Beyond CS Bridge

Plan For Today

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

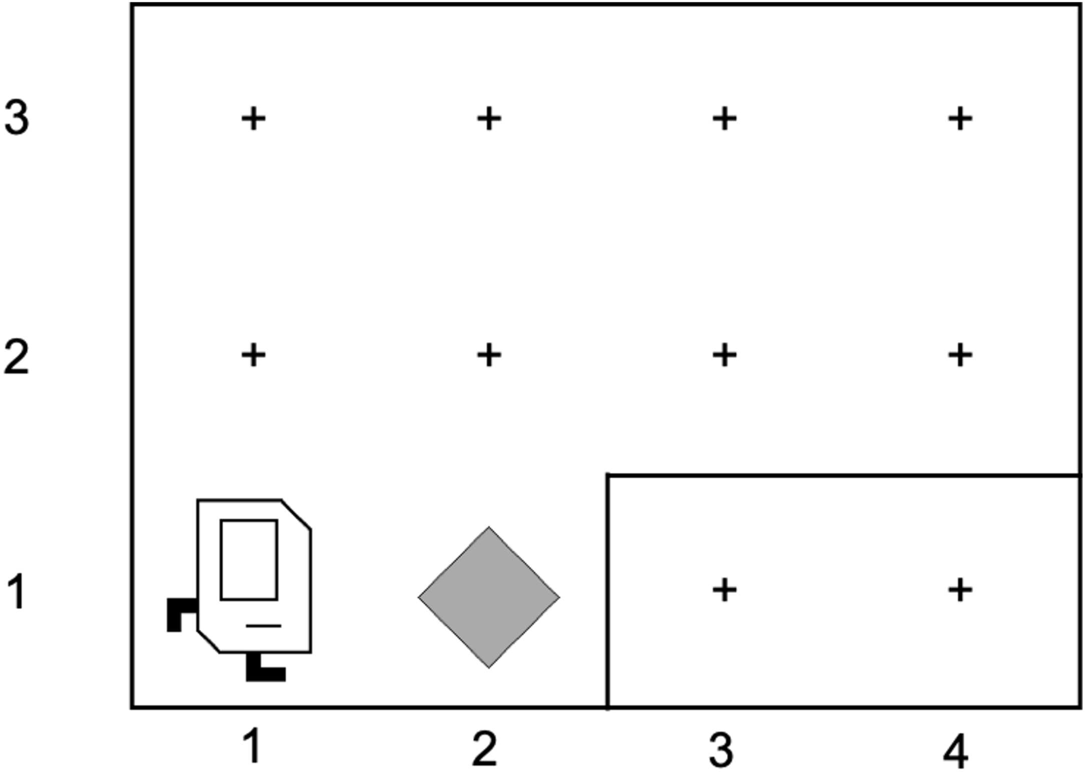
Plan For Today

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

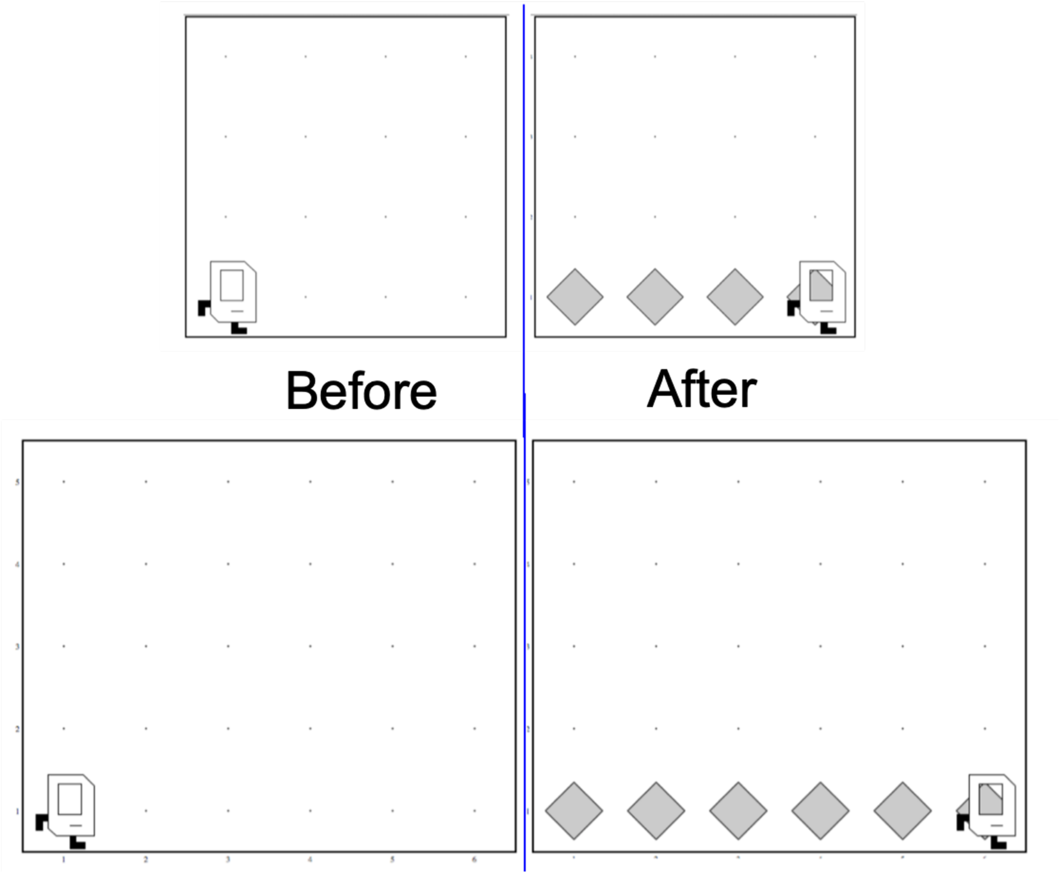
Quick Recap

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 Learining and Computer Vision
17. Music Information Retrieval

