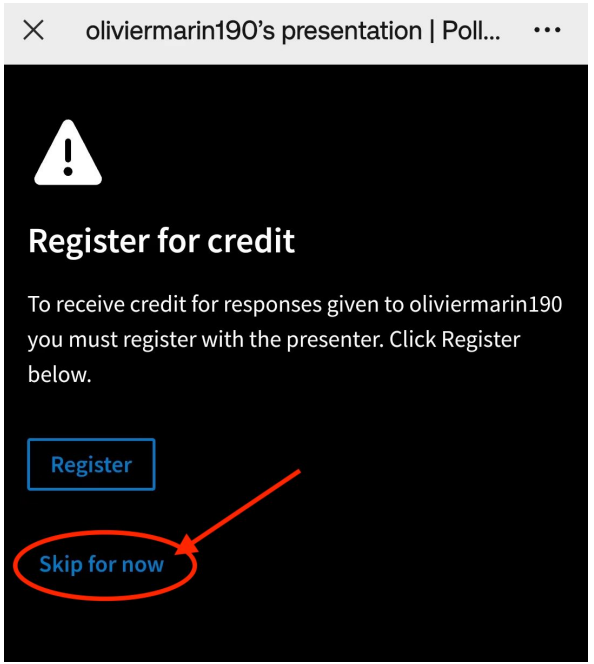
The Art of Programming
A Beginner's Gateway

For the China Welfare Institute

Olivier Marin
https://wp.nyu.edu/omarin/

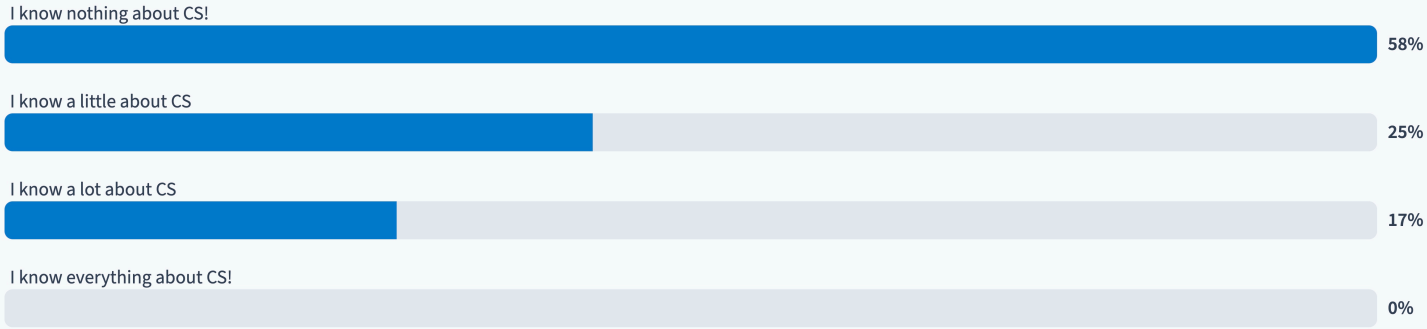
Participating to the poll

PollEv.com/oliviermarin190



Join by Web **PollEv.com/oliviermarin190**

How much do you know about Computer Science (CS)?



What Computer Science is *not*

Computer Science (CS) is ***not*** the study of computers

Astronomy isn't the study of telescopes
Biology isn't the study of microscopes
Chemistry isn't the study of test tubes

Science is not about tools…

I don't know why your computer is broken, nor how to fix it
I don't know why your WiFi connexion is slow today

What *is* Computer Science?

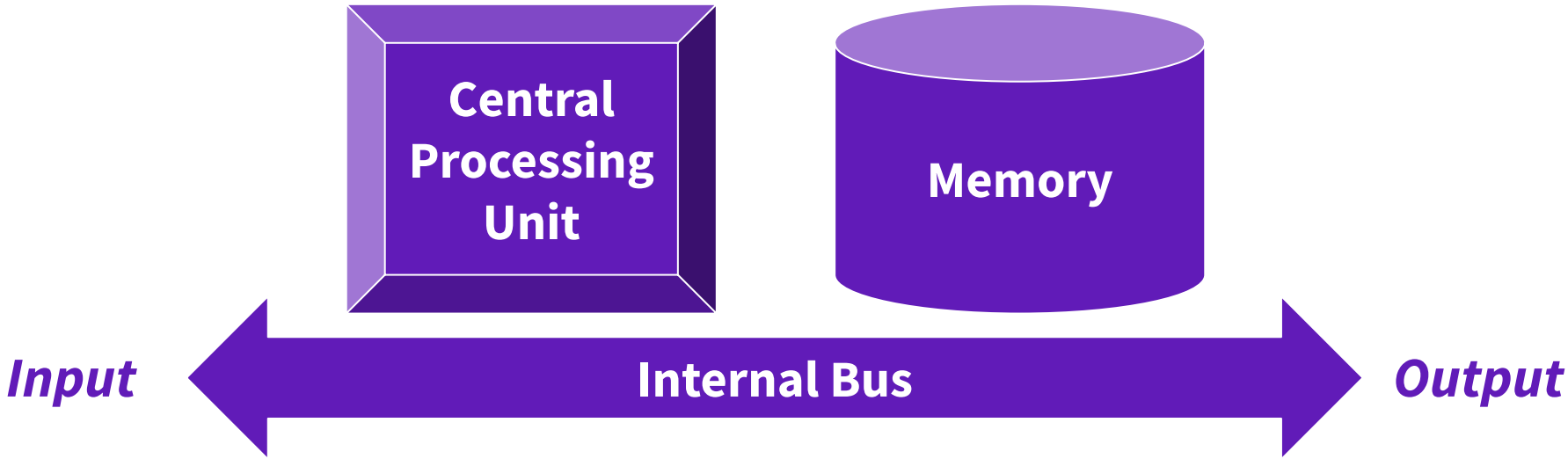
Computer Science is the study of *algorithms*

