

# Programming Thinking

## Introduction

Pepe García



SCHOOL OF  
SCIENCE &  
TECHNOLOGY

## The Professor

Pepe García

## The Professor

Pepe García  
jgarciah@faculty.ie.edu

## The Professor

Pepe García

[jgarciah@faculty.ie.edu](mailto:jgarciah@faculty.ie.edu)

Ask me anything



SCHOOL OF  
SCIENCE &  
TECHNOLOGY

## The Course

- 7 sessions



## The Course

- 7 sessions
- 1 final exam



## Grading

This course is graded as NGS/NGU, and the note will depend on the final exam.

The final exam consists of multiple choice/multiple answer questions, and is open book.

Criteria	Score %
Final Exam	100 %

## Grading

The grading for this course will either be **Non Graded Satisfactory** or **Non Graded Unsatisfactory**.

If you get 50% or more in the overall score, you get **NGS**, and **NGU** otherwise.





## Participation

Please, raise your hand at any point in class if you want to ask something, make an useful comment, or answer a question. (if remote, use Zoom's raise hand feature, so that it's easier to track it)



# Learning Objectives

In this course we will build the fundamentals for the rest of the courses in the masters that rely on programming. We will:



# Learning Objectives

In this course we will build the fundamentals for the rest of the courses in the masters that rely on programming. We will:

- Learn What's programming



# Learning Objectives

In this course we will build the fundamentals for the rest of the courses in the masters that rely on programming. We will:

- Learn What's programming
- Understand how computers execute programs



# Learning Objectives

In this course we will build the fundamentals for the rest of the courses in the masters that rely on programming. We will:

- Learn What's programming
- Understand how computers execute programs
- Learn the basics of Python



# Plan for this session

- Know each other a little bit!

# Plan for this session

- Know each other a little bit!
- Learn about software

# Plan for this session

- Know each other a little bit!
- Learn about software
- Understand what are algorithms and data structures





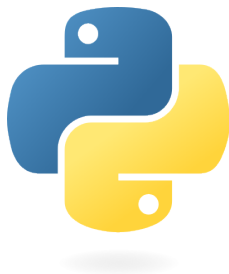
# Plan for this session

- Know each other a little bit!
- Learn about software
- Understand what are algorithms and data structures
- Install Anaconda

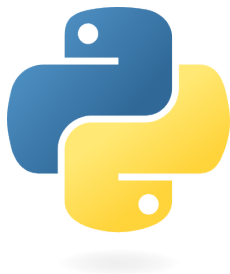


# Join the Slack channel!





Throughout this course we will use Python as our programming language, but there are many more!



Throughout this course we will use Python as our programming language, but there are many more!

What programming languages have you heard of?

There are several ways of classifying programming languages.

## Languages classification

Language	Paradigm	Execution	Purpose
Python	imperative	interpreted	general
Java	object oriented	compiled	general
Javascript	imperative	interpreted	general
Haskell	functional	compiled	general
SQL	declarative	interpreted	specific
HTML	declarative	interpreted	specific

There are several ways of classifying programming languages.

## Languages classification

Language	Paradigm	Execution	Purpose
<b>Python</b>	<b>imperative</b>	<b>interpreted</b>	<b>general</b>
Java	object oriented	compiled	general
Javascript	imperative	interpreted	general
Haskell	functional	compiled	general
SQL	declarative	interpreted	specific
HTML	declarative	interpreted	specific

## Python

Python is one of the most used languages right now. Its applications range from Data Science to Web servers



# How do we write code?



# How do we write code?

Coding is basically putting words together following a programming language specification.



# How do we write code?

Usually, we put these words directly in a text file and then execute it as a program.

```
31 def __init__(self, settings):
32     self.file = None
33     self.fingerprints = set()
34     self.logdups = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, "requests.log"),
39                         "a")
40         self.file.seek(0)
41         self.fingerprints.update(self.request_fingerprint(request) for request in self.requests)
42
43 @classmethod
44 def from_settings(cls, settings):
45     debug = settings.getbool("DEBUG", False)
46     return cls(job_dir(settings), debug)
47
48 def request_seen(self, request):
49     fp = self.request_fingerprint(request)
50     if fp in self.fingerprints:
51         return True
52     self.fingerprints.add(fp)
53     if self.file:
54         self.file.write(fp + os.linesep)
55
56 def request_fingerprint(self, request):
57     return request_fingerprint(request)
```



# How do we write code?

But we can feed these words directly into the programming language **console**.



## Python console

Let's see how we can use the console to code!



# Install Anaconda platform

Now we will install the Anaconda platform in our computers.

- 1 go to <https://www.anaconda.com/products/distribution>
- 2 Download Anaconda distribution for your computer.



# Install Anaconda platform

## Checkpoint

Is anybody lost or has problems installing the software?



# Install VS Code

- 1 Go to <https://code.visualstudio.com/Download>.
- 2 Download VS Code for your computer.



## Checkpoint

Is anybody lost or has problems installing the software?