First Day

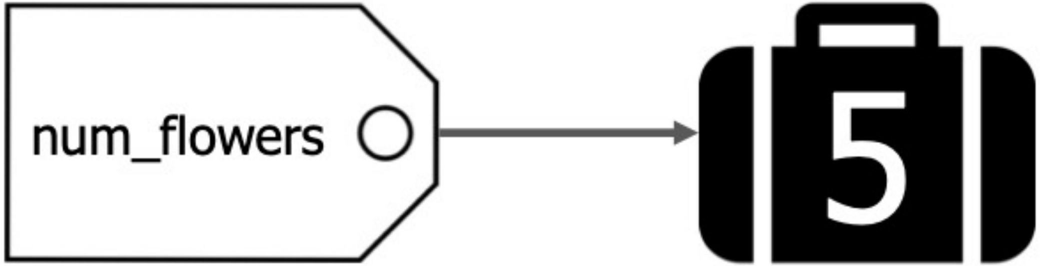


Generalization

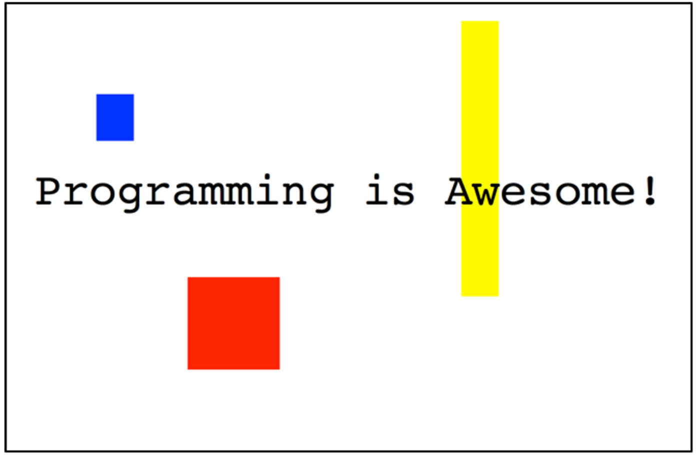

Before

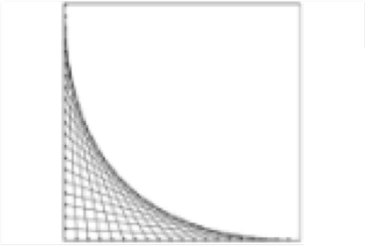
After

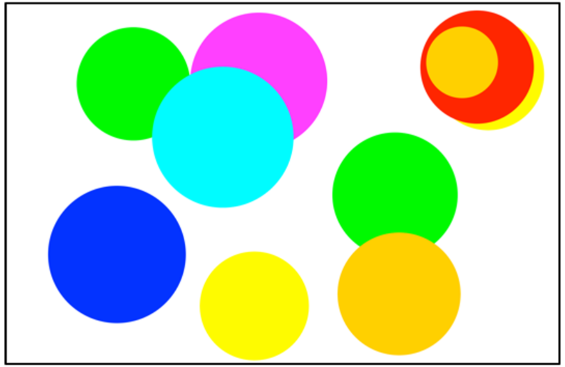
Variables

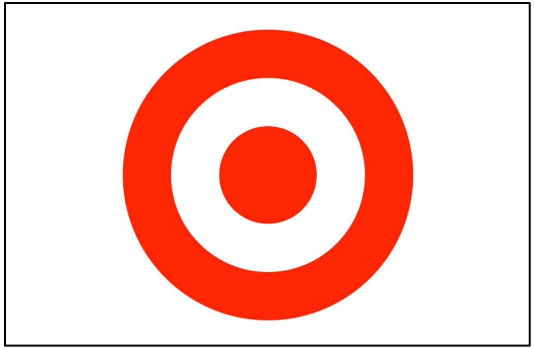


Graphics

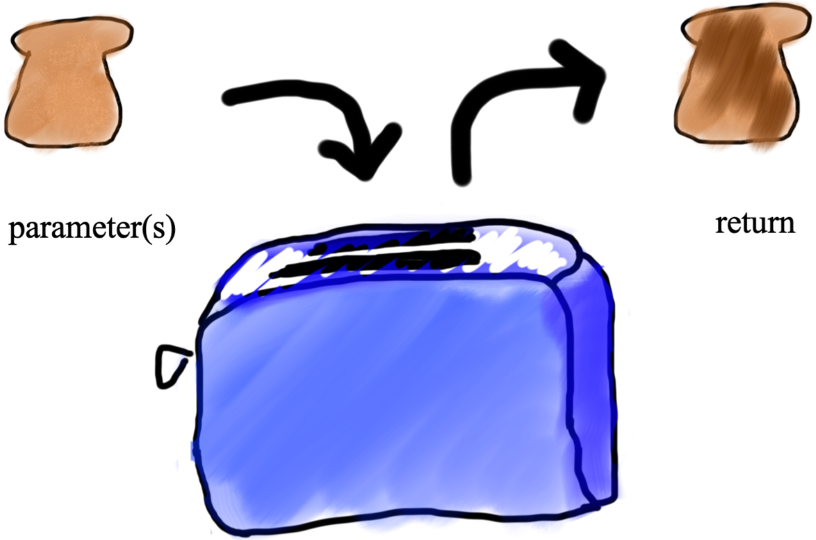






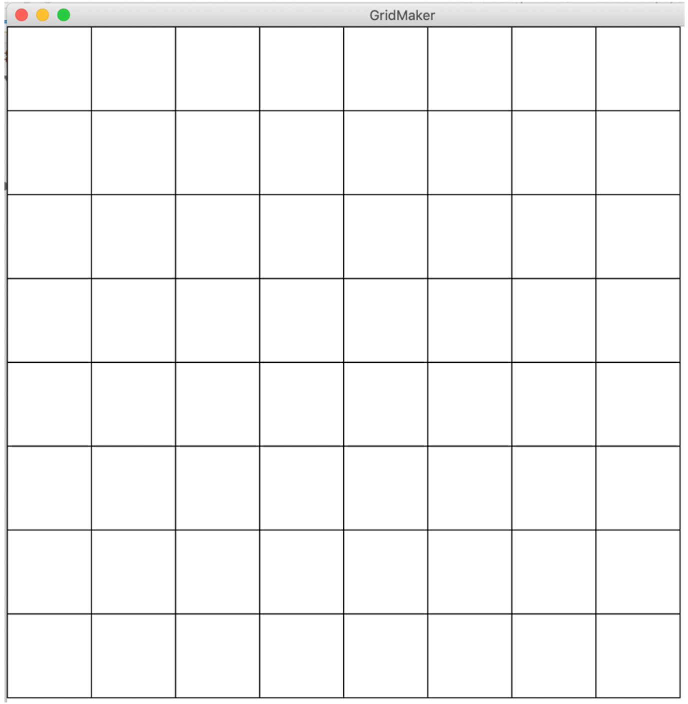


Functions and Advanced Loops

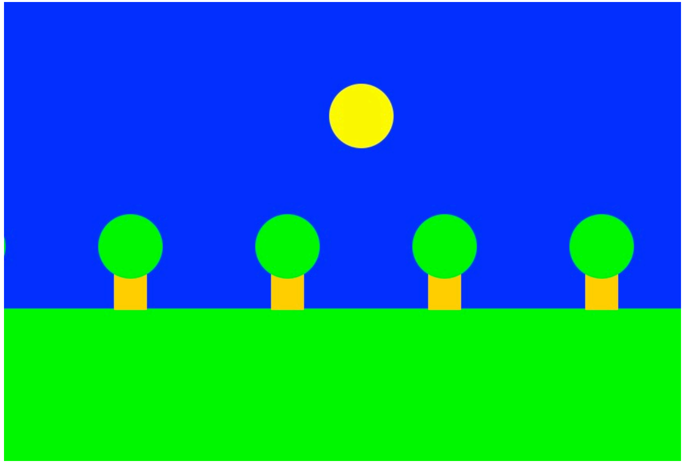


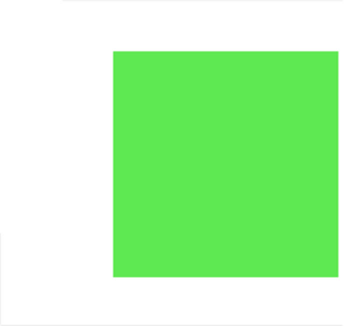
parameter(s)

