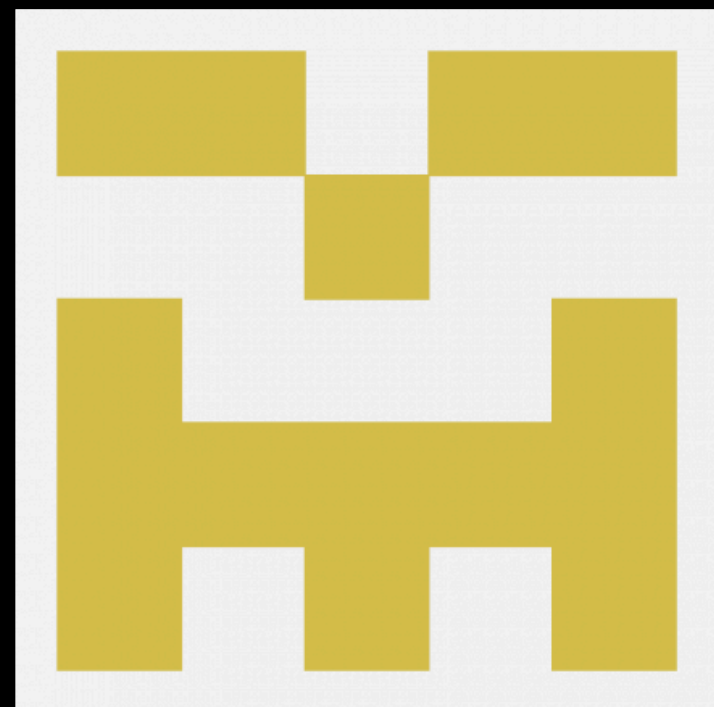# Python Workshop Series 2020 Spring

Session 01: Introduction to Programming in Python

29th Jan 2020

Huang He (Mark)

Github / Telegram: @MarkHershey
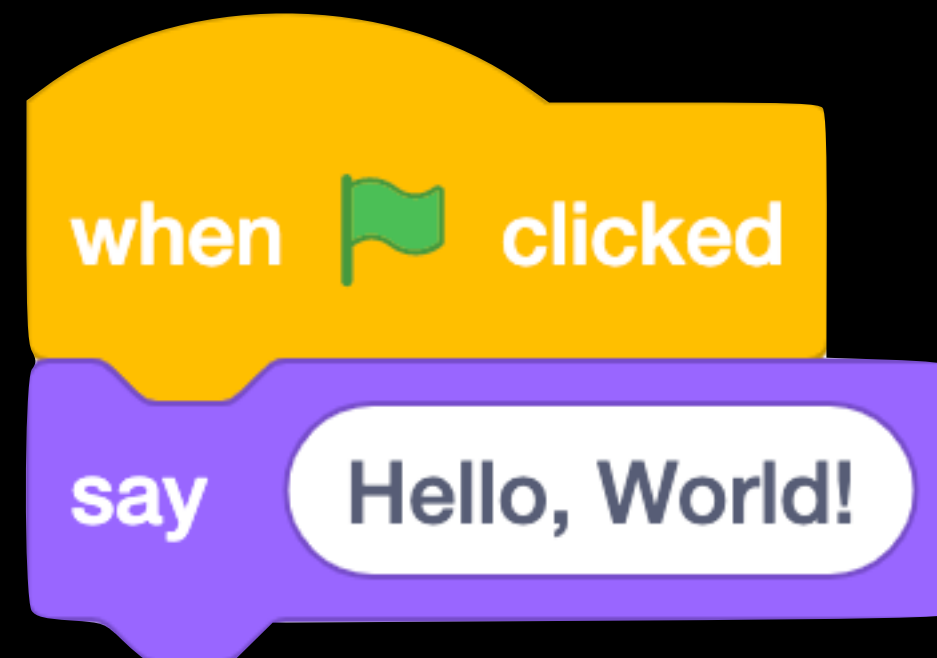
https://www.linkedin.com/in/huanghe97/

Inspired by David J. Malan

01

Computational Thinking

Scratch

What is programming?

input → [ ] → output

Program / Code / Software / Algo
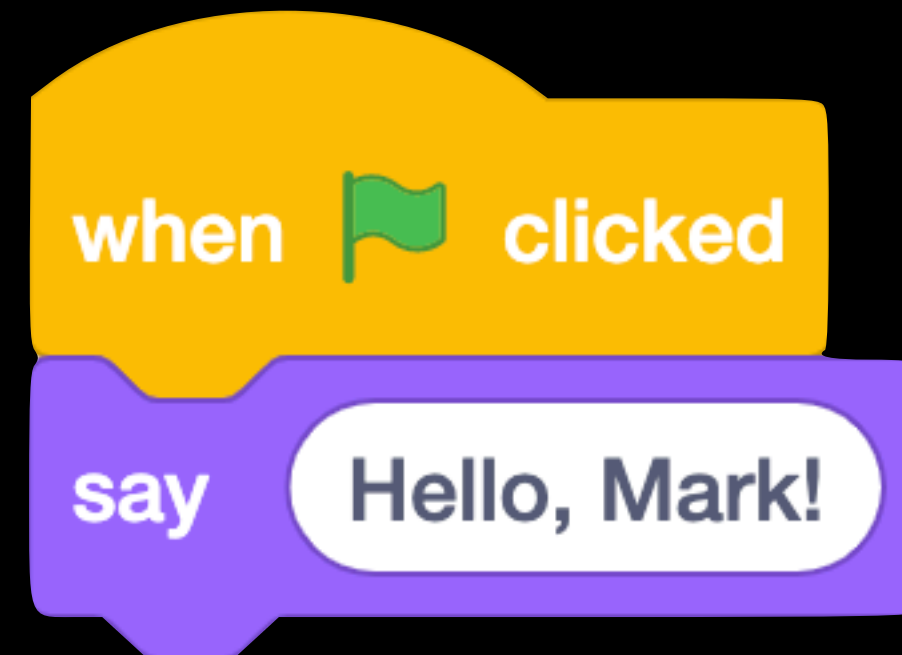
pseudocode

