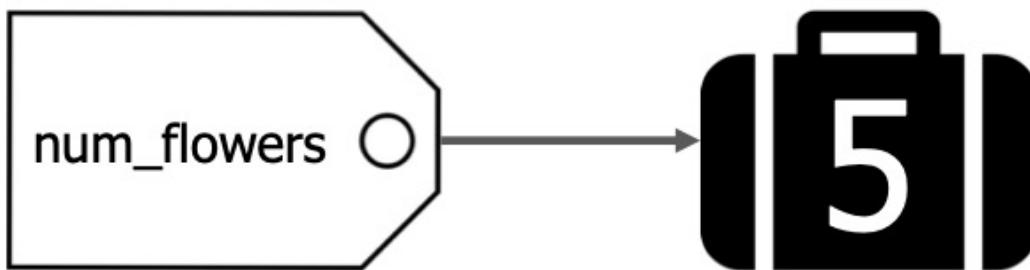
Before



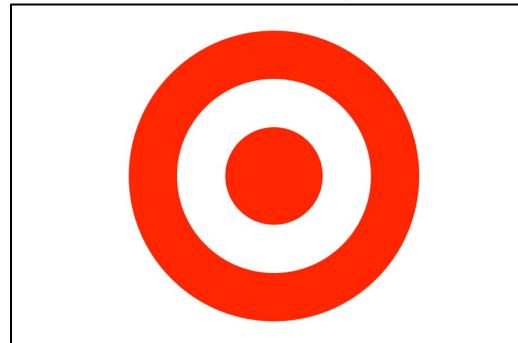
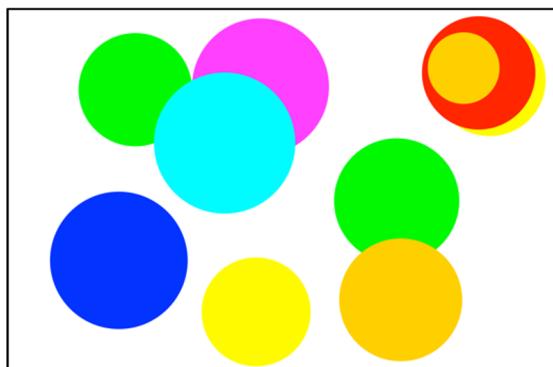
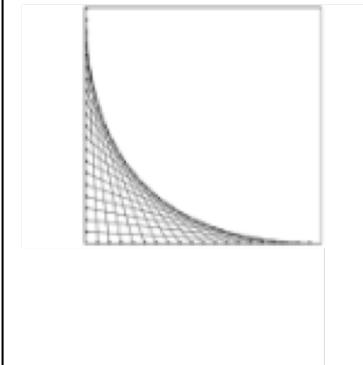
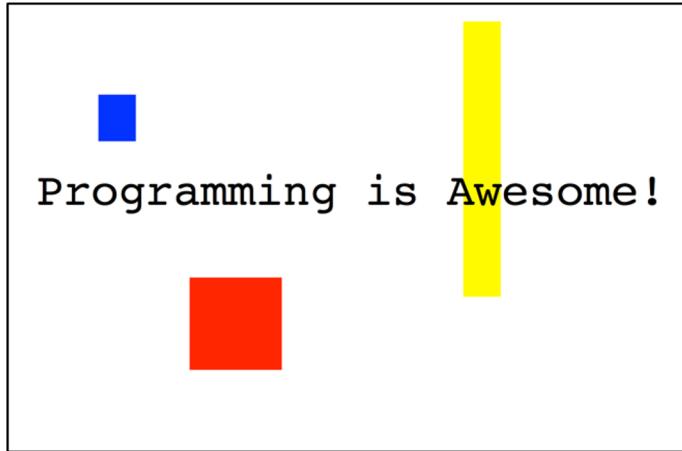
After