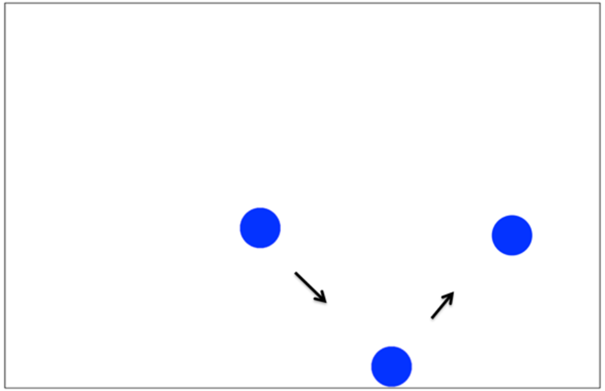
return



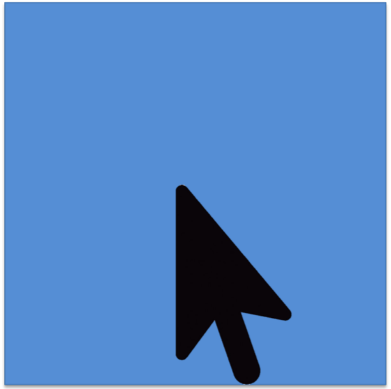
Animation

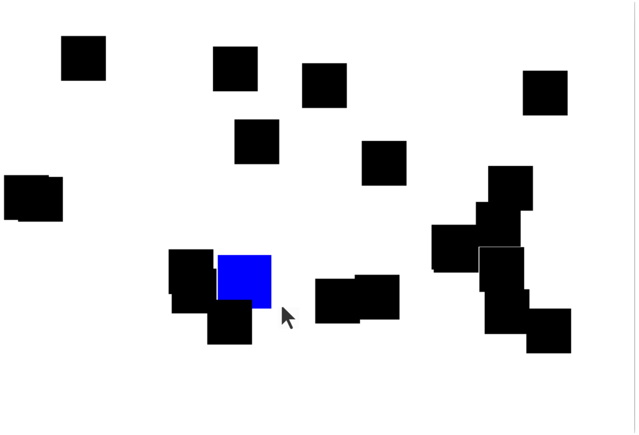




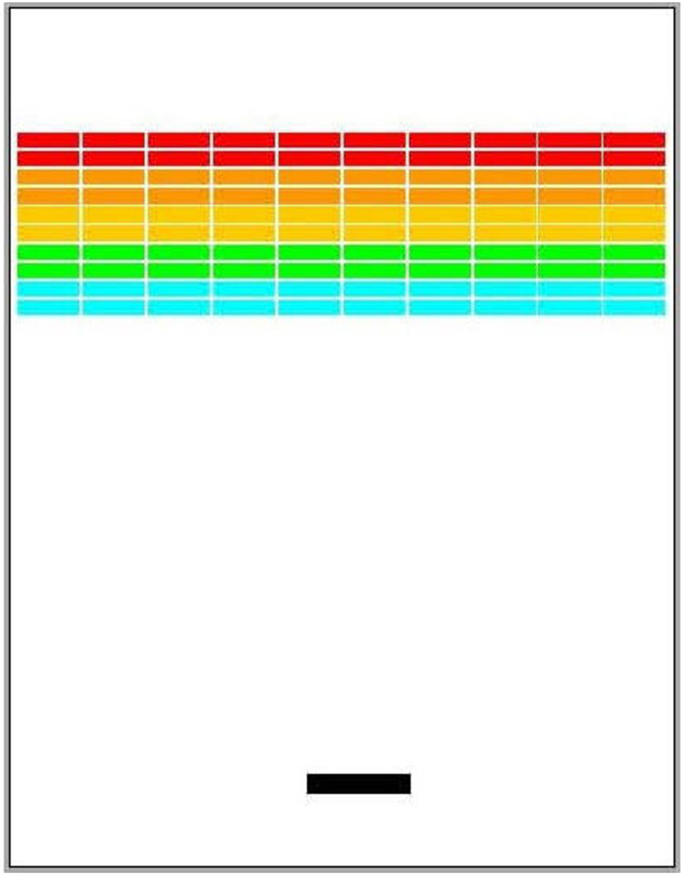


Lists and The Mouse

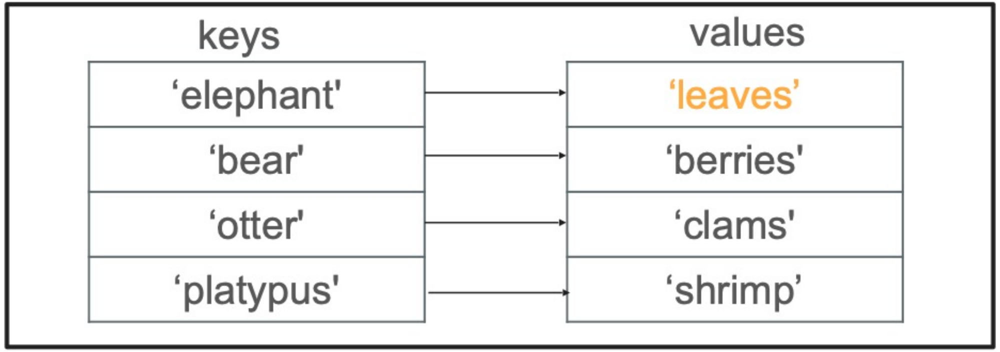




Breakout



Dictionaries



keys	values
‘elephant'	‘leaves’
‘bear'	‘berries'
‘otter'	‘clams'
‘platypus'	‘shrimp’

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

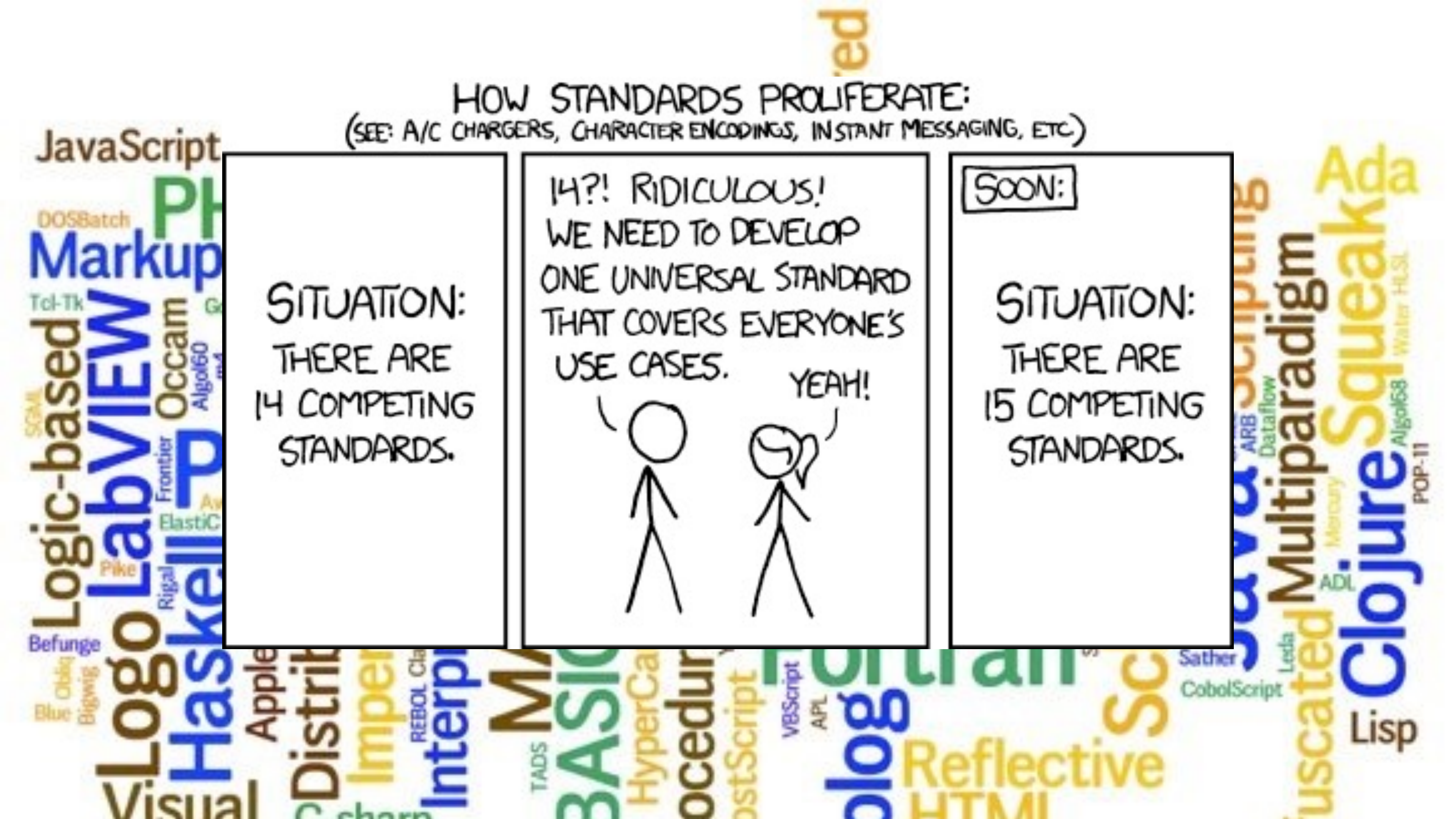
Lots of practice!