Pseudocode

```
While store operating:

    Wait at the counter until customer comes

    If someone approaches the counter:

            Say "Hey, what can I get for you?", with smile

            If customer replies "Chocolate Waffle"

                    Start making chocolate waffle

                    Say "chocolate waffle is 1 dollar and 20 cents."

            If customer replies "Plain Waffle"

                    Start making Plain waffle

                    Say "plain waffle is 1 dollar."

    If payment received

            Check if payment amount is correct

            Say "thank you"

    If the waffle is ready:

            Pass waffle to the current customer
```
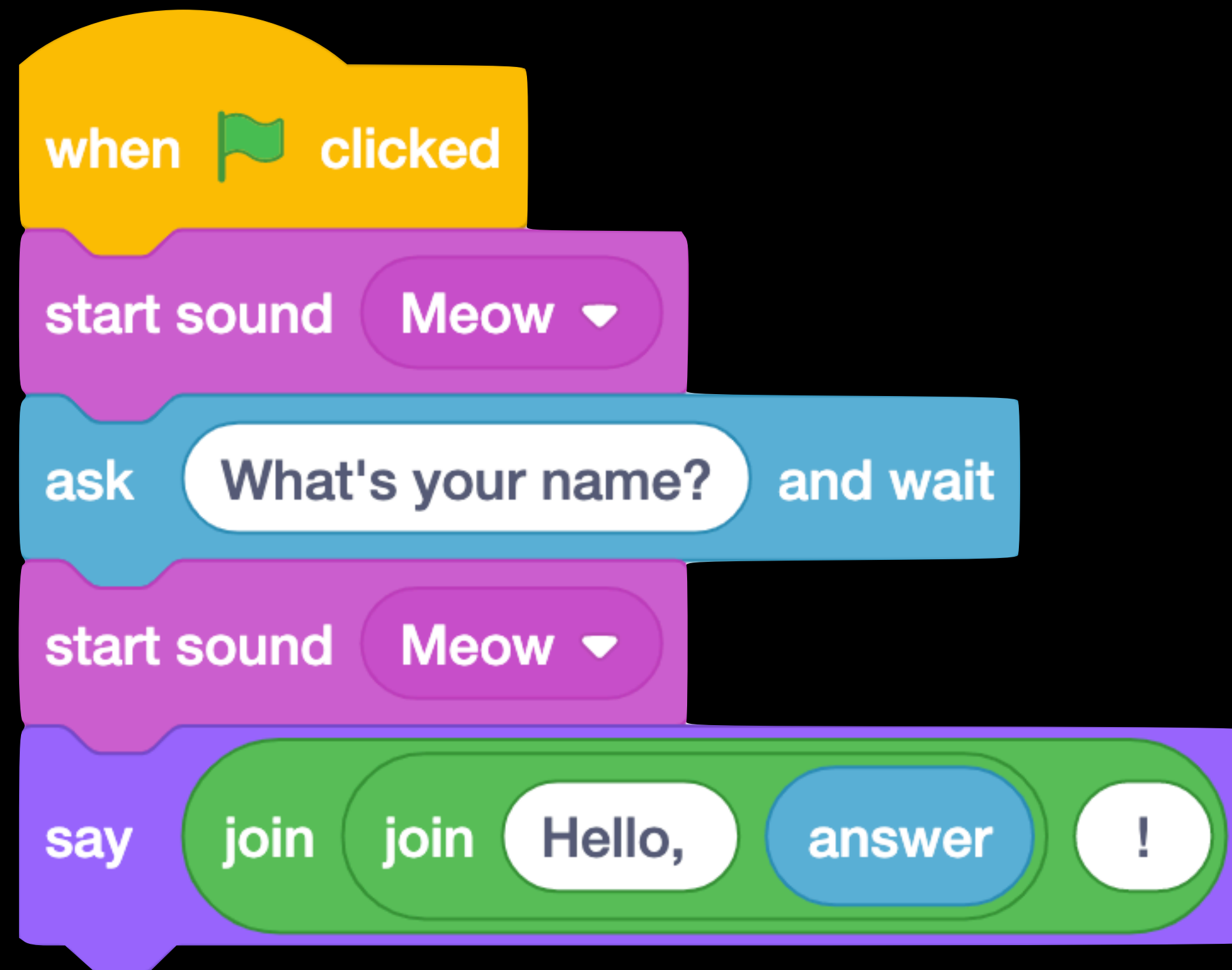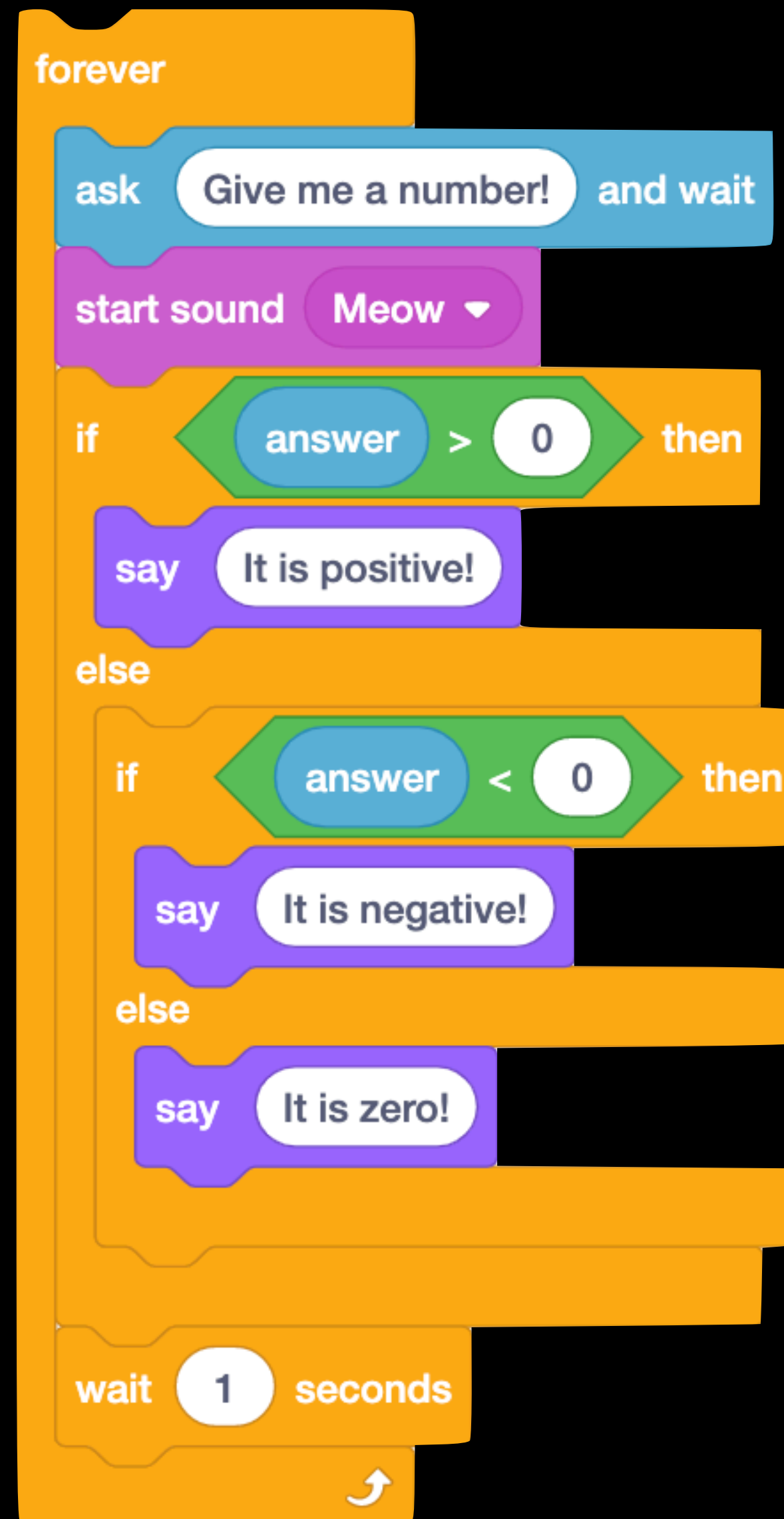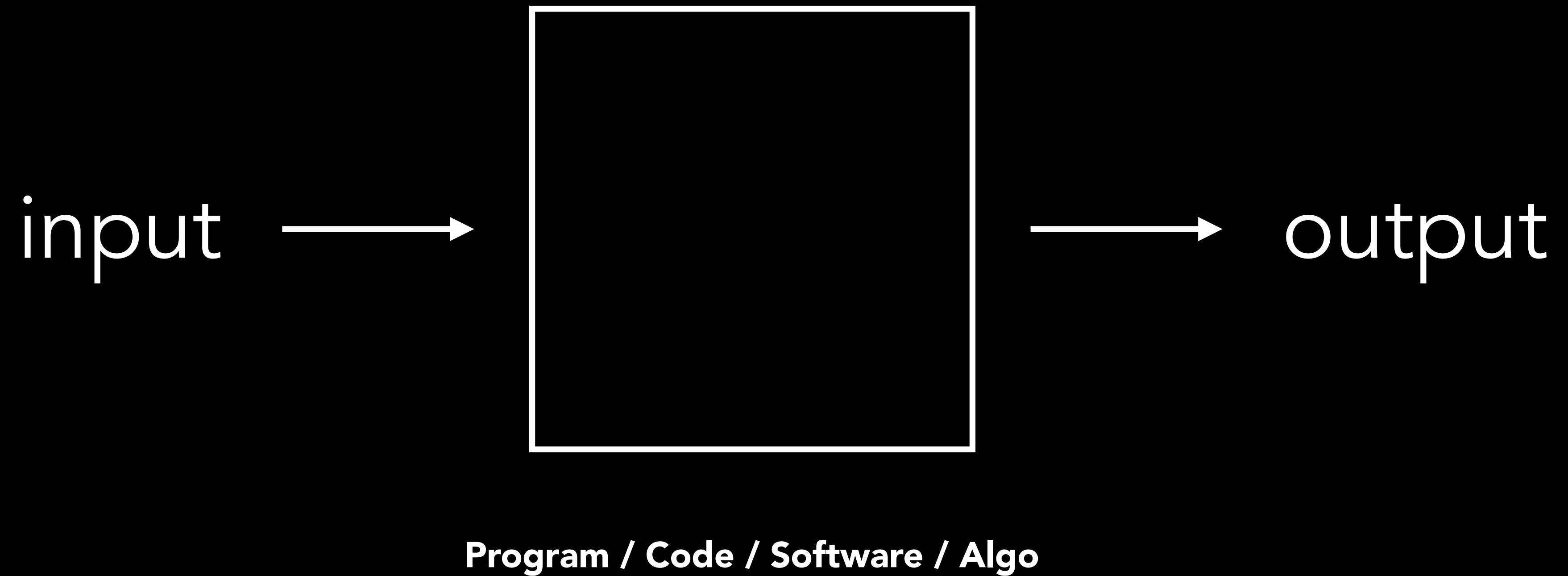
Pseudocode

```
0   While store operating:

1            Wait at the counter until customer comes

2                    If someone approaches the counter:

3                            Say "Hey, what can I get for you?", with smile

4                            If customer replies "Chocolate Waffle"

5                                    Start making chocolate waffle

6                                    Say "chocolate waffle is 1 dollar and 20 cents."

7                            If customer replies "Plain Waffle"

8                                    Start making Plain waffle

9                                    Say "Plain waffle is 1 dollar."

10               If payment received

11                       Check if payment amount is correct

12                       Say "thank you"

13               If the waffle is ready:

14                       Pass waffle to the current customer
```