An algorithm is a precisely defined process for solving a problem
Usually in a way that can be automated easily

An algorithm consists of a sequence of instructions

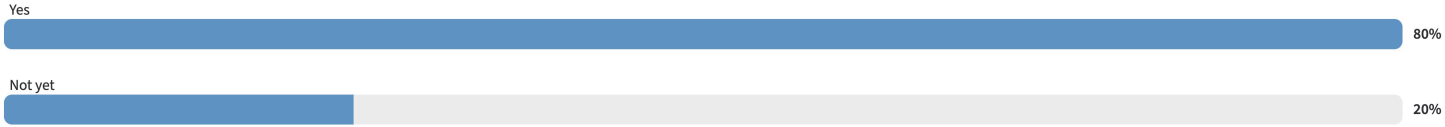
Introductions: This is a Computer

Introductions: This is ALSO a Computer



Join by Web **PollEv.com/oliviermarin190**

Have you ever programmed before?



Programming: a First Definition

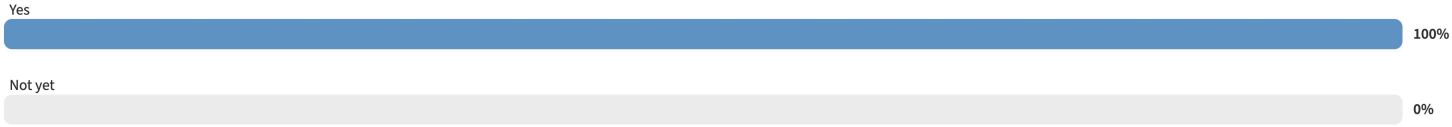
1. Take up a task
2. Decide how to complete the task
3. Give instructions to apply the decision

Join by Web **PollEv.com/oliviermarin190**

Let's think about at your daily life responsibilities

Join by Web **PollEv.com/oliviermarin190**

Have you ever programmed before? (2)



Programming: a Second Definition

1. Get the computer to do something for you

Programming tutorial 1: Let's make a sandwich!

Functions: How to issue an instruction

A function is a predefined operation that issues a command in a programming language

Your first function in Python:

```
print(__)  #displays the argument between the parentheses
```

Example:

```
>>> print("message")
    message
```

Programming tutorial 2: Let's say hello

Saying hello: Solution 1

```
print("Hello World!")
```

Handling Data: Values, Types, and Variables

Computers are meant to manipulate data for you

- You enter input values, and you expect to get output values
- A computer only understands 2 values: `0` and `1`
- But we can use these to represent much more information!
- 3 types: *Boolean (`bool`)*, *Integer (`int`)*, *String (`str`)*
- A variable is a memory location where the computer stores values

Saying hello: Solution 2

```
phrase = "Hello World!"
print(phrase)
phrase = "大家好!"
print(phrase)
```

Programming tutorial 3: Let's ask for input

Functions: How to ask for input

Your second function in Python:

```
input(<message>)    #displays the message between the parentheses
                    #waits for a value from the user
                    #returns this value as a string
```

Example:

```
>>> input("Enter a number between 1 and 9:")
    Enter a number between 1 and 9:<user prompt>
```

Asking for input: Solution 1

```
name = input("Who do you want to greet?")
print("Hello", name, "!")
```

Asking for input: Solution 2

```
value1 = input("What is the first value?")
value2 = input("What is the second value?")
print(int(value1) + int(value2))
```

Conditional Structures: How to make decisions

Conditional Structure in Python:

```
check = True
if (check):
    <instruction 1>
else:
    <instruction 2>
```

Programming tutorial 4: Let's make a decision

Making a decision: Solution

```
answer = input("Are you OK?")
if (answer == "Yes"):
    print("Good!")
else:
    print("How can I help?")
```