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.)
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.
JavaScript
PH
DOSBatch
Markup
Tcl-Tk
Logic-based
LabVIEW
Occam
Algol60
SGML
Frontier
ElasticC
Pike
Rigal
Haskell
Befunge
Logo
Obliq
Bigwig
Blue
Visual
Apple
Distrib
Imper
REBOL
Interp
C sharp
MA
BASIC
TADS
HyperCa
ocedur
ostScript
VBScript
APL
olog
ortran
Reflective
HTML
Sc
Sather
CobolScript
Leda
uscated
Lisp
Clojure
ADL
Mercury
POP-11
Algol68
Water HLSL
Squeak
Ada
Multiparadigm
Dataflow
ARB
Java
ripting
ed

Fizzbuzz

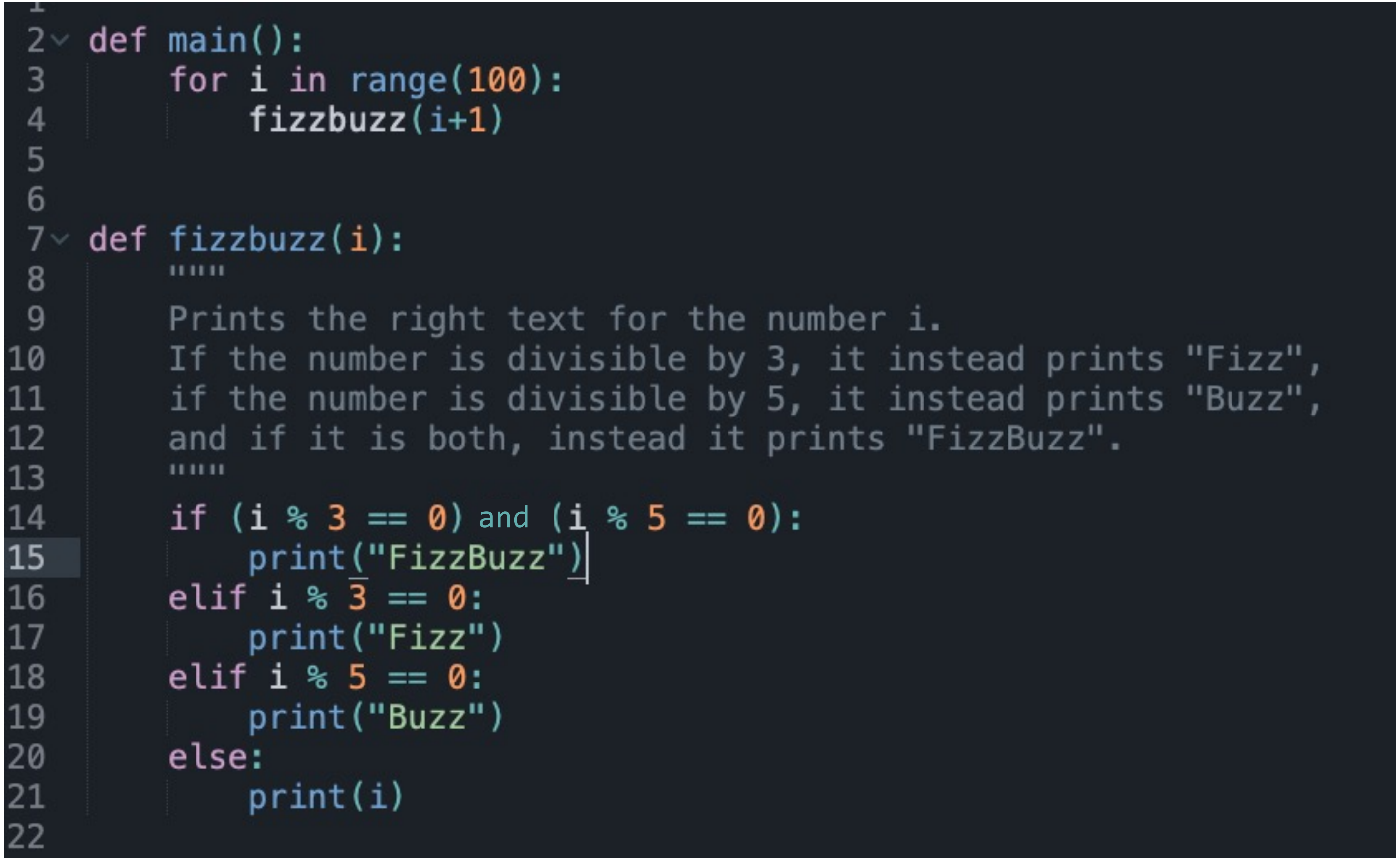
Fizzbuzz is a classic coding problem.

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)

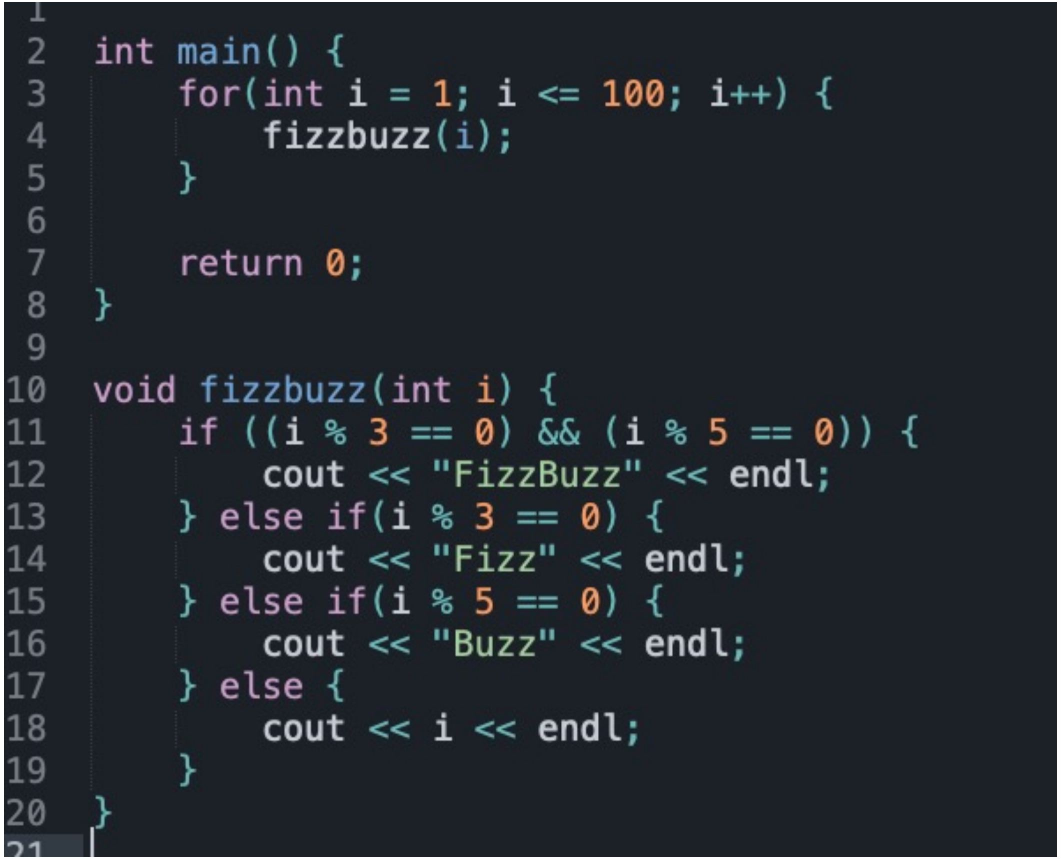


```python
def main():
    for i in range(100):
        fizzbuzz(i+1)


def fizzbuzz(i):
    """
    Prints the right text for the number i.
    If the number is divisible by 3, it instead prints "Fizz",
    if the number is divisible by 5, it instead prints "Buzz",
    and if it is both, instead it prints "FizzBuzz".
    """
    if (i % 3 == 0) and (i % 5 == 0):
        print("FizzBuzz")
    elif i % 3 == 0:
        print("Fizz")
    elif i % 5 == 0:
        print("Buzz")
    else:
        print(i)

