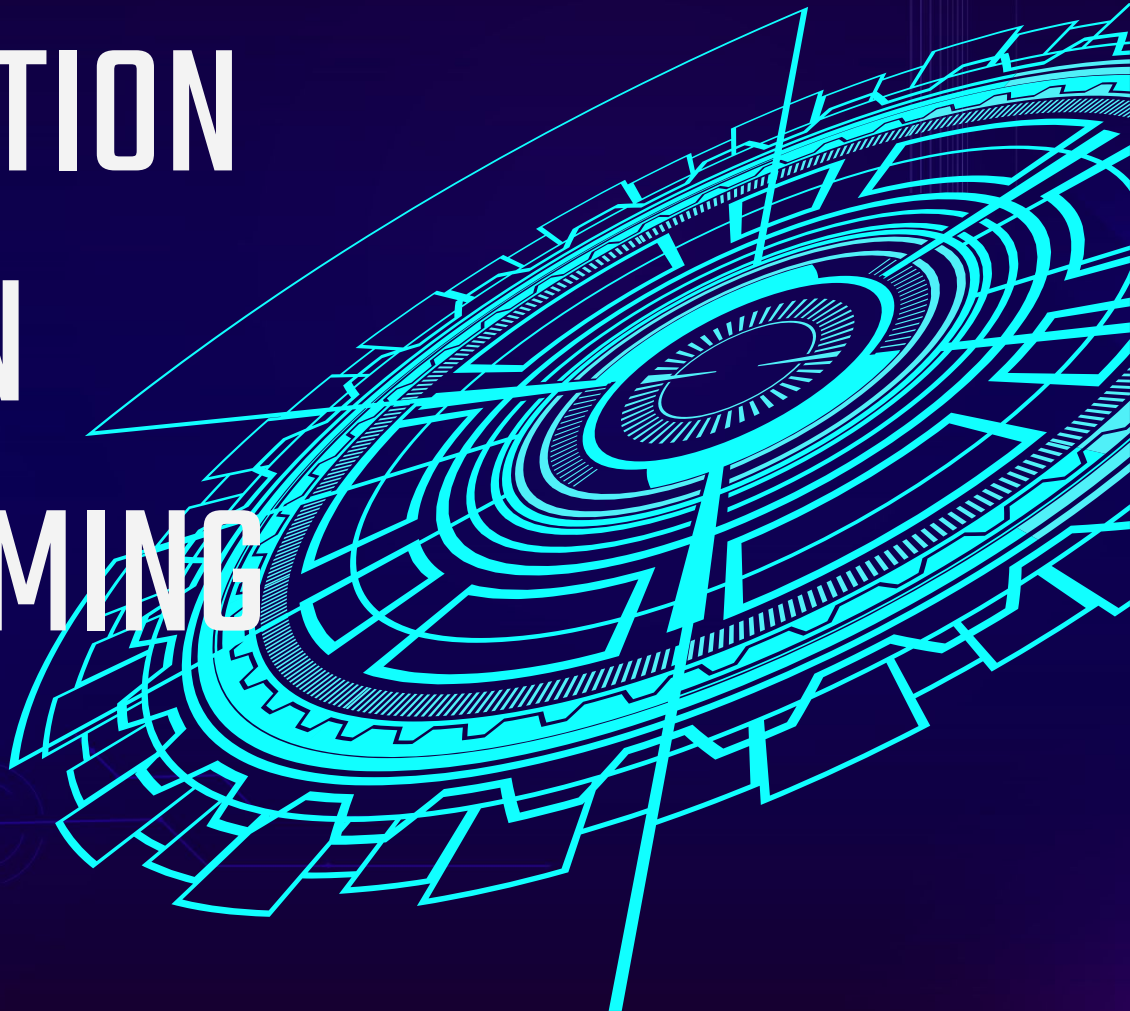


# INTRODUCTION TO PYTHON PROGRAMMING

WEEK  
1



# A LITTLE ABOUT ME...

I am a sophomore at Bellaire High School in Houston TX. I love photography and videography, and I have a cat named Olive.