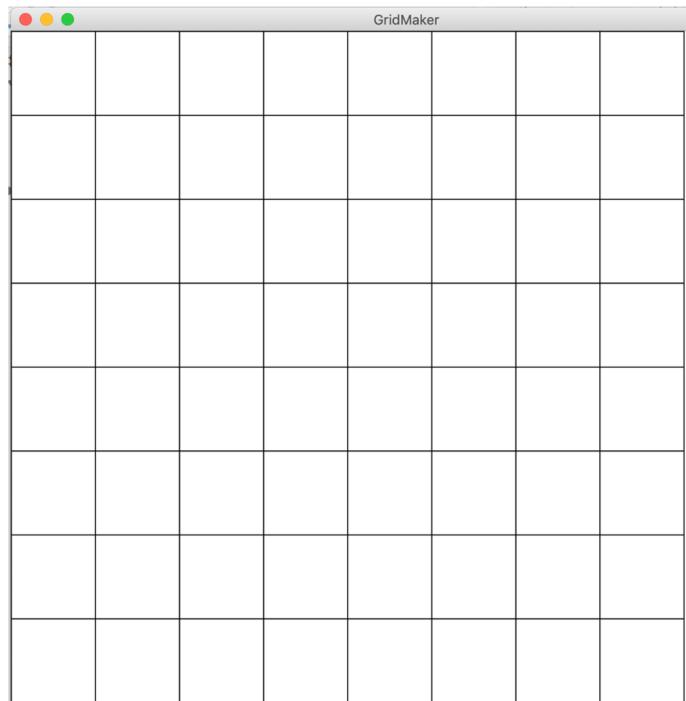
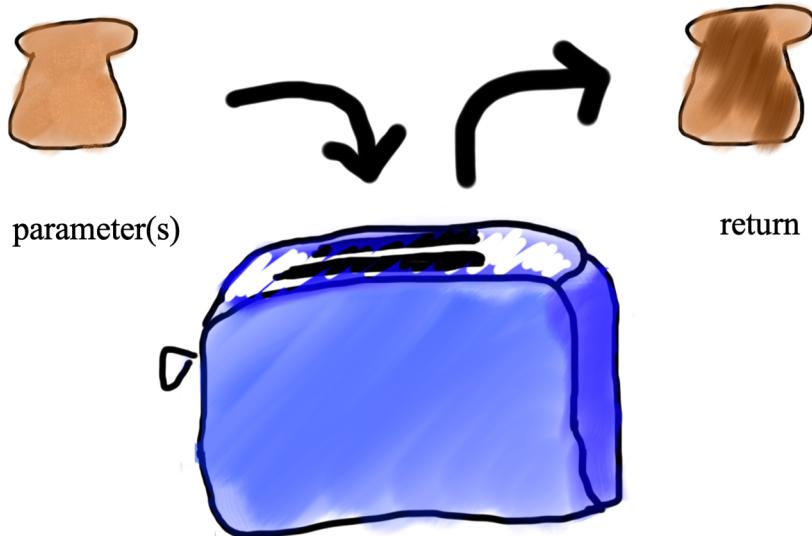
# variables



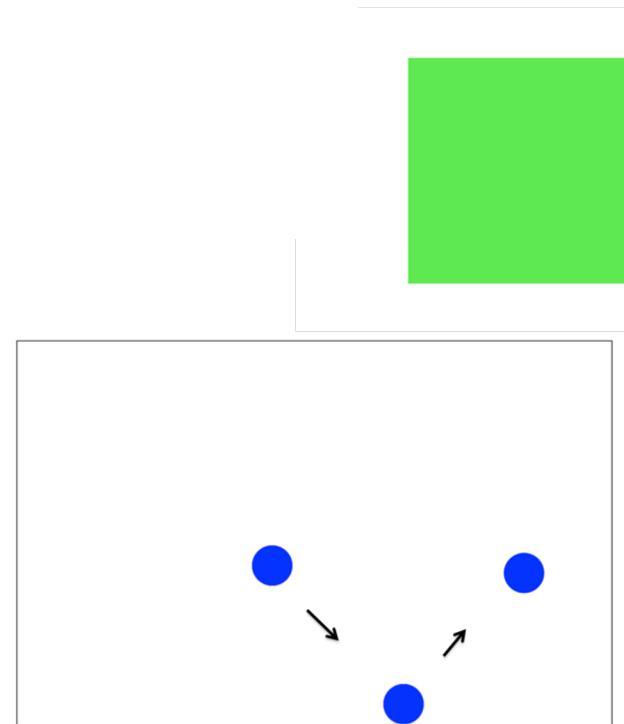
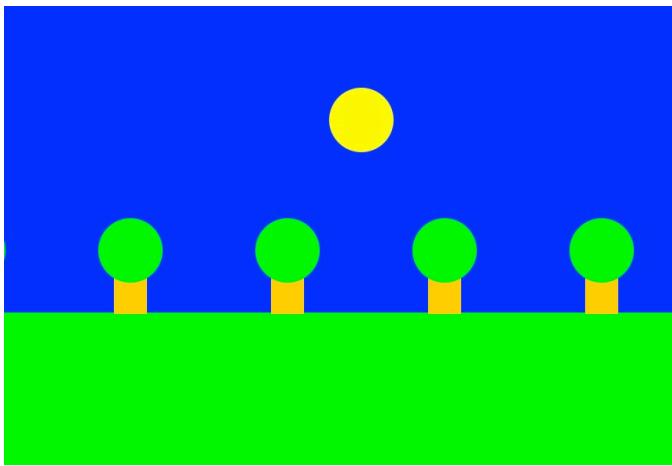
# Graphics