```

Fizzbuzz (C++)



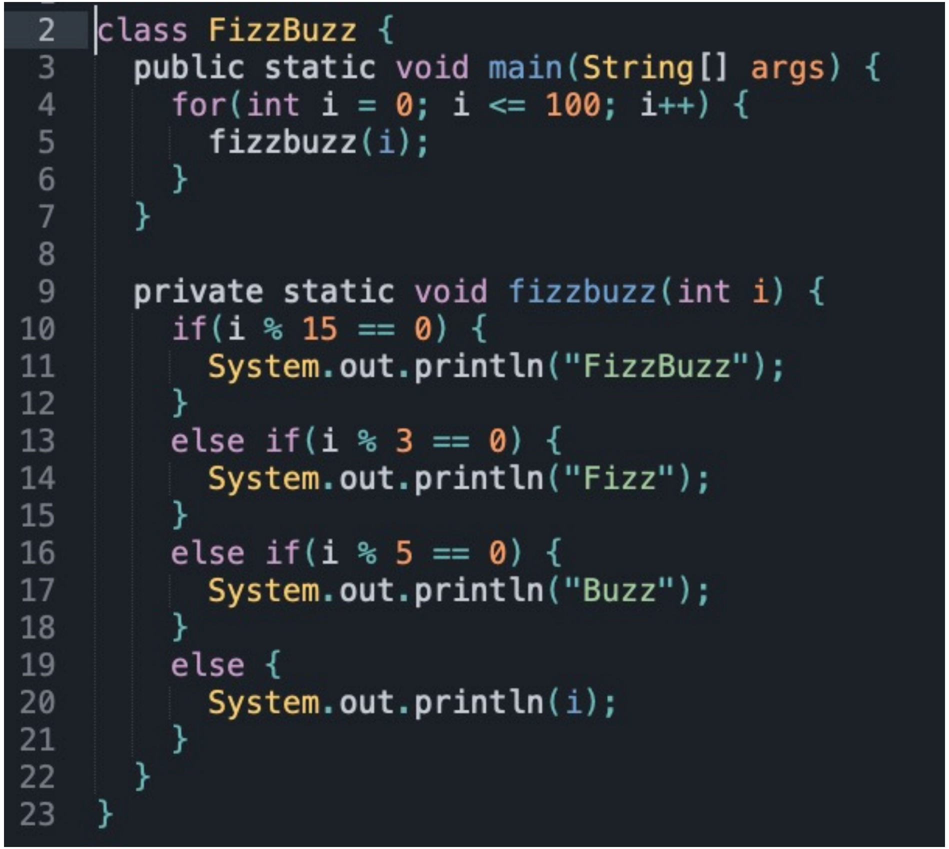
```cpp

int main() {
    for(int i = 1; i <= 100; i++) {
        fizzbuzz(i);
    }

    return 0;
}

void fizzbuzz(int i) {
    if ((i % 3 == 0) && (i % 5 == 0)) {
        cout << "FizzBuzz" << endl;
    } else if(i % 3 == 0) {
        cout << "Fizz" << endl;
    } else if(i % 5 == 0) {
        cout << "Buzz" << endl;
    } else {
        cout << i << endl;
    }
}
```

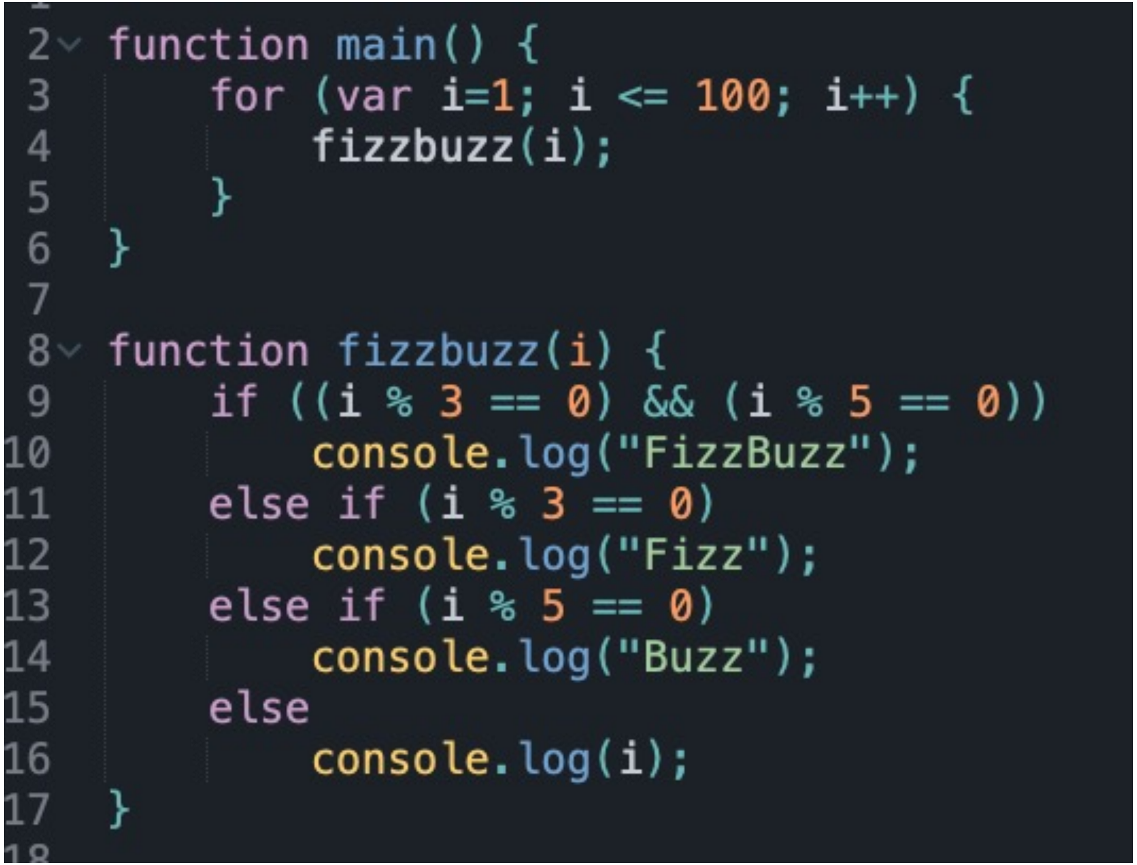
Fizzbuzz (Java)



```java
class FizzBuzz {
  public static void main(String[] args) {
    for(int i = 0; i <= 100; i++) {
      fizzbuzz(i);
    }
  }

  private static void fizzbuzz(int i) {
    if(i % 15 == 0) {
      System.out.println("FizzBuzz");
    }
    else if(i % 3 == 0) {
      System.out.println("Fizz");
    }
    else if(i % 5 == 0) {
      System.out.println("Buzz");
    }
    else {
      System.out.println(i);
    }
  }
}
```