Pseudocode

```
0    While store operating:

1            Wait at the counter until customer comes

2                    If someone approaches the counter:

3                            Say "Hey, what can I get for you?", with smile

4                            If customer replies "Chocolate Waffle"

5                                    Start making chocolate waffle

6                                    Say "chocolate waffle is 1 dollar and 20 cents."

7                            If customer replies "Plain Waffle"

8                                    Start making Plain waffle

9                                    Say "Plain waffle is 1 dollar."

10                   If payment received

11                           Check if payment amount is correct

12                           Say "thank you"

13                   If the waffle is ready:

14                           Pass waffle to the current customer
```
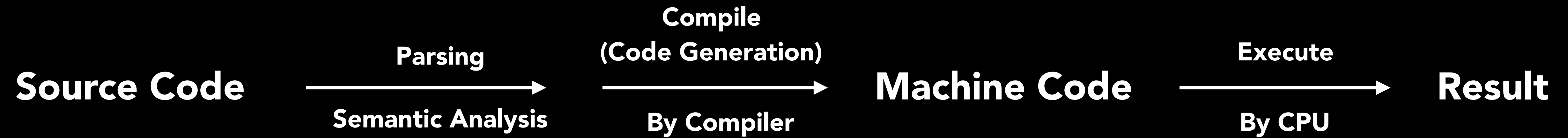
Pseudocode

```
0    while storeIsOperating():

1         if customerApproches():

2              if someoneApproaches():

3                   sayWithSmile("Hey, what can I get for you?")

4                   if customerReply == "Chocolate Waffle"

5                        startMaking("Chocolate Waffle")

6                        say("chocolate waffle is 1 dollar and 20 cents.")

7                   if customerReply == "Plain Waffle"

8                        startMaking("Plain Waffle")

9                        say("Plain waffle is 1 dollar.")

10            if paymentReceived():

11                 checkPayment()

12                 say("thank you")

13            if waffleReady():

14                 passWaffle(currentCustomer)
```
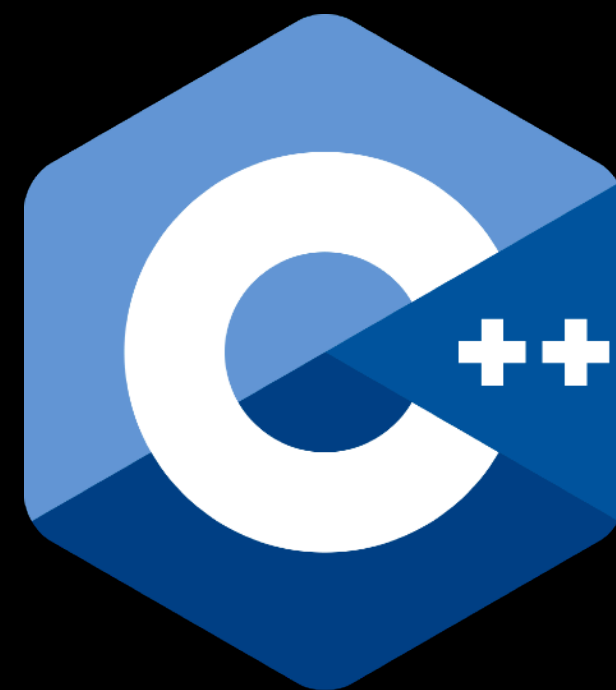
02

