

Answer key: Function Ladder

INSTRUCTOR COPY

Every check is a call in Spyder's IPython console. Nothing here needs `print`.

Part 1 answers Rungs 1–3

- 1 `lucky()` shows 7.
- 2 `add_one(5)` shows 6; the blank is `return x + 1`. Pupil's own function: any `def` with one parameter and one `return`; accept it when the console value matches what they wrote on the chip. `add_one(10)` shows 11.
- 3 `hello()` shows **nothing**. The console goes to a fresh prompt with no `Out` line: `hello` ran, set `greeting`, gave no value back. Talking point: `return` is the only way a value reaches the console.

```
def lucky():  
    return 7  
  
def add_one(x):  
    return x + 1  
  
def hello():  
    greeting = "hi"
```

```
def average(a, b):  
    return (a + b) / 2  
  
def product(a, b):  
    return a * b
```

Part 2 answers Rungs 4–5: two inputs, then your four

- 4 `average(3, 5)` shows 4.0 (the decimal is right: `/` gives a float). `product(6, 7)` shows 42.
- 5 Model answers, the exact names `calculator.py` calls:

```
def add(a, b):  
    return a + b  
  
def subtract(a, b):  
    return a - b  
  
def multiply(a, b):  
    return a * b  
  
def divide(a, b):  
    return a / b
```

Chips: `add(2, 3)` 5, `subtract(10, 4)` 6, `multiply(6, 7)` 42, `divide(9, 2)` 4.5. Tracker: four ticks only when the console value matches; send back any rung with a mismatch. Gate before `calculator.py`: all four in one file, names exact (`subtract`, `Add`, `multiply` all fail, section 18).

Part 3 answers Rungs 6–7

- 6 `add(multiply(3, 4), 2)` shows 14 (inside first: 12, then 14). `divide(add(6, 8), 2)` shows 7.0 (again the `.0` is correct).
- 7 `power(2, 10)` shows 1024.

```
def power(a, b):  
    return a ** b
```

`divide(5, 0)`: console shows a traceback, exact IPython wording:

```
ZeroDivisionError: division by zero
```

Above it: `Traceback (most recent call last):` and the line reference. Expect the error text; do not teach a fix, if comes later (section 19).

Notes Common mistakes

- **No `return`** : the function runs, the console shows nothing (rung 3). Worse later: `calculator.py` receives `None` .
- **name typos:** `add` , `subtract` , `multiply` , `divide` are spelled by `calculator.py` ; one wrong letter and it finds nothing. Check names before values.
- **Wrong indent:** the `return` must be indented under the `def` . A body that is only a comment (or empty) gives `IndentationError: expected an indented block` ; a blank line is not a body.
- **`return` inside the wrong spot:** code after `return` in the same body never runs, harmless here but visible if they add lines.
- **`print` instead of `return`** : the chip may still show a number, but it is text, not the value; chaining in rung 6 then fails with `TypeError` . `print` is not taught yet; if a pupil found it, redirect to `return` .
- **`//` in `divide`** : gives `4` for `divide(9, 2)`, not `4.5` (`%` gives `1`). Correct operator is `/` .