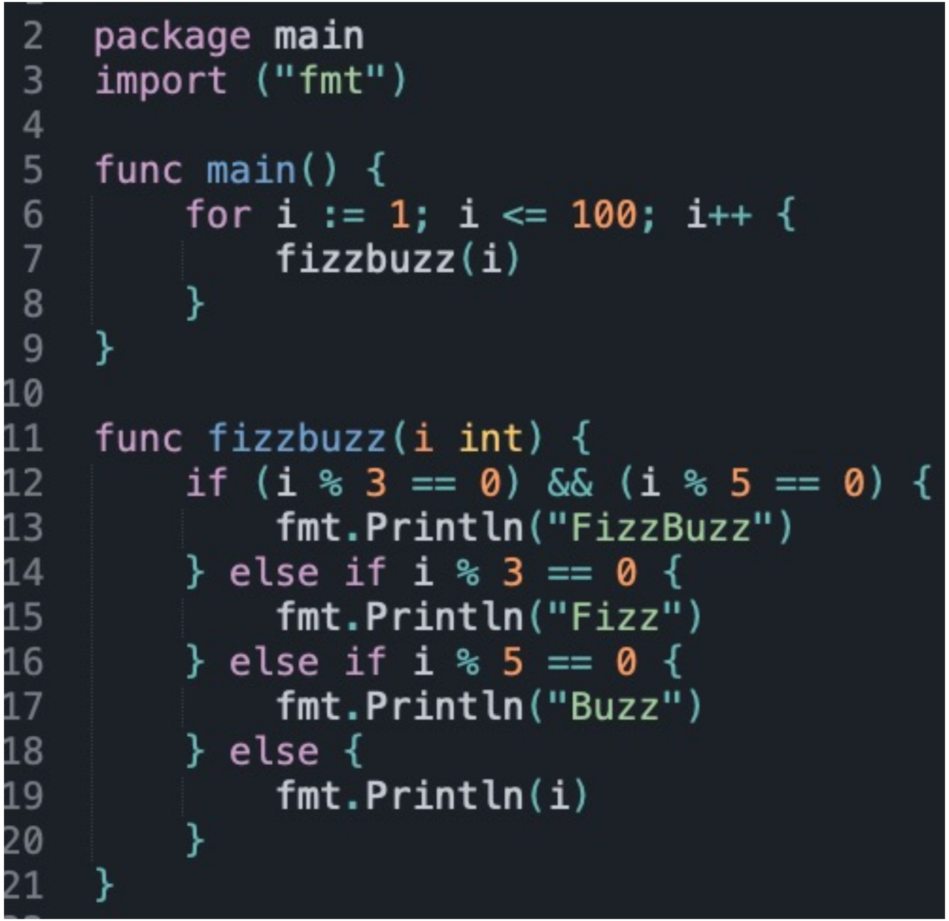
Fizzbuzz (JavaScript)



```
function main() {
    for (var i=1; i <= 100; i++) {
        fizzbuzz(i);
    }
}

function fizzbuzz(i) {
    if ((i % 3 == 0) && (i % 5 == 0))
        console.log("FizzBuzz");
    else if (i % 3 == 0)
        console.log("Fizz");
    else if (i % 5 == 0)
        console.log("Buzz");
    else
        console.log(i);
}
```

Fizzbuzz (Go)

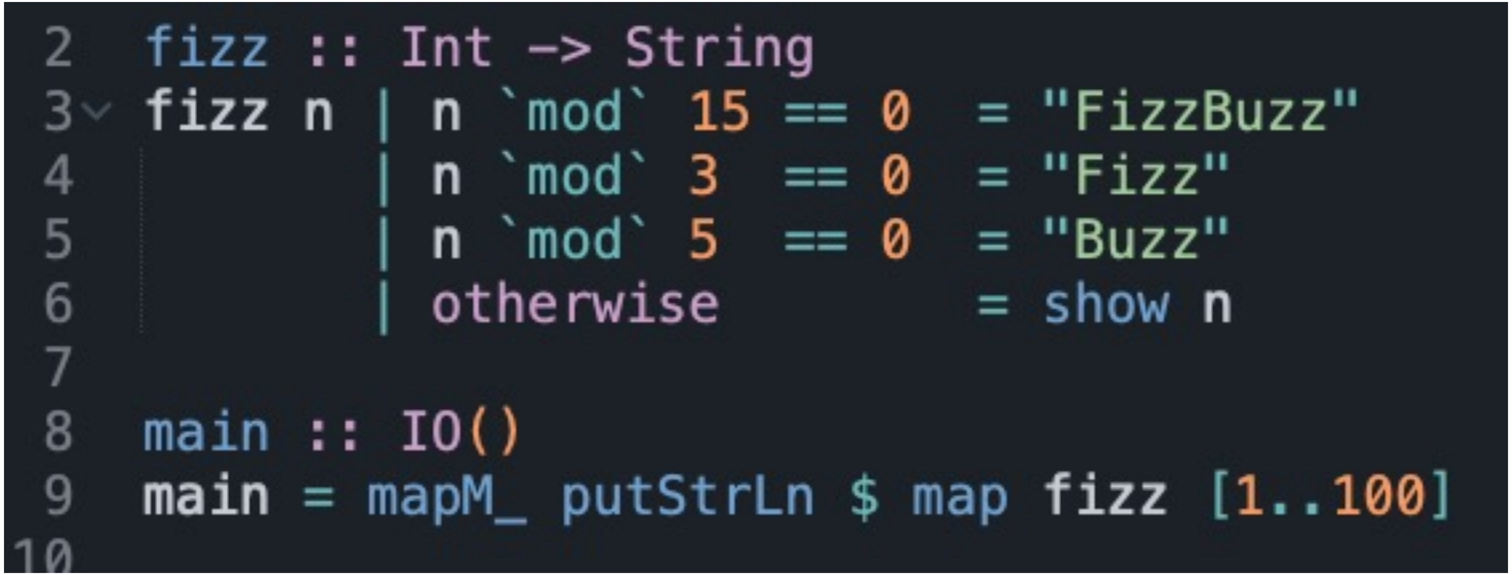


```go
package main
import ("fmt")

func main() {
    for i := 1; i <= 100; i++ {
        fizzbuzz(i)
    }
}

func fizzbuzz(i int) {
    if (i % 3 == 0) && (i % 5 == 0) {
        fmt.Println("FizzBuzz")
    } else if i % 3 == 0 {
        fmt.Println("Fizz")
    } else if i % 5 == 0 {
        fmt.Println("Buzz")
    } else {
        fmt.Println(i)
    }
}
```