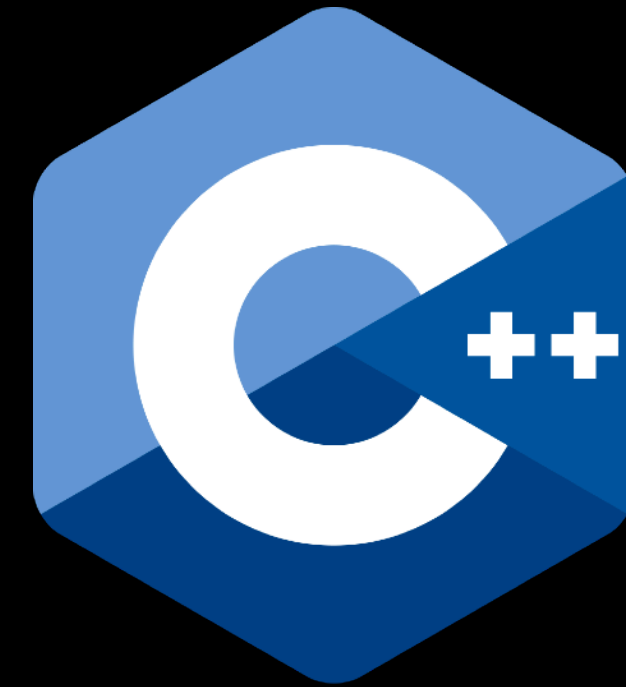Terminologies

Source Code? Machine Code? Executables? the Runtime?

Source Code → *Parsing* / *Semantic Analysis* → *Compile (Code Generation)* / *By Compiler* → **Machine Code** → *Execute* / *By CPU* → **Result**

Source Code → *Parsing* / *Semantic Analysis* → *Interpret & Execute* / *By Interpreter* → **Result**

Interpreted Language? Scripting Language? Compiled Language?

Compiled Programming Languages

Interpreted Programming Languages

High Level / Low Level Programming Languages?

**Domain-specific
Visual Programming**

Very high-level Languages

High-level Languages

Low-level Languages

Dynamo DesignScript

Scratch

Python

C#

C

Assembly Language

Machine Code

Modules? Libraries? Packages?

IDE? Debugger? Text Editor?

Atom, Vim, Emacs, Sublime Text, Visual Studio Code, PyCharm, Spyder

Bugs? Errors? Exceptions?