# Programs



SCHOOL OF  
SCIENCE &  
TECHNOLOGY

## What is a program?

A program is a piece of software with a specific task.



## What is a program?

A program is a piece of software with a specific task. This task can be something **big**, like handling a nuclear reactor, or something **small** like Ctrl-c/Ctrl-v.

## What is a program?

A program is a piece of software with a specific task.

This task can be something **big**, like handling a nuclear reactor, or something **small** like Ctrl-c/Ctrl-v.

There are two main components of programs, **algorithms** & **data structures**.





What is an algorithm?

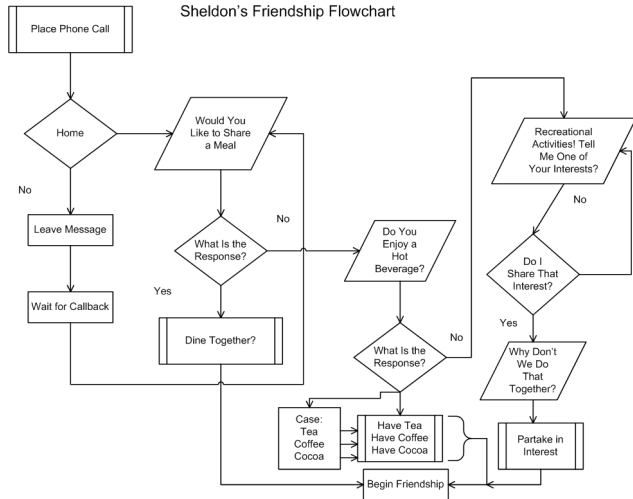


## What is an algorithm?

An algorithm is a sequence of steps that guide the computer in how to solve a problem

## link to the video

Sheldon's Friendship Flowchart





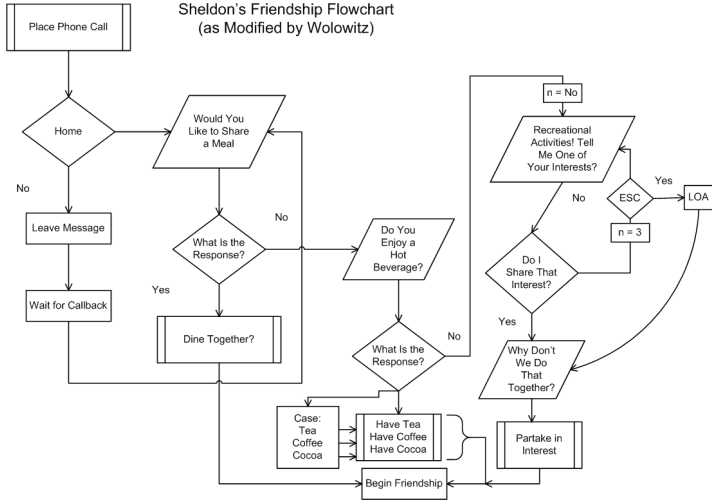
What's wrong with this algorithm? why did Wolowitz need to fix it?

What's wrong with this algorithm? why did Wolowitz need to fix it?

There was a **bug**, an infinite loop

# Algorithms

Sheldon's Friendship Flowchart  
(as Modified by Wolowitz)



What other cases of bugs do we know?

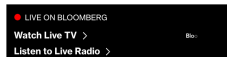
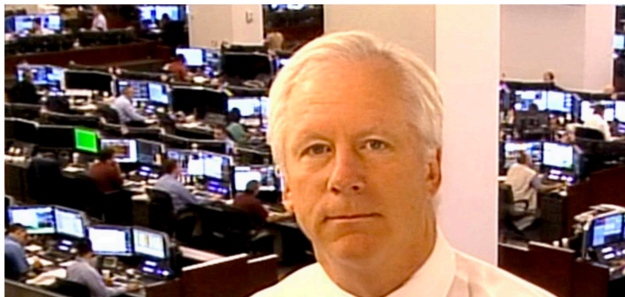


Business

## Knight Shows How to Lose \$440 Million in 30 Minutes

By Matthew Philips

August 2, 2012, 11:10 PM GMT+1



<https://www.bloomberg.com/news/articles/2012-08-02/knight-shows-how-to-lose-440-million-in-30-minutes>

We use different data structures depending what we want to represent.

- Strings: text
- Lists: Twitter's timeline
- Dictionaries: phonebook, DNS
- Stacks: undo/redo ...



# Recap

- We'll use Python for learning programming in this course.





- We'll use Python for learning programming in this course.
- Algorithms, like cooking recipes, will guide our program to perform what we want.

- We'll use Python for learning programming in this course.
- Algorithms, like cooking recipes, will guide our program to perform what we want.
- Different data structures will be used depending on the purpose of our program.

**What Is Code** is a great essay by Paul Ford. (it's a bit long, you don't need to read it for tomorrow)

<https://www.bloomberg.com/graphics/2015-paul-ford-what-is-code/>

Netflix' explained (Coding episode)