Tell me about yourself!

Where are you from and what's your favorite thing to do?





# TABLE OF CONTENTS

01

## INTRODUCTION TO PROGRAMMING

Let's dive deeper into  
the world of  
Programming!

02


## FUNDAMENTAL TYPES

What are the building  
blocks of programming?

03

## BASIC ARITHMETIC

Math... but with code!



# INTRODUCTION TO PROGRAMMING

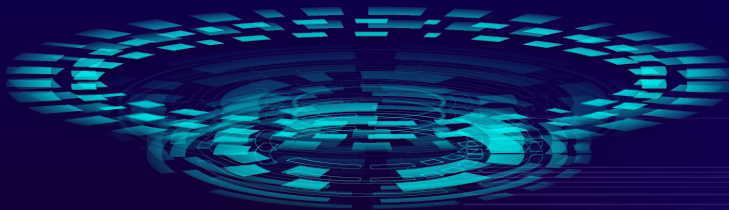
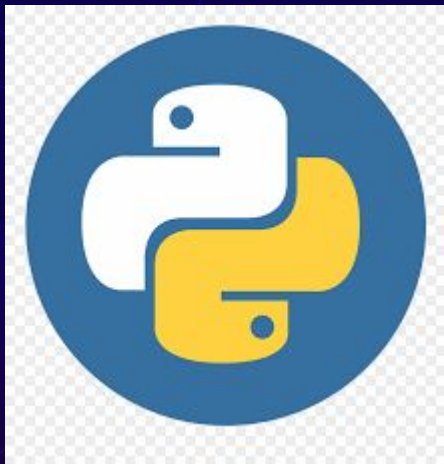
So, what is coding?

We hear it all the time, and it's quite simple.

Code is instructions for a computer to understand and perform.

But, there are differences in good code and bad code.

Good code is human-readable; it's easy for humans to read and be able to comprehend. It's reusable! We never want to repeat an action multiple times. It's accurate, we want to make sure the code works as intended for all situations





Code can be written in many languages!