03

Python 101

Hello

Name

Variable

Python Data Types

Numbers

`int | long | float | complex | bool`

Sequences

`list | tuple | str`

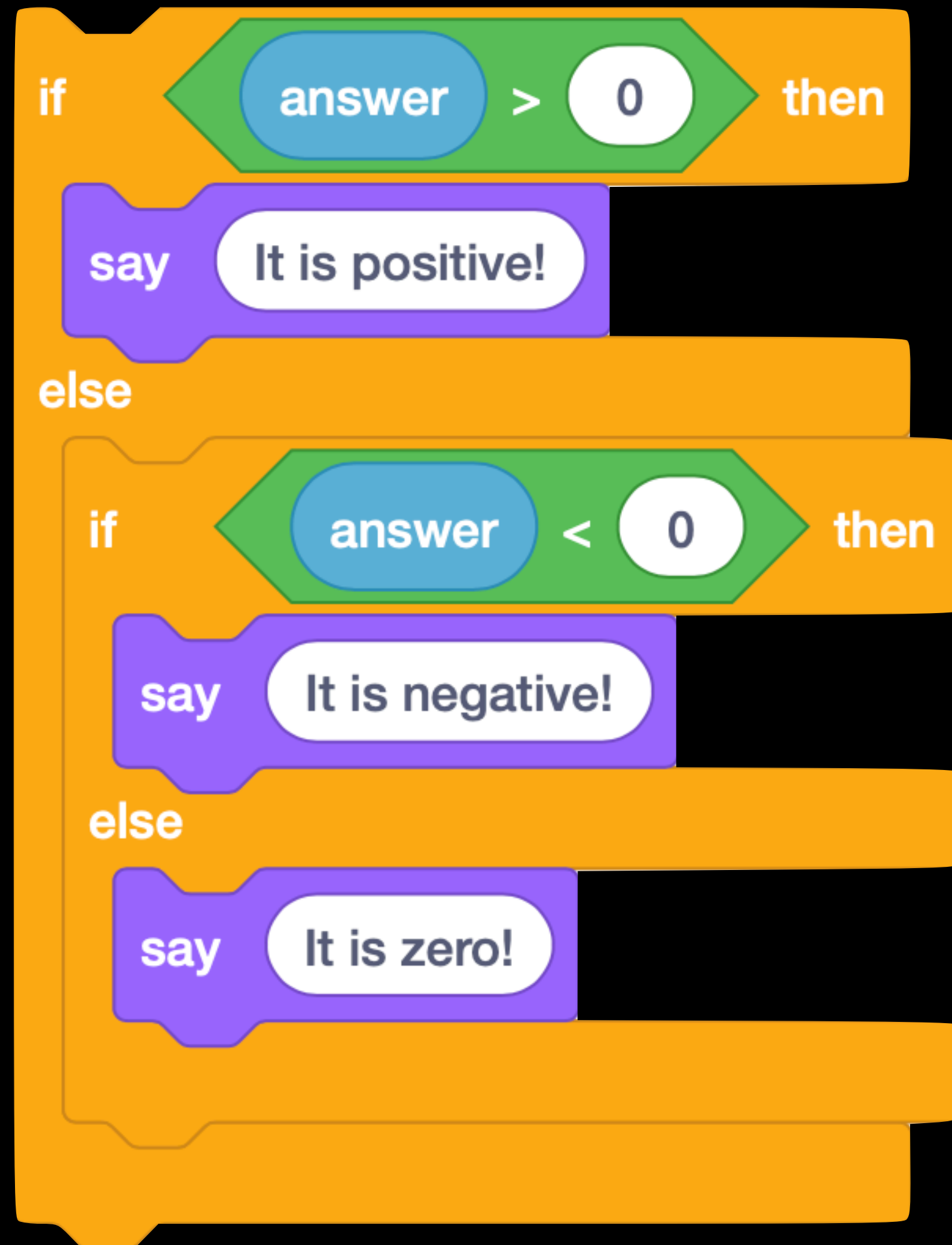## Mappings / Key-value Pairs

`dict`

Set

`set`

Nothing

`None`

Conditions

If … Else if … Else …


```python
if answer > 0:
    print("It is positive!")
elif answer < 0:
    print("It is negative!")
else:
    print("It is zero!")
```

Operators

Arithmetic Operators	`+ - * ** / // %`
Assignment Operators	`= += -= *= **= /= //= %=`
Relational Operators	`== != > >= < <=`
Boolean Operators	`and or not`
Conditional Operators	`if elif else`

Sequences

Ordered	**Range**	Sequence of numbers
	List	Sequence of mutable values
	Tuple	Sequence of immutable values
Unordered	**Dictionary**	Collection of key-value pairs
	Set	Collection of unique values

Mutable		**Range**	Sequence of numbers
		List	Sequence of mutable values
Immutable		**Tuple**	Sequence of immutable values
Mutable		**Dictionary**	Collection of key-value pairs
		Set	Collection of unique values

Loops

The While Loop
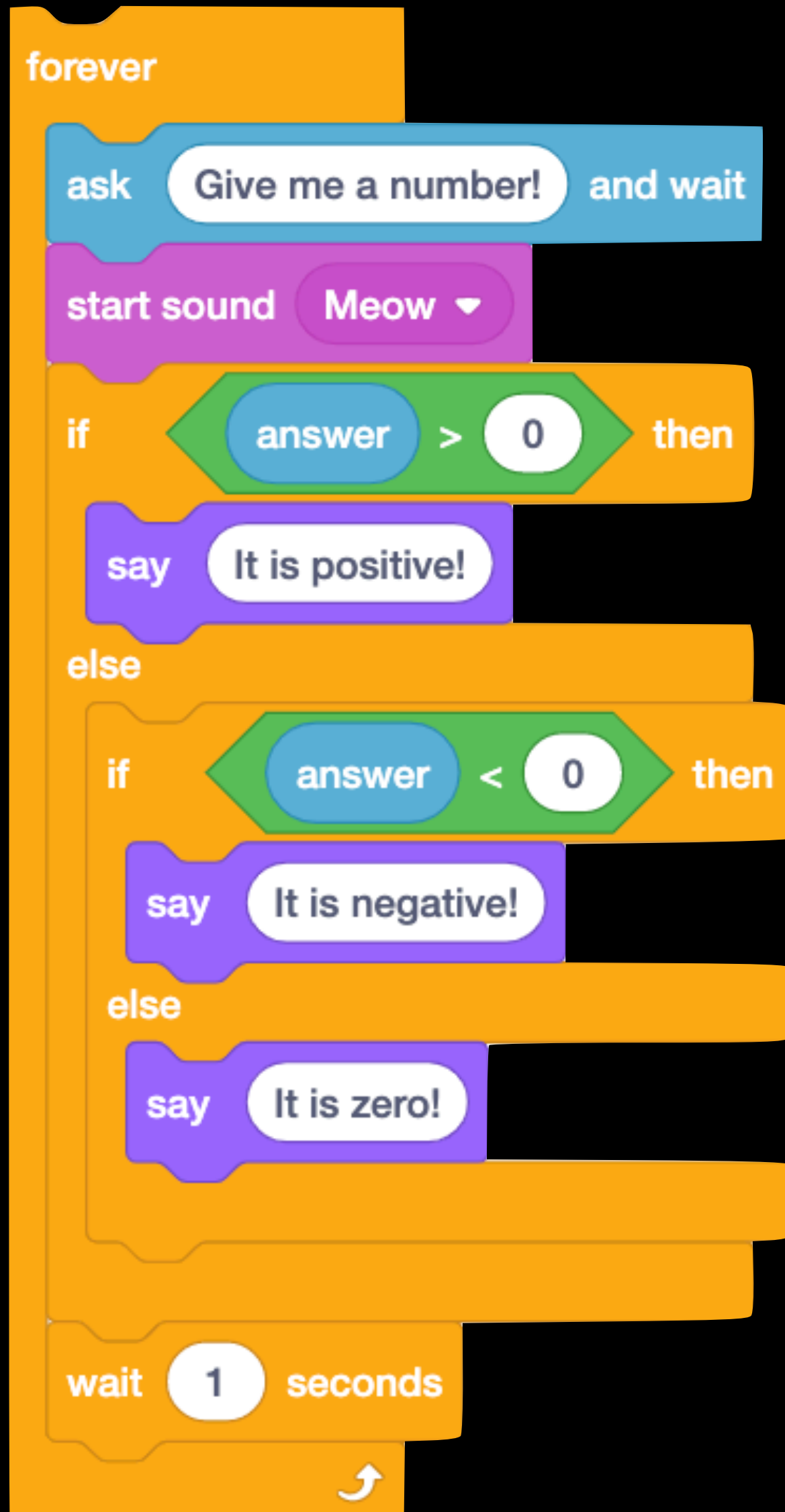

```python
while True:

    answer = input("Give me a number!")
    meow()

    if answer > 0:
        print("It is positive!")
    elif answer < 0:
        print("It is negative!")
    else:
        print("It is zero!")

    time.sleep(1)
```