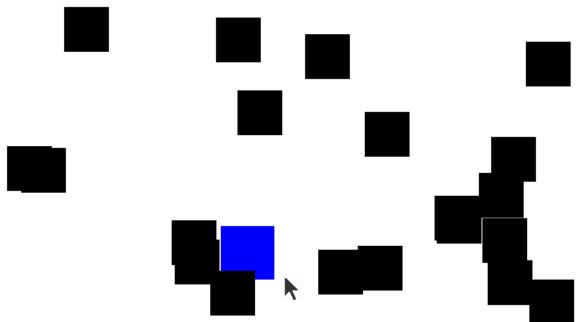
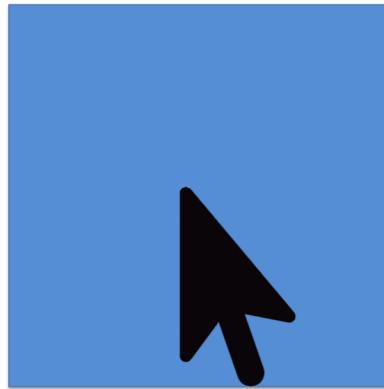
# Functions and Advanced Loops



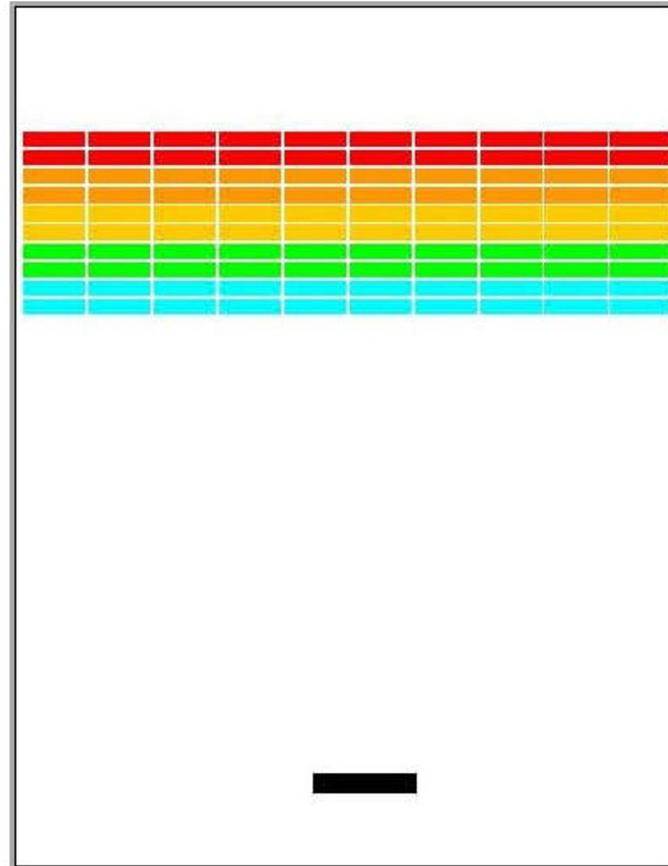
# Animation



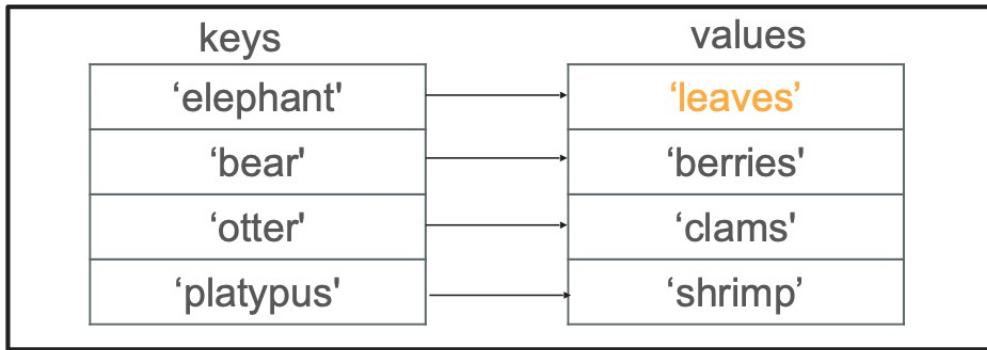
# Lists and The Mouse



# Breakout



# Dictionaries



# Final Project



# Plan For Today

- Where We've Been
- **Where We Can Go Next**
  - Learning Resources
  - Python
- Thank you!

# Self-study

- <https://see.stanford.edu/Course/CS106A> (Java)
- <https://web.stanford.edu/class/archive/cs/cs106a/cs106a.1216/> (Python)
- <https://teachcs4good.org/> (Java)
- <https://online.stanford.edu>
- <https://coursera.org>
- <https://khanacademy.org>
- <https://pluralsight.com>
- <https://docs.python.org/3/>
- <https://turkey21.csbridge.org>

# Self-study

But what is a neural network?

<https://www.youtube.com/watch?v=aircAruvnKk>

The Essence of Calculus

<https://www.youtube.com/watch?v=WUvTyaaNkzM>

# Self-study

Lots of practice!

# useful download links

<https://www.jetbrains.com/pycharm/>

*Demo: Creating a new Python program  
and installing a Python package*

what else is out there?

## HOW STANDARDS PROLIFERATE:

(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)

JavaScript

DOSBatch  
Markup

Tcl-Tk  
SGML

Logic-based  
View

Pike

Logo

Rigal  
Befunge

Blue  
Chippy  
Bigwip

HTML

Frontier  
Occam

Algo60  
G  
m4

ElasticC

Ay

Haskell

Apple

SITUATION:  
THERE ARE  
14 COMPETING  
STANDARDS.

14?! RIDICULOUS!  
WE NEED TO DEVELOP  
ONE UNIVERSAL STANDARD  
THAT COVERS EVERYONE'S  
USE CASES.



