Name ____________________ Date ____________

Unit 9 · Sequences and functions — teaching slides

18 slides, formatted for printing.

9.1 Generating sequences

Sequences and terms

- A **sequence** is a list of numbers in a definite order.
- Each number is a **term**.
- We speak of the 1st term, the 2nd term, and so on.
- The order matters — the same numbers rearranged are a different sequence.

```
3, 7, 11, 15, 19, …
1st term 3 2nd term 7 3rd term 11
```

The term-to-term rule

- It tells you how to get from one term to the **next**.
- “Add 4”, “subtract 7”, “multiply by 3”.
- To use it you must already know the previous term.
- So it is a stepping rule, not a shortcut.

```
first term 3, rule “add 4”
3 → 7 → 11 → 15 → 19
```

You need both

- A rule alone does not fix a sequence.
- “Add 4” fits 1, 5, 9, … and also 3, 7, 11, …
- The first term pins it down.
- So always state both when you describe a sequence.

```
rule “add 4”, first term 1 → 1, 5, 9, 13
rule “add 4”, first term 3 → 3, 7, 11, 15
```

Linear sequences

- A sequence is **linear** if the gap is always the same.
- That constant gap is the **common difference**.
- Find it by subtracting a term from the next one.
- Then check it is the same all the way along — one gap is not proof.

```
5, 9, 13, 17
9 − 5 = 4 13 − 9 = 4 17 − 13 = 4
common difference 4
```

Counting the steps

- Going from the 1st term to the kth takes k − 1 steps.
- The first term needs no step — you are already there.
- So the kth term is first + (k − 1) × difference.
- Getting this off by one is the single commonest error in the unit.

```
first 4, difference 6, want the 10th
9 steps, not 10
4 + 9 × 6 = 58
```

Sequences that are not linear

- Some sequences multiply instead of adding.
- Some have gaps that themselves form a pattern.
- Always check the gaps before assuming linear.
- The famous ones are worth knowing by sight.

```
2, 6, 18, 54 × 3 each time
1, 4, 9, 16 square numbers
1, 3, 6, 10 triangular numbers
```

9.2 Using the nth term

Why the nth term is better

- The term-to-term rule makes you step through every term.
- For the 100th term that is 99 steps.
- The **nth term** rule jumps straight there.
- It is a rule about *position*, not about the previous term.

```
term-to-term: 3 → 7 → 11 → 15 → … 99 steps
nth term 4n − 1: put n = 100 → 399
```

The coefficient is the difference

- Look at the gap between terms.
- That gap is the number in front of n.
- A gap of 5 means the rule starts 5n.
- A decreasing sequence gives a negative coefficient.

```
5, 9, 13, 17 gap 4 → starts 4n
20, 17, 14 gap −3 → starts −3n
```

Finding the constant

- Write out what the coefficient alone would give.
- Compare with the sequence you actually have.
- The difference is the constant to add or subtract.
- Then write the full rule.

```
want 7, 10, 13, 16
3n gives 3, 6, 9, 12
always 4 more → 3n + 4
```

Check by substituting

- Put n = 1, then n = 2, then n = 3.
- Compare each with the sequence you were given.
- If all three match, the rule is right.
- This takes seconds and catches nearly every mistake.

```
rule 3n + 4
n=1 → 7 ✓ n=2 → 10 ✓ n=3 → 13 ✓
```

Finding a position

- To ask “which term is 50?”, set the rule equal to 50.
- Then solve the equation for n.
- The answer must be a whole number.
- That is how you also test whether a number is in the sequence at all.

```
6n − 4 = 50
6n = 54
n = 9 → it is the 9th term
```

When a number is not in the sequence

- Set the rule equal to the number and solve as usual.
- If n comes out as a fraction, the number is not a term.
- Positions are counting numbers — there is no 24.5th term.
- Say so explicitly; that reasoning is what earns the mark.

```
is 100 in 4n + 2?
4n = 98, n = 24.5
not a whole position → no
```

9.3 Functions and function machines

What a function is

- A **function** takes an input and gives exactly one output.
- The same input always gives the same output.
- A **function machine** draws it as a chain of operations.
- The number going in is the **input**; the number coming out is the **output**.

```
input 4 → [ × 3 ] → output 12
input 7 → [ × 3 ] → output 21
```

Two-step machines

- Two operations, applied in order, left to right.
- Do the first, then the second — never both at once.
- Write the number after each step so nothing is lost.
- The order is part of the rule.

```
input 5 → [ × 2 ] → 10 → [ + 1 ] → 11
input 5 → [ + 1 ] → 6 → [ × 2 ] → 12
different order, different answer
```

Working backwards

- Given the output, find the input by undoing.
- Use the **inverse** of each operation.
- Go in **reverse order** — last operation undone first.
- Like taking off your shoes before your socks.

```
output 25, machine × 3 then + 4
undo + 4: 25 − 4 = 21
undo × 3: 21 ÷ 3 = 7
```

The inverse pairs

- Addition and subtraction undo each other.
- Multiplication and division undo each other.
- That is all you need for Stage 7.
- Write the inverse above each box before you start.

```
+ 3 ↔ − 3
× 5 ↔ ÷ 5
− 7 ↔ + 7
```

Writing it as algebra

- Replace the input with a letter, usually x.
- Apply the operations to that letter.
- The result is the function, written y = …
- Brackets show which operation happens first.

```
× 2 then + 1 → y = 2x + 1
+ 1 then × 2 → y = 2(x + 1)
```

Mapping diagrams

- Two columns: inputs on the left, outputs on the right.
- One arrow from each input to its output.
- Every input has exactly one arrow — that is what makes it a function.
- The pattern of the arrows shows the rule at a glance.

```
1 → 5 rule: y = 4x + 1
2 → 9
3 → 13
```