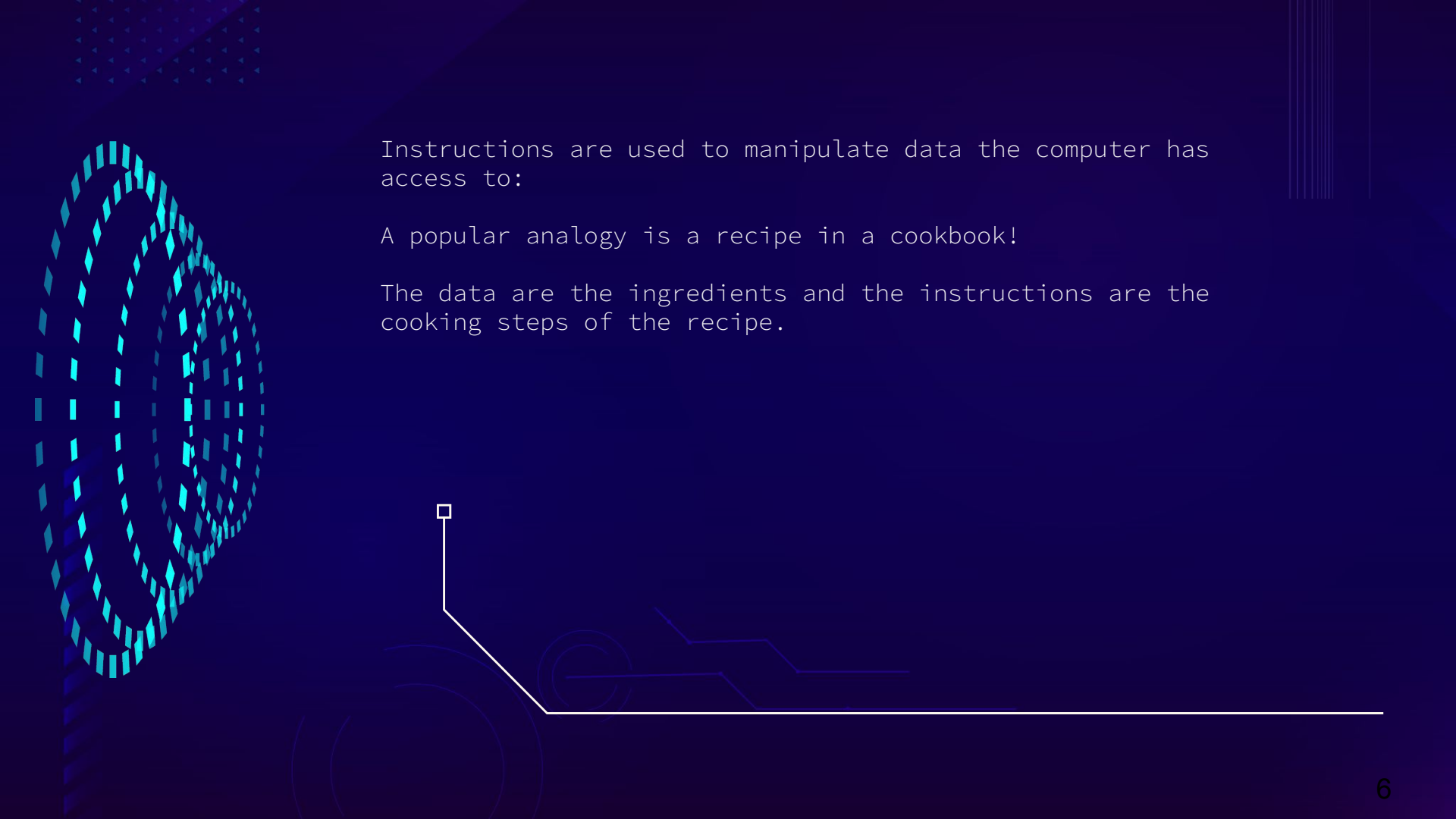
Some examples are: C, C++, C#, Java, and Javascript.

Each language has its own features that make them useful for different specific problems.

In this class, we will be learning **python**, which is one of the biggest and fastest growing languages in the programming industry!

It's relatively simple to learn, and can help immensely with learning other languages later on.