YEAH!

SOON:

SITUATION:  
THERE ARE  
15 COMPETING  
STANDARDS.

Ada  
Scripting  
Java  
Multiparadigm  
Clojure  
Fuscated  
Sc  
Lisp  
Mercury  
POP-11  
ADL  
ARB  
Dataflow  
Water HSL  
Sather  
Leeda  
CobolScript  
TADS  
Basic  
HyperCard  
Procedure  
PostScript  
VBScript  
API  
Routin  
blog  
Reflective  
HTML

# Fizzbuzz

Fizzbuzz is a classic coding problem.

In the game Fizz Buzz, players take turns counting up from one. If a player's turn lands on a number that's divisible by 3, she should say "Fizz" instead of the number, and if it lands on a number that's divisible by 5, she should say "Buzz" instead of the number. If the number is both a multiple of 3 and of 5, she should say "Fizzbuzz" instead of the number. A spectator sport, it is not. What it is, however, is an interesting problem in control flow and parameter usage.

# Fizzbuzz

```
1  
2  
Fizz  
4  
Buzz  
Fizz  
7  
8  
Fizz  
Buzz  
11  
Fizz  
13  
14  
Fizzbuzz  
16
```

# Fizzbuzz (Python)

```
1
2 def main():
3     for i in range(100):
4         fizzbuzz(i+1)
5
6
7 def fizzbuzz(i):
8     """
9     Prints the right text for the number i.
10    If the number is divisible by 3, it instead prints "Fizz",
11    if the number is divisible by 5, it instead prints "Buzz",
12    and if it is both, instead it prints "FizzBuzz".
13    """
14    if (i % 3 == 0) && (i % 5 == 0):
15        print("FizzBuzz")
16    elif i % 3 == 0:
17        print("Fizz")
18    elif i % 5 == 0:
19        print("Buzz")
20    else:
21        print(i)
22
```