Making a decision: Solution

```
value1 = input("What is the first value?")
value2 = input("What is the second value?")
if (value1 == value2):
    print("Same values")
elif (value1 > value2):
    print("The first is larger")
else:
    print("The first is smaller")
```

Repetition Structures: How to repeat instructions

Repetition Structure in Python:

```
check = True
while (check):
    <instruction 1>
<instruction 2>
```

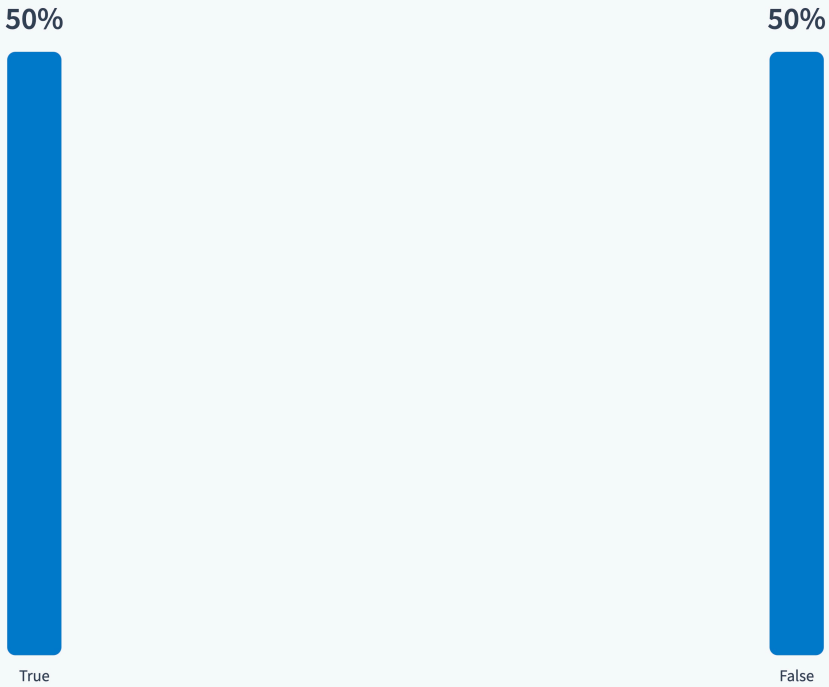
Programming tutorial 5: Let's count to 10

Counting to 10: Solution

```
i = 1
while (i <= 10):
    print(i, end=' ')
    i = i + 1
```

