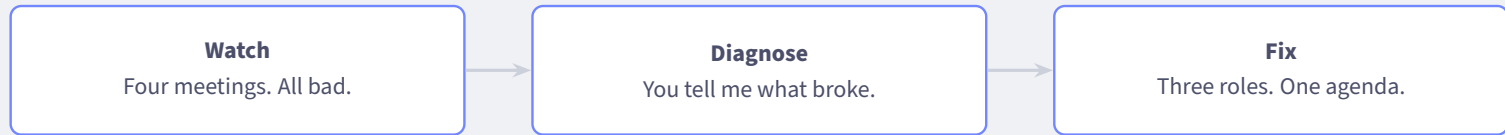


Meetings

Four meetings that went wrong



Meeting

Agenda. Turns. Decisions.

Workshop

No agenda. Laptops. Work.

The project itself you build on your own, or together in a workshop. How you split that is up to you. Neither of them is the meeting.

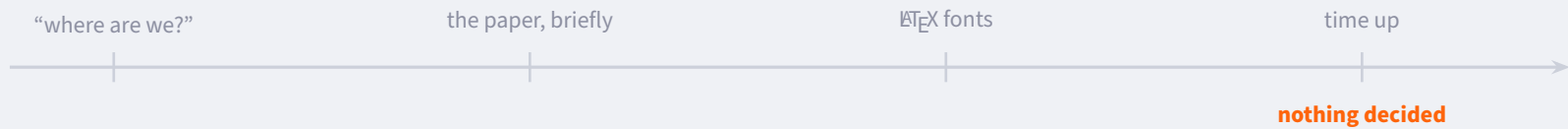
Which one have your group sessions been so far?

Inform

Discuss

Decide

If you cannot say which one an item is, it is not ready.



What does the secretary write in the minutes?

10:00 Item 1: report on Chapter 3.

10:04 B spots a bug. Everyone leans in.

10:40 Bug fixed. A good fix.

10:41 Items 2, 3, 4 never reached.

10:42 D, silent throughout, leaves.

Real work got done. What did it cost D?

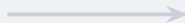


35 of 40 minutes. Everyone agreed.
Minutes: three decisions, no disagreement.

The minutes look perfect. What is missing?

In the room

Good discussion. A real decision.

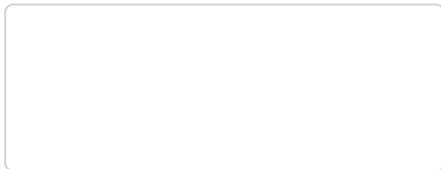


A week later

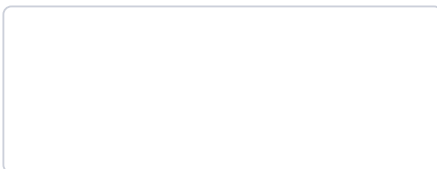
Who agreed what? No minutes.

Nothing went wrong in the room. What did it cost?

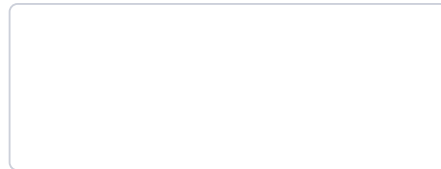
Chair



Secretary



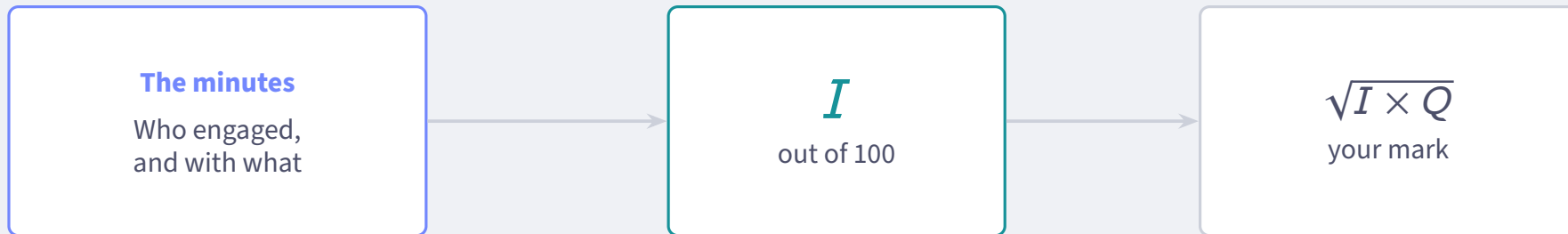
Members



What does each of them actually do? And how often do they change?