# Fizzbuzz (C++)

```
1 int main() {
2     for(int i = 1; i <= 100; i++) {
3         fizzbuzz(i);
4     }
5
6     return 0;
7 }
8
9
10 void fizzbuzz(int i) {
11     if ((i % 3 == 0) && (i % 5 == 0)) {
12         cout << "FizzBuzz" << endl;
13     } else if(i % 3 == 0) {
14         cout << "Fizz" << endl;
15     } else if(i % 5 == 0) {
16         cout << "Buzz" << endl;
17     } else {
18         cout << i << endl;
19     }
20 }
21 }
```

# Fizzbuzz (Java)

```
2 class FizzBuzz {
3     public static void main(String[] args) {
4         for(int i = 0; i <= 100; i++) {
5             fizzbuzz(i);
6         }
7     }
8
9     private static void fizzbuzz(int i) {
10        if(i % 15 == 0) {
11            System.out.println("FizzBuzz");
12        }
13        else if(i % 3 == 0) {
14            System.out.println("Fizz");
15        }
16        else if(i % 5 == 0) {
17            System.out.println("Buzz");
18        }
19        else {
20            System.out.println(i);
21        }
22    }
23 }
```

# Fizzbuzz (JavaScript)

```
1
2< function main() {
3     for (var i=1; i <= 100; i++) {
4         fizzbuzz(i);
5     }
6 }
7
8< function fizzbuzz(i) {
9     if ((i % 3 == 0) && (i % 5 == 0))
10        console.log("FizzBuzz");
11    else if (i % 3 == 0)
12        console.log("Fizz");
13    else if (i % 5 == 0)
14        console.log("Buzz");
15    else
16        console.log(i);
17 }
18
```

# Fizzbuzz (Go)

```
2 package main
3 import ("fmt")
4
5 func main() {
6     for i := 1; i <= 100; i++ {
7         fizzbuzz(i)
8     }
9 }
10
11 func fizzbuzz(i int) {
12     if (i % 3 == 0) && (i % 5 == 0) {
13         fmt.Println("FizzBuzz")
14     } else if i % 3 == 0 {
15         fmt.Println("Fizz")
16     } else if i % 5 == 0 {
17         fmt.Println("Buzz")
18     } else {
19         fmt.Println(i)
20     }
21 }
```

# Fizzbuzz (Haskell)

```
2  fizz :: Int -> String
3~ fizz n | n `mod` 15 == 0 = "FizzBuzz"
4~         | n `mod` 3 == 0 = "Fizz"
5~         | n `mod` 5 == 0 = "Buzz"
6~         | otherwise      = show n
7
8 main :: IO()
9 main = mapM_ putStrLn $ map fizz [1..100]
10
```

# Fizzbuzz (><>)

```
1 0voa ~/?=0:\n2 voa      0000'Buzz'~< /\n3 >1+:aa*1+=?; ::5%:{3%:@*?\?/'zzif'0000/\n4 ^oa      n:~~/\n5
```



There are a lot of programming languages!

But they all share the same core concepts



Stay in touch on Ed!