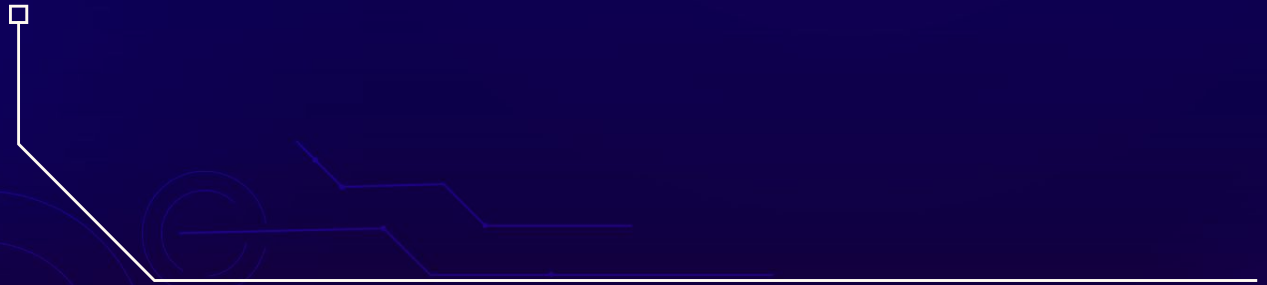




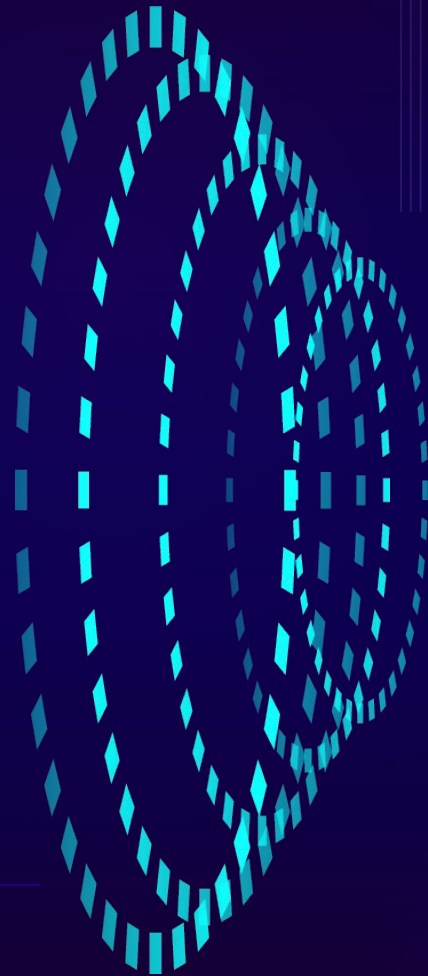
Instructions are used to manipulate data the computer has access to:

A popular analogy is a recipe in a cookbook!

The data are the ingredients and the instructions are the cooking steps of the recipe.



# FUNDAMENTAL TYPES





# WHAT ARE THEY?

Integers (int), Floating point numbers (float),  
Strings (str), Booleans (bool)

Building blocks of a program

Can be combined to create more complex ways of  
storing data or information in your programs



# INTEGERS

Data type that stores an integer value (just like math!)

Cannot store decimal values

Syntax is `int`



# FLOATING POINT NUMBERS

Used to store decimals

Examples: -4.2, 0.0, 100.4923

Used to store fractional values too

Syntax is float

# STRINGS

Stores text values

Denoted by placing value of the string in single or double quotation marks (' , " )

Examples: "hello, world" 'strings are so cool'

If your word contains an apostrophe or quotation marks, you have to use the vice versa, and make sure you don't forget it!

# BOOLEANS

Stores 2 states, true or false

We will discuss booleans more when we cover conditional statements, but for now, remember that any boolean is either equivalent to true or false

# WHAT ARE VARIABLES?

Used to store data as a value that code can refer back to at any time and instruction needs to read it or change it

It avoids repeating value, so that we can reuse it

It also gives a name that is clear for a human to understand

Examples: `x=2 y=12, smoothie = "mine"`

# HOW TO USE THEM?

Variables are assigned using the assignment operator, which in Python is (=)

Example: `x=2`  
variable name is `x`  
variable data is `2`  
variable type is `int`

Example: `name= 'alex'`  
variable name = `name`  
variable data = `'alex'`  
variable type = `str`

# PRACTICE!

Let's do some introductions in Python!

Go to <https://py3.codeskulptor.org> and follow along!



# BASIC ARITHMETIC!

- Addition
- Subtraction
- Multiplication
- Division
- Exponentiation

Integer Division (`//`) is different in Python. It divides WHOLE numbers. This is called “floor division”.

$2/4$  is 0.5, but in Python, it would be 0. Similarly,  $9/4$  is 2.25, but in Python, it would be 2.

Don't fret! Python also has regular division (`/`), which works like a regular calculator.

# STRING CONCATENATION

You can combine strings!

Let's go to `pyskulptor` to do some examples.

Keep in mind that you can not combine integers or any other data types with each other. If you want to, cast the integer as a string by using a `str()` function.

# FUN TRICK: REPEATING!

Something fun you can do is that you can repeat strings!

A multiplication operator can repeat a string given how many times you want.

IDLE example:

```
x= 'hello'  
y=x*5  
print(y)
```