Lesson Plan: Function Machine

Year Group: 8 | **Duration:** 50 minutes | **Topic:** Functions, Parameters & Return Values

1. Overview

Core Concept: Functions — reusable named blocks that accept inputs (parameters) and produce outputs (return values).

Learning Objectives:

- Define a function as a named, reusable process
- Distinguish between defining and calling a function
- Use parameters to make functions flexible
- Predict return values and chain functions together

Key Vocabulary:

Term	Definition
Function	A named, reusable block that performs a specific task
Parameter	A named input that a function uses
Argument	The actual value passed when a function is called
Return value	The output a function sends back when it finishes
Define	Write the recipe — create the function
Call	Use the recipe — run the function with specific arguments

2. Before the Lesson

Print and Cut:

- ☐ `resource-input-slips.md` — 1 set per group
- ☐ `worksheet-function-design.md` — 1 per student

Teacher Preparation:

- Prepare 3 "secret function" rules on folded paper:
  - Machine 1: `double_and_add_3(x) = 2 × x + 3`
  - Machine 2: `square_it(x) = x × x`
  - Machine 3: `average(a, b) = (a + b) ÷ 2`

3. Timed Lesson Flow

0–5 min — Hook: The Mystery Machine

1. *"I have a function machine. You give me a number. I give you a number back."*
2. Take inputs: 2 → 7, 5 → 13, 0 → 3, 10 → 23. Class guesses the rule.
3. *"The rule stayed the same every time. Only the input changed. That's a function."*

5–10 min — Introduce Functions

1. Reveal: `DEFINE double_and_add_3(x): RETURN 2 × x + 3`
2. Components: name, parameter, rule, return value.
3. *"Defining = writing the recipe. Calling = making the dish."*

10–20 min — Groups Design a Secret Function

Groups secretly design a function: name, parameter(s), rule, 3 test cases.

20–35 min — Feed the Machine!

Groups rotate — other groups feed input slips in, machine writes outputs. After 6 inputs, guess the rule.

35–45 min — Worksheet: Design More Functions

Individual work — two more functions, one with 2 parameters, function chaining.

45–50 min — Debrief: why are functions useful? (reuse, readable names, hide complexity — link to abstraction)

---

4. Teacher Facilitation Notes

What to look for:

- Students confusing defining and calling — *"Are you writing the recipe or making the dish?"*
- Functions that are too vague (e.g., "does maths") — require a specific, computable rule
- Students forgetting parameters — *"What information does your function need?"*

Common misconceptions:

- A function always does the same thing — same PROCESS, different inputs give different outputs
- You need to know how a function works to use it — no, abstraction means you just need name + parameters + return type

---

5. Extension Tasks

1. Chain: `double_and_add_3(square_it(3))` — solve step by step
2. Write a function with no parameters that always returns the same value — when would this be useful?
3. Research: what is a "pure function"? How does it differ from one with side effects?

---

6. Key Takeaway

> **A function is a named, reusable process. Parameters make it flexible — the same function works with different inputs. Define once, call anywhere.**