Join by Web PollEv.com/oliviermarin190

A computer program will only do what you tell it to

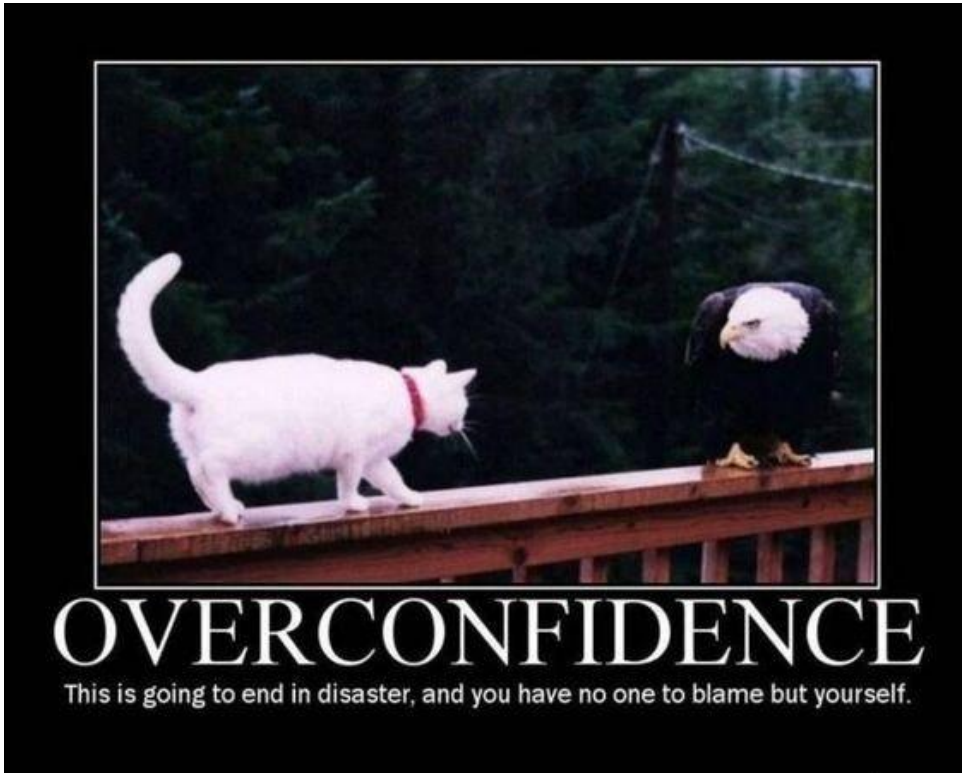


The Tale of the Intrepid Programmer

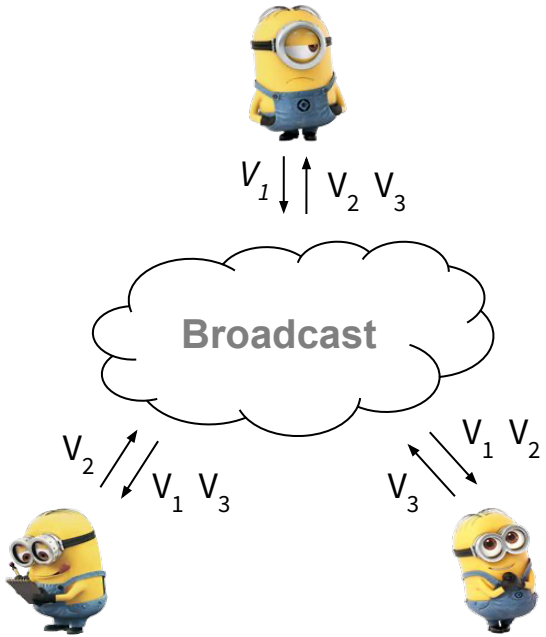
Common CS fallacies

- ✗ Everything goes fast
- ✗ Everything is deterministic
- ✗ Everything is safe

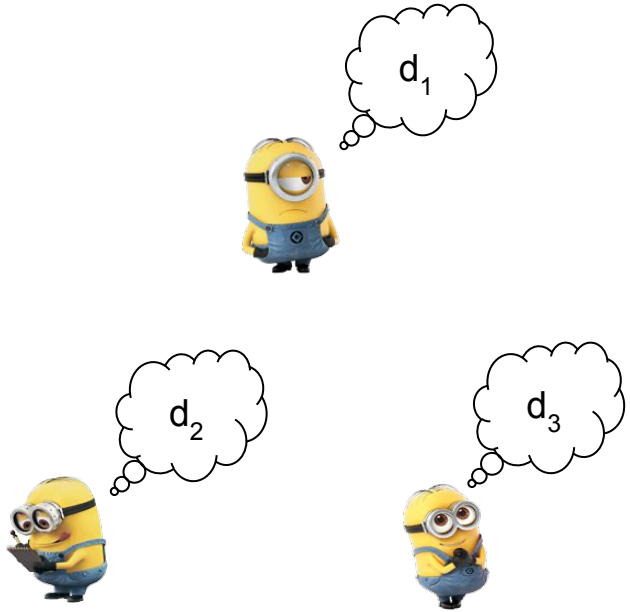
Murphy's Law



Anatomy of a Distributed Consensus



Step 1: Propose

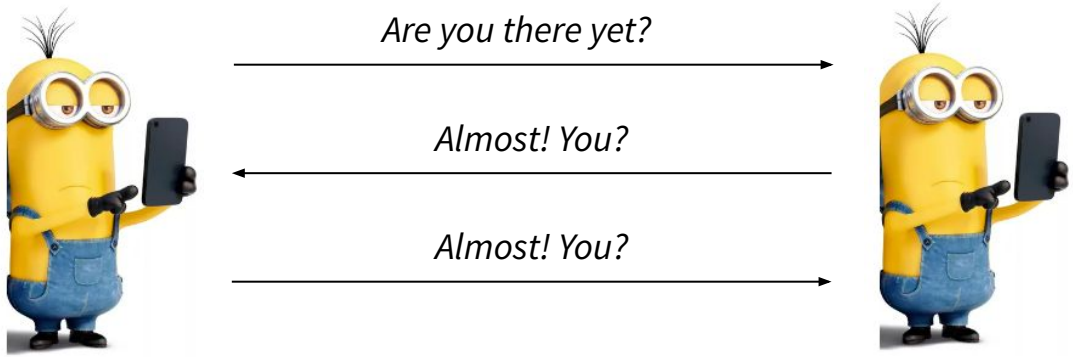


Step 2: Decide

Programming tutorial 6: Let's Agree on Something

The 火锅 Conundrum (aka the *Two Generals' Problem*)

- Several friends want to get together to eat a hot pot
- None of them wants to wait alone at the restaurant
- They communicate by WeChat
- Each of them texts to check that the other has reached the restaurant



Takeaway points

A program is a sequence of instructions

Finding the right sequence is hard!

It takes a lot of practice, and even then:

- you *will* make mistakes
- the universe *does actually* conspire against you

What to do?

Be patient
Learn from your failures
Keep at it until you're (somewhat) confident
Accept that *you can't control everything*