# Joy of Building





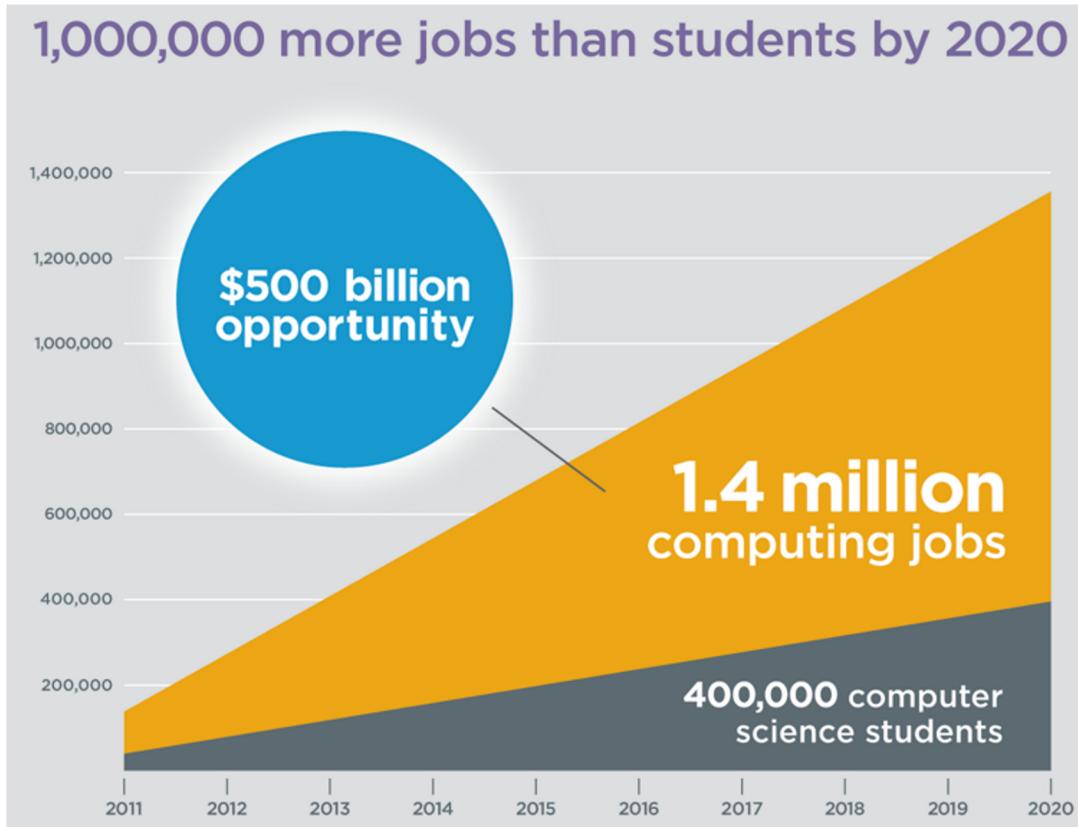
# Closest Thing to Magic



# Now is the time!



# oh and it's useful

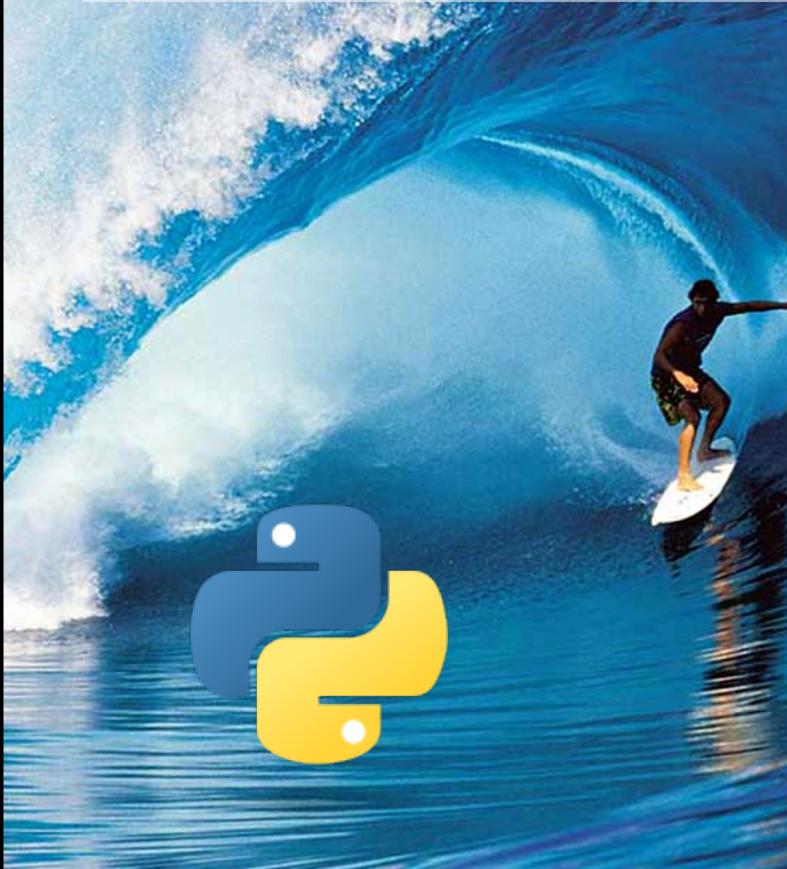


# Everyone is welcome



Tell your friends :-)

# Keep Learning by Doing



```
# good life
while True:
    learn()
    play()
    love()
    time.sleep(8)
```

we hope we excited  
you about learning  
more computer  
science!

You should be  
proud of  
yourselves