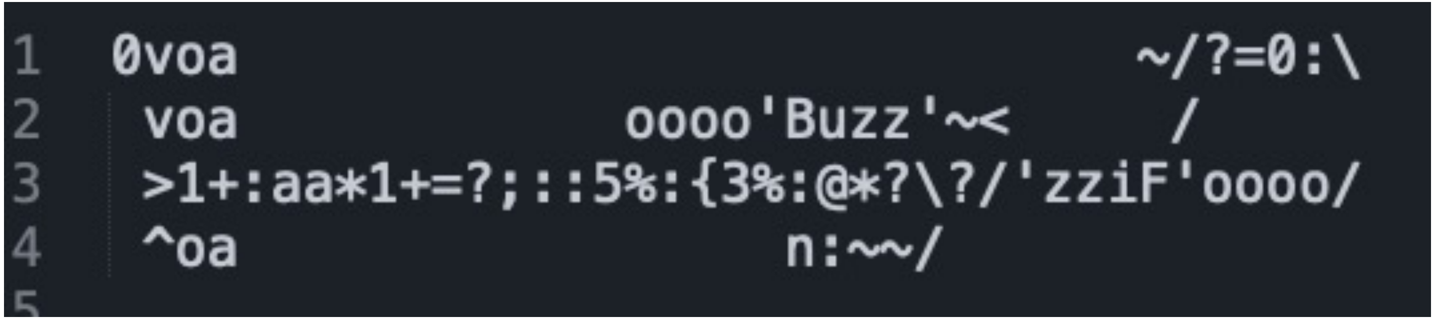
Fizzbuzz (Haskell)



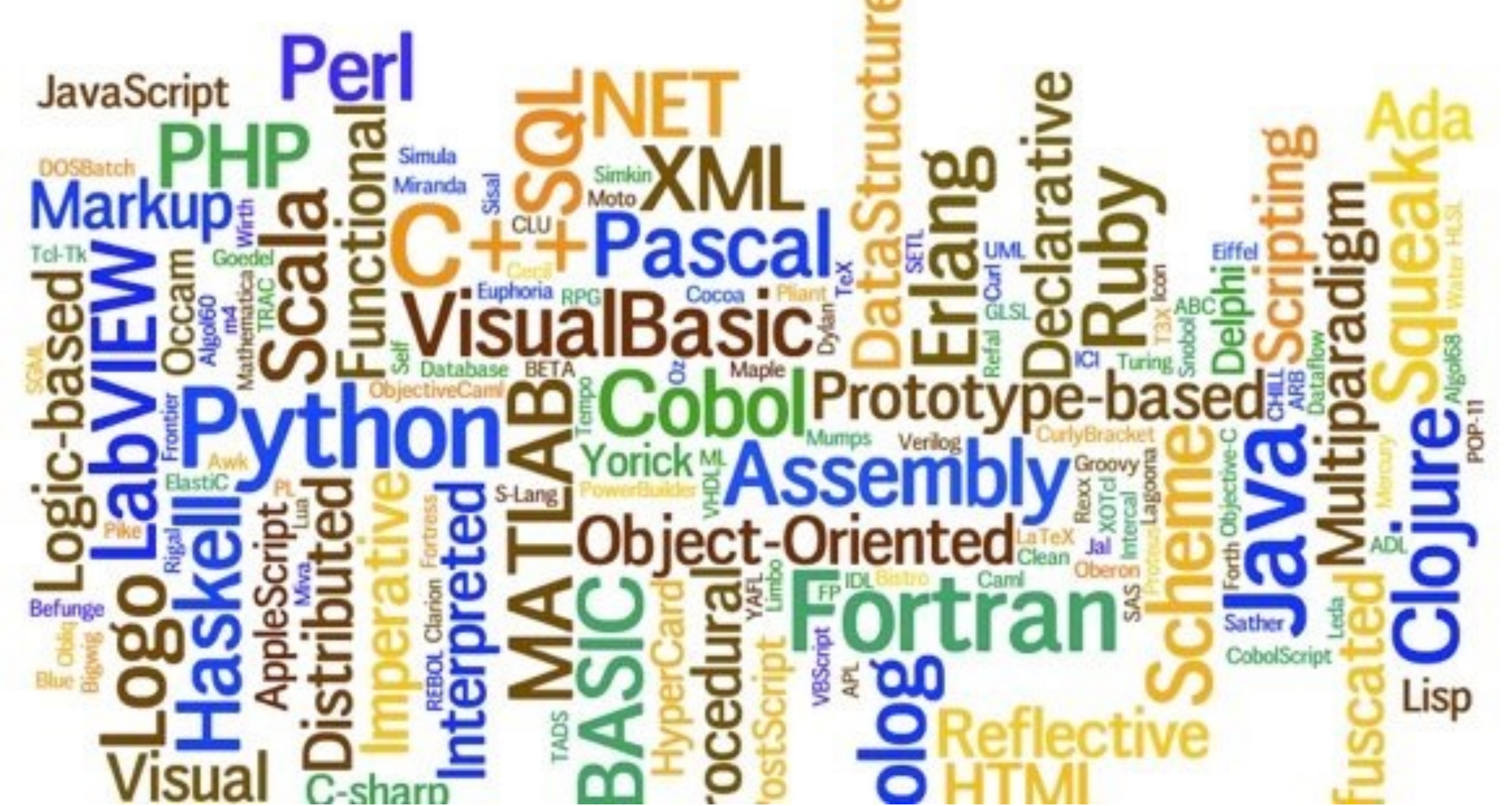
```haskell
fizz :: Int -> String
fizz n | n `mod` 15 == 0  = "FizzBuzz"
       | n `mod` 3  == 0  = "Fizz"
       | n `mod` 5  == 0  = "Buzz"
       | otherwise        = show n

main :: IO()
main = mapM_ putStrLn $ map fizz [1..100]
```

Fizzbuzz (><>)



```
0voa                            ~/?=0:\
 voa            oooo'Buzz'~<     /
 >1+:aa*1+=?;::5%:{3%:@*?\?/'zziF'oooo/
 ^oa                 n:~~/
```


JavaScript
Perl
PHP
DOSBatch
Markup
Simula
Miranda
Sisal
SQL
NET
XML
Simkin
Moto
CLU
C++
Pascal
DataStructure
Erlang
Declarative
Ruby
Ada
Scripting
Multiparadigm
Squeak
Water
HLSL
Tcl-Tk
Goedel
Wirth
Scala
Functional
Cecil
Euphoria
RPG
Cocoa
Pliant
TeX
SETL
UML
Eiffel
Icon
Logic-based
LabVIEW
Occam
Algol60
m4
Mathematica
TRAC
VisualBasic
Dylan
Curl
GLSL
ABC
T3X
Delphi
Snobol
SGML
Self
Database
BETA
Oz
Maple
ICI
Turing
Refal
Dataflow
ARB
CHILL
Algol68
ObjectiveCaml
Python
MATLAB
Cobol
Prototype-based
Tempo
Frontier
Awk
Mumps
Verilog
CurlyBracket
POP-11
Mercury
Clojure
Java
Scheme
Yorick
ML
Assembly
Groovy
ElasticC
PL
PowerBuilder
S-Lang
VHDL
Rexx
XOTcl
Lagoona
Intercal
Objective-C
Pike
Lua
Fortress
Object-Oriented
LaTeX
Jal
Clean
ADL
Forth
Rigal
Haskell
AppleScript
Distributed
Imperative
Interpreted
Oberon
Befunge
Miva
Logo
YAFL
Limbo
IDL
Bistro
Caml
FP
Fortran
SAS
Sather
Leda
Obliq
Bigwig
Blue
REBOL
Clarion
BASIC
HyperCard
rocedural
ostScript
VBScript
APL
olog
CobolScript
fuscated
Lisp
Reflective
HTML
TADS
Visual
C-sharp

There are a lot of programming languages!