- 1. Inform.** What each member read.
- 2. Discuss.** Questions, and the project.
- 3. Decide.** Who does what, by when.

Every week. The secretary writes it down as you go.



Meet **this week** in your own group slot.

Bring the completed Week 5 minutes to **the next session**.

No minutes handed in means no record of engagement, for every member of the group.

Variables, conditionals, loops

Storing a value, and doing something a number of times.

Functions and data structures

Naming a calculation, and choosing how to hold a collection.

You read these **on your own**. The next four slides are the parts that catch people.

```
number = 4  
if number = 5:  
    print("five")
```

What does Python do here, and why?

One sign assigns a value. Two signs ask a question.

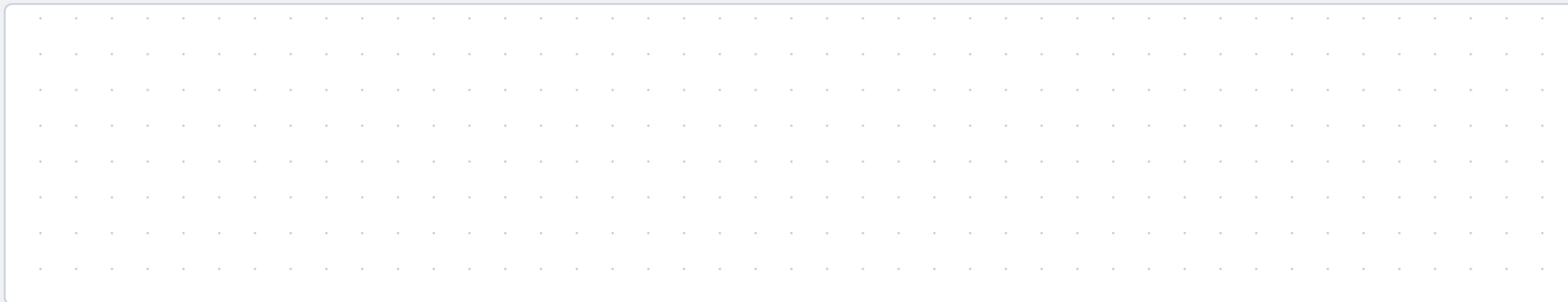
```
total = 0
number = 1
while number <= 10:
    total = total + number
```

This runs forever. Add the missing line.

A `while` loop stops only when something in the body changes the condition.

```
def mean(values):  
    """Return the mean of a list of numbers."""  
    return sum(values) / len(values)
```

You have written the same three lines in two places. Now what?



If it has a name worth giving it, it is a function. The docstring is part of the mark.

list

tuple

set

dict

Which one holds: the roots of a polynomial, a point in the plane, the vertices of a graph, a lookup from name to mark?

Ask the mathematics first. Does order matter? Can there be repeats? Does it change?

You pick a **problem in mathematics**.

You write a small **Python library** that does something useful with it.

You write a **three page paper** about it, and you **record a presentation**.

That is the whole project. Everything else on the website is detail about how it is marked.

1. **On your own.** Work through this week's chapter of Python for Mathematics.
2. **In your group slot.** Hold the meeting. Each of you reports what you read.
3. **In your own time.** Build the library, on your own or together in a workshop.
4. **At the next class.** Hand the minutes in.

Five times over: weeks 4, 5, 8, 9 and 10. Those five are what *I* is counted from.

Weeks 4-5

Meet. Learn Python. Pick a topic.

Weeks 6-7

No meetings. Individual coursework.

Weeks 8-10

Meet again. Build it. Write the paper.

Week 11

Hand in eight files, as a group.

The gap in the middle is deliberate. You are not behind if nothing happens in weeks 6 and 7.

This week. Meet, and bring the Week 5 minutes to the next session.

Before Week 8. Decide what your library does, and email me the idea. One paragraph is enough.

Any time at all. Ask in the contact hour, or email knightva@cardiff.ac.uk.

‘We do not know what we are supposed to be doing’ is a reasonable thing to say. Say it early.

1. Confirm your **secretary rota**:
2. Confirm **when and where** you meet:
3. Next week's chair:
4. **Meet this week. Bring the minutes to the next session.**