For Loop

Traversing a sequence

```python
for i in someList:
    # do something about i
    print(i)
```

range(start, stop, step)

```python
for _ in range(100):
    # do something 100 times
    pass
```

`break`:
break out of the loop

`continue`:
**skip the remaining code in the current iteration, enter the
next iteration**
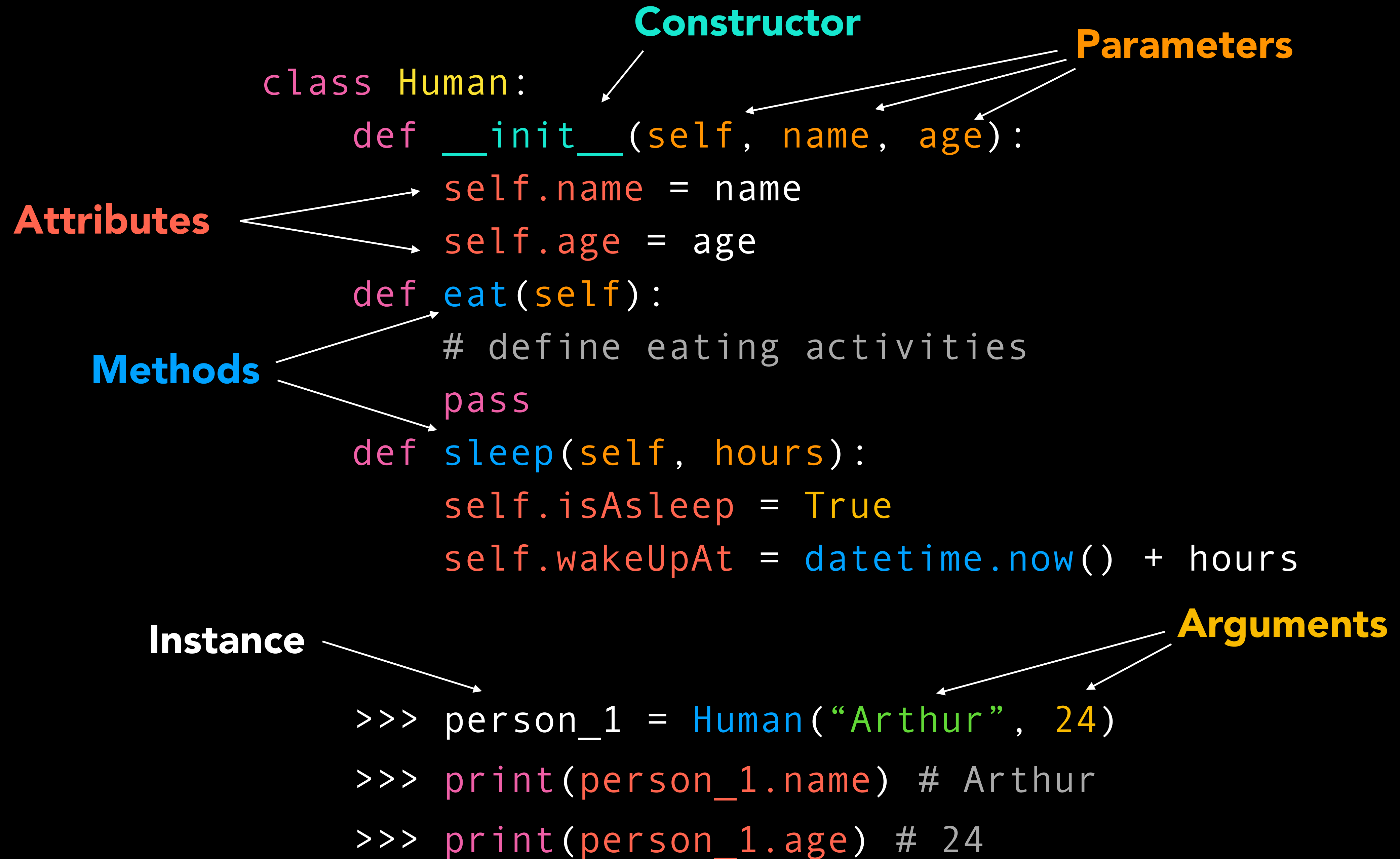

`pass`:
syntax placeholder

Sets

Dictionaries

Functions

Modules

Classes

Parameter A named entity in a function (or method) definition that specifies an argument that the function can accept

Argument A value passed to a function (or method) when calling it

Method Like a function, but it runs on an object

Attribute Instances of a class have attributes, which are pieces of data associated with them, they can be accessed by putting a dot and the attribute name after an instance.

```python
class Human:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def eat(self):
        # define eating activities
        pass
    def sleep(self, hours):
        self.isAsleep = True
        self.wakeUpAt = datetime.now() + hours


>>> person_1 = Human("Arthur", 24)
>>> print(person_1.name) # Arthur
>>> print(person_1.age) # 24
```

Constructor

Parameters

Attributes

Methods

Instance

Arguments

Statements / Keywords in Python

global   import

nonlocal   return

yield   del

raise   pass
                break
assert   continue

Sorting a Sequence in Python

```python
sorted(iterable, key=None, reverse=False)
```

