GROUP THEORY - A VISUAL INTRODUCTION

MATHEMATICS & COMPUTER SCIENCE

IAN & NEELU @ADA

- **Our toolkit** Python in a jupyter notebook
- **Objects, actions, and structure** What maths is really about
- **Using Python to visualise structure** How computer science helps
- **What is a group structure** and why does it matter?
- **Looking for groups** computationally

USING THE JUPYTER NOTEBOOK

Open `0_welcome.ipynb` and follow the instructions

OBJECTS, ACTIONS, AND STRUCTURE

- Find some mathematical objects, or invent them
- Combine them somehow
- Look for patterns

OBJECTS

- Numbers, obviously
- Different kinds of numbers! Integers, rational numbers, real numbers, imaginary numbers, infinite numbers ...
- Symmetries. Transformations
- Functions
- Vectors. Matrices

ACTIONS

- Take two numbers and do the usual $+ - \times \div$
- Take two transformations and do one followed by the other
- Take two functions and find the composite function
- Add two vectors

STRUCTURE

EXAMPLE

- **Objects** The numbers 0, 1, 2
- **Action** Addition *modulo* 3

Remember *modulo* 3 means "find the remainder when you divide by 3"

So

$$4 = 1 \mod 3$$

And

$$15 = 0 \mod 3$$

In Python it's written %3

$+$	0	1	2
0			
1			
2			

EXAMPLE

- **Objects** Rotations of an equilateral triangle
- **Action** Composition (one followed by another)

Followed by	Rot 0°	Rot 120°	Rot 240°
Rot 0°			
Rot 120°			
Rot 240°			

$+$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Followed by	Rot $0°$	Rot $120°$	Rot $240°$
Rot $0°$	Rot $0°$	Rot $120°$	Rot $240°$
Rot $120°$	Rot $120°$	Rot $240°$	Rot $0°$
Rot $240°$	Rot $240°$	Rot $0°$	Rot $120°$

What do you notice about the two tables?

"Addition mod 3" has the same *structure* as "Rotations of an equilateral triangle"

(Technically we'd say they're **isomorphic**)

EXAMPLE

- **Objects** The numbers 0,1,2,3
- **Action** Multiplication *modulo* 4

$\times$	0	1	2	3
0				
1				
2				
3				

EXAMPLE

- **Objects** The symmetries of a rectangle
- I =identity (do nothing)
- R =rotate (through $180°$)
- H =horizontal reflection
- V =vertical reflection
- **Action** Composition (one followed by another)

$\circ$	I	R	H	V
I				
R				
H				
V				

$\times$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

$\circ$	I	R	H	V
I	I	R	H	V
R	R	I	V	H
H	H	V	I	R
V	V	H	R	I

What do you notice about the tables?

- They're not the same structure
- They both have an **identity** object
- Some objects have **inverses**
- What's going on with 2×2?

Filling out these tables by hand gets tedious, so let's use Python to speed things up.

Make a new cell at the bottom of your notebook and follow along.

```
objects = [1,2,3,4]

def action(a, b):
    return (a * b)%5


for a in objects:
    print([action(a, b) for b in objects])
```

GROUPS

If a structure meets certain requirements, we call it a **group**

- **Closure** The action shouldn't produce objects we haven't seen before
- **Identity** There should be an object that doesn't do anything
- **Inverses** Every object should have an inverse. An object and its inverse should produce the identity
- **Associativity** eg $(a+b)+c = a+(b+c)$

Look at the tables so far. Can you tell from the tables which ones are *groups*?

Now open `1_exploring_structure.ipynb` and follow the instructions