ANSWERS

0.5 Your own commands

Python quick start · Robot club · about 12 min

What this lesson is about

def, parameters and return: naming a piece of work so the program reads like what it does.

- Naming a piece of work
- Telling it how much
- Why bother

Questions

7 marks in all

1. What happens when Python reads the `def side():` lines? [1 mark]

- ☐ **A** Nothing yet; the lines inside run only when you call side()
- ☐ **B** The robot drives one side straight away
- ☐ **C** The robot drives one side, twice
- ☐ **D** Python gives an error until you call it

> **Answer:** A. `def` only gives a name to some lines. They run each time you call the name with `side()`.

2. What does this program print? [1 mark]

```
def side():
    print("drive")
    print("turn")

side()
side()
```

> **Answer:**
>
> ```
> drive
> turn
> drive
> turn
> ```
>
> Each call to `side()` runs both lines inside it, so two calls print drive, turn, drive, turn.

3. What does this program print? [1 mark]

```
def hello():
    print("hello")

print("start")
```

> **Answer:**
>
> ```
> start
> ```
>
> The function is defined but never called, so its print never runs. Only `print("start")` does.

4. What does this program print? [1 mark]

```
def side(cm):
    print("forward", cm)

def square(cm):
    for i in range(4):
        side(cm)

square(20)
```

Answer:

```
forward 20
forward 20
forward 20
forward 20
```

`square(20)` passes 20 in as `cm`, and the loop calls `side(20)` four times, one line each.

5. What does this program print? [1 mark]

```
def gap_now(reading):
    return round(reading)

print("the gap is", gap_now(27.6), "cm")
```

Answer:

```
the gap is 28 cm
```

`return` hands a value back to where the function was called. `round(27.6)` is 28, so that is what gets printed.

6. In `def side(cm):`, what is `cm` called? [1 mark]

Answer: parameter. A parameter is a name that stands for whatever the caller passes in, so `side(40)` makes `cm` 40.

7. Why is it worth putting a square into a `square(cm)` function? Choose all that apply. [1 mark]

Tick every answer that is true.

- ☐ **A** The program says what it does
- ☐ **B** A fix happens in one place, not four
- ☐ **C** You can try that piece on its own
- ☐ **D** It makes the robot drive more accurately
- ☐ **E** It makes the program run faster

Answer: A, B, C. Functions help people read, fix and test a program. The robot drives exactly the same either way.

The task: draw an L

Write a function `leg(cm)` that drives that far forward and then turns right, and use it twice to walk the robot through both green squares: 40 cm, corner, 30 cm.

```
from bugbot import *
connect()

def leg(cm):
    # drive, then turn
    forward(50, distance=cm)

leg(40)
stop()
```

The hint students can ask for: leg(cm) drives that far and then turns right. Call it with 40, then with 30.

A solution

```
from bugbot import *
connect()

def leg(cm):
    forward(50, distance=cm)
    turn_right(35, angle=90)

leg(40)
leg(30)
stop()
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.