04

Put Into Practice

Make a guess, what is the most frequently used English word in this book?

which word below is the least frequently used among these five?
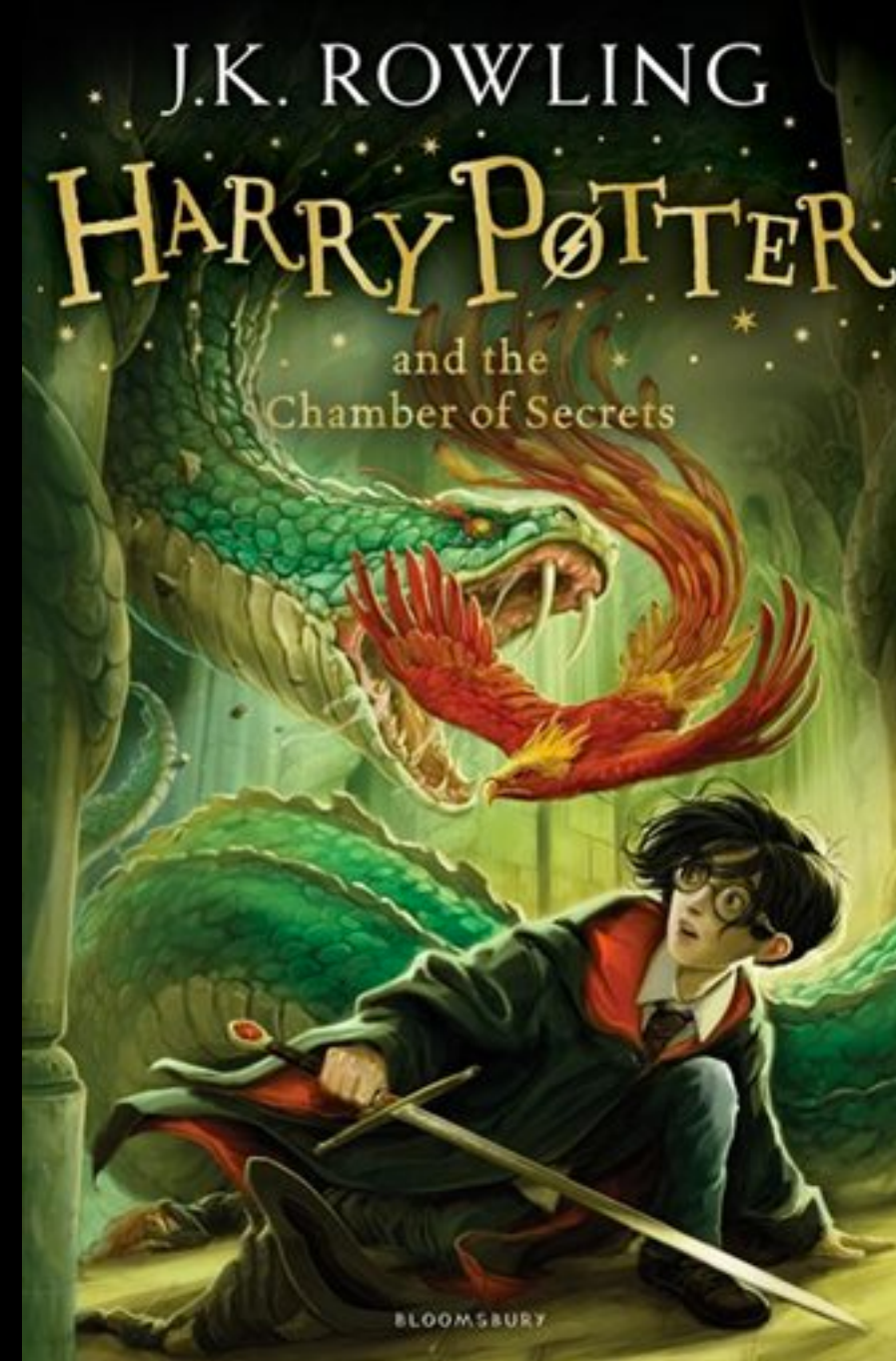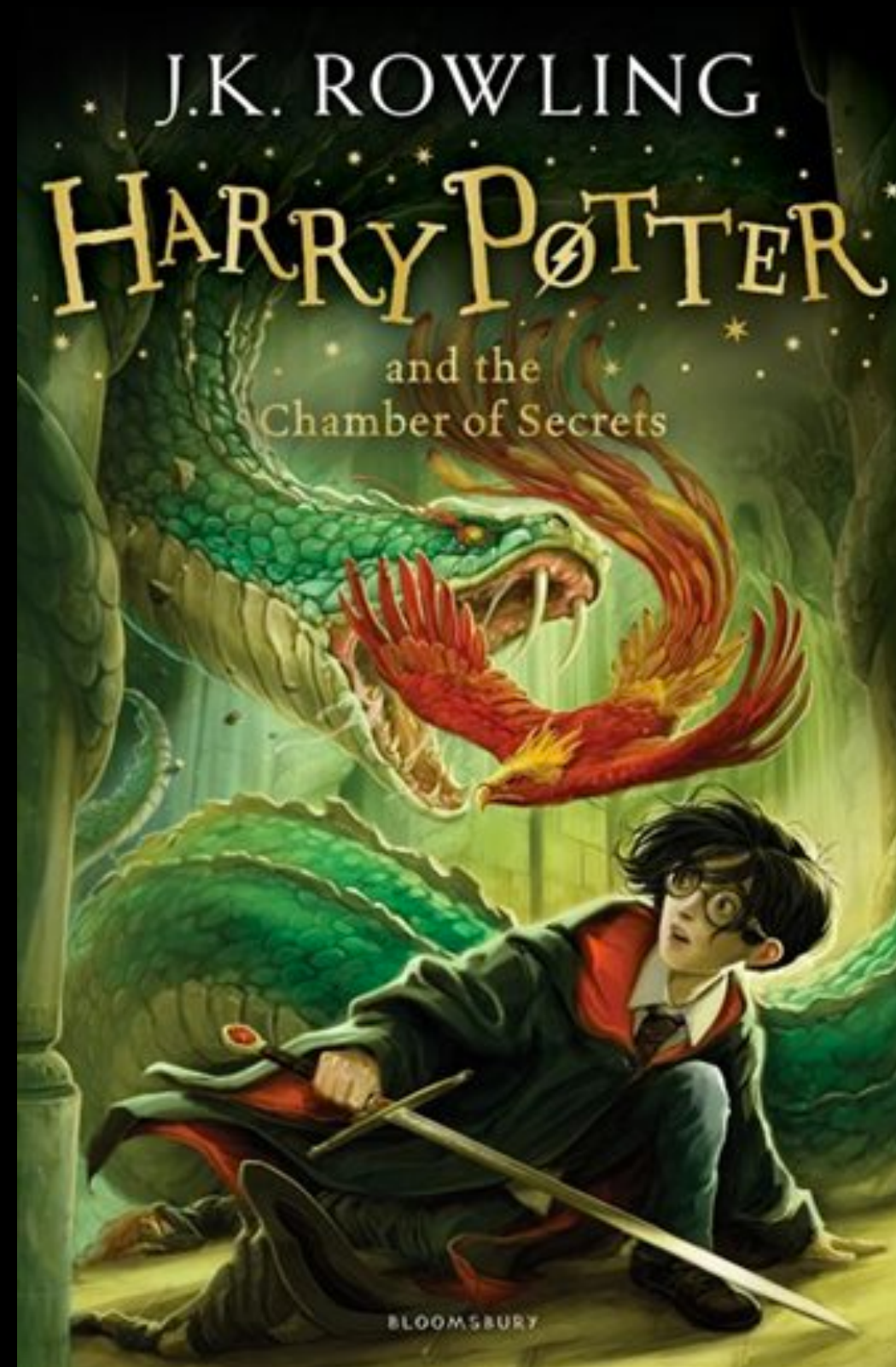
Options: It   Harry   The   Said   A
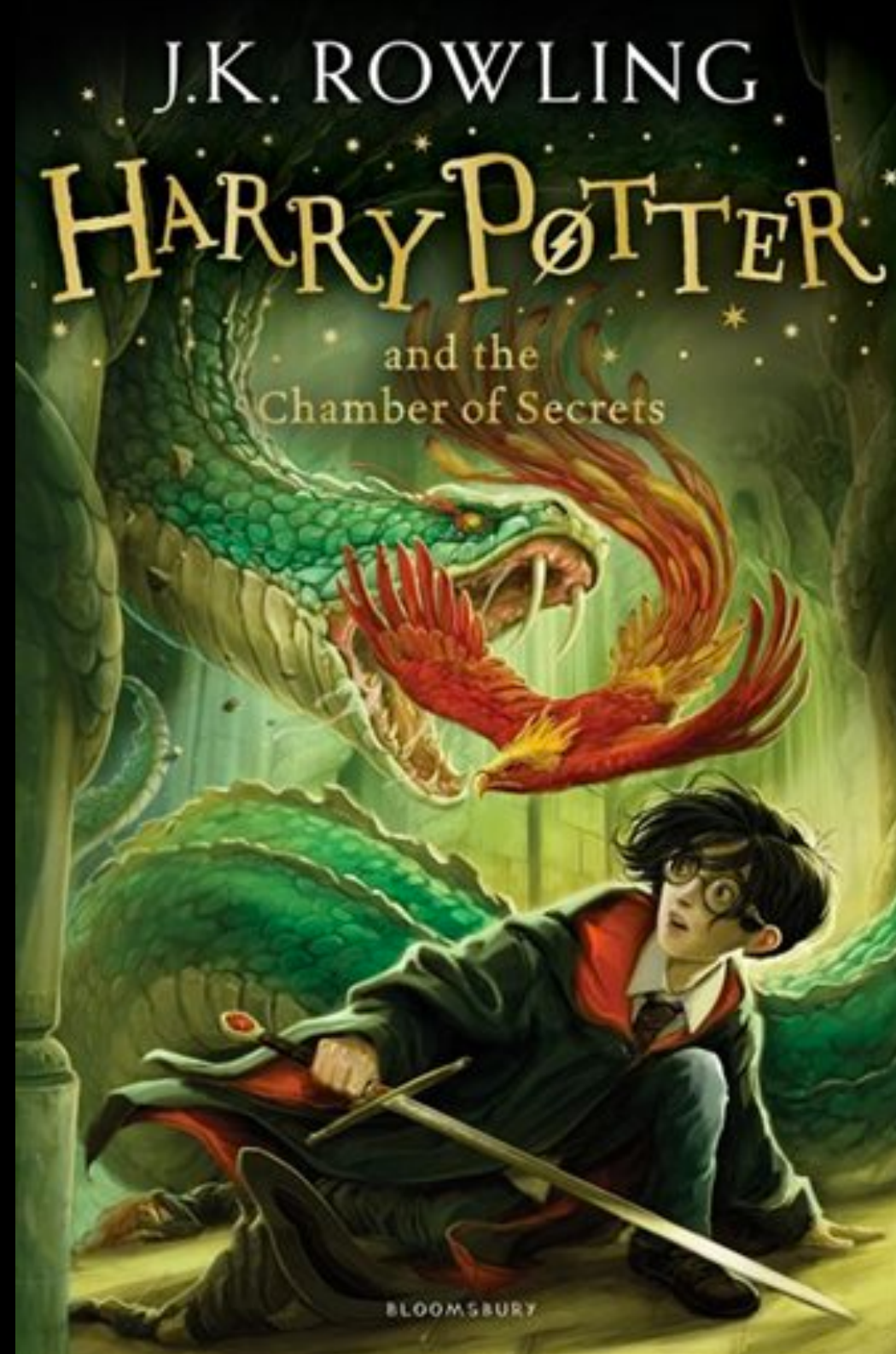

J. K. Rowling - Harry Potter and the Chamber of Secrets

Let's find out!

J. K. Rowling - Harry Potter and the Chamber of Secrets

Let's find out!

You are given a sample text file called Doc0.txt

You are given the book content in a text file called Doc1.txt

What is the total number of words in this book?

What is the total number of unique words in this book?

What is the top 10 most frequently used words in this book?

J. K. Rowling - Harry Potter and the Chamber of Secrets

05

Going Further

Explore More Python Modules

Requests   is an elegant and simple HTTP library for Python.

Numpy   is the fundamental package for scientific computing with Python.

Openpyxl   is a Python library to read/write Excel

Matplotlib   is a Python 2D plotting library

Thank you