But they all share the same core concepts

Stay in touch on Ed!

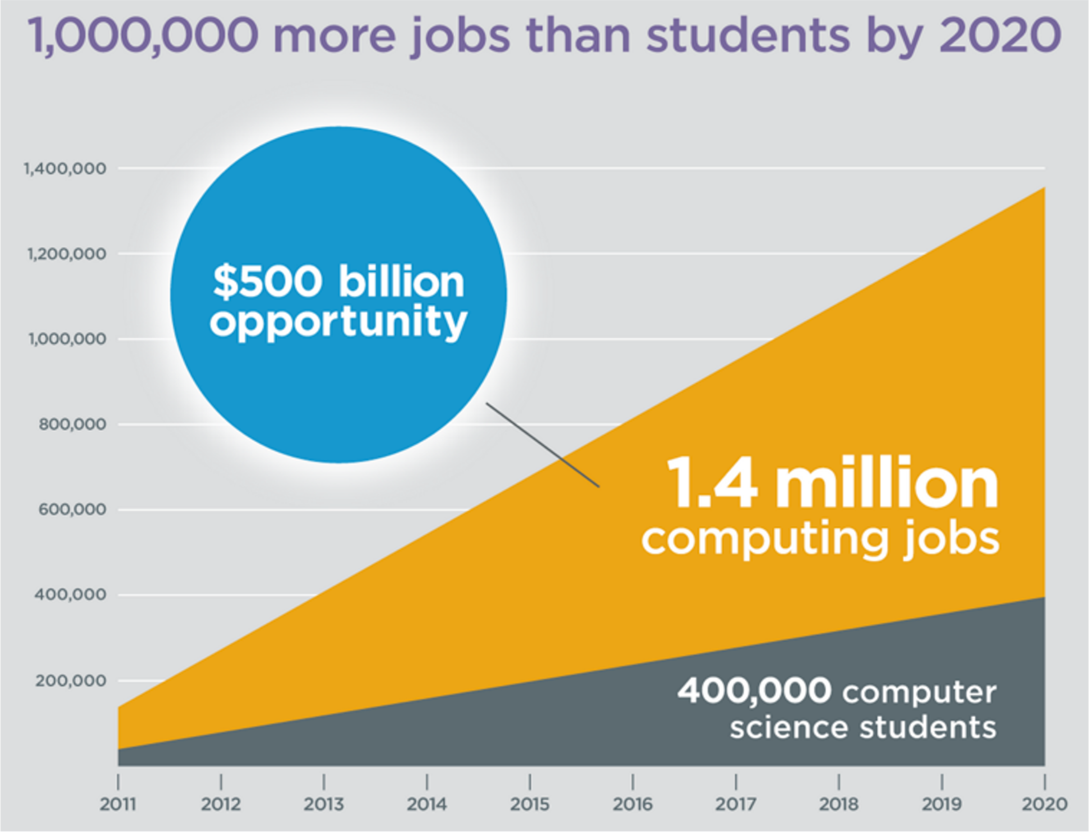
Joy of Building

NINTENDO DS
ANGRY BIRDS
#1 APP OF ALL TIME
Walt Disney Pictures
presents
TOY
STORY
Kyrgyzstan
827
203
257
10
19
185
45
36
1079
154
39
Issyk Kul
Biosphere
Reserve
Almaty
Kapchagay
Taraz
Shymkent
Aksu
Kizilsu
Kashgar
Tumushuke

Closest Thing to Magic

Now is the time!

oh and it's useful

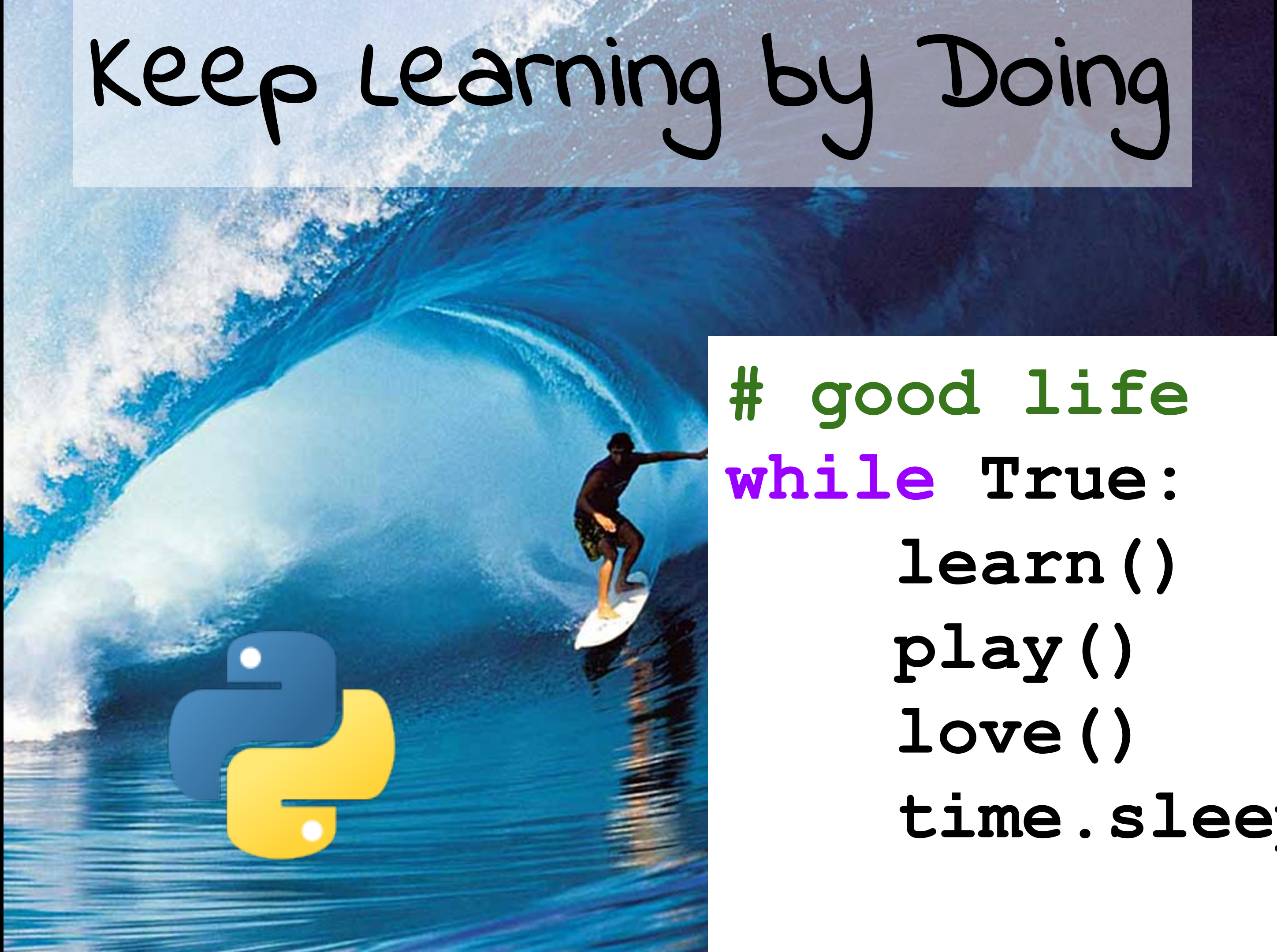


Code.org

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