



0.5 Your own commands

Python quick start · Robot club · about 12 min

Name _____

Class _____

Date _____

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

2. What does this program print?

[1 mark]

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

3. What does this program print?

[1 mark]

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

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)
```

5. What does this program print?

[1 mark]

```
def gap_now(reading):  
    return round(reading)  
  
print("the gap is", gap_now(27.6), "cm")
```

6. In `def side(cm):`, what is `cm` called?

[1 mark]

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

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()
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/start-5-functions/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Add a `flash()` function that blinks the light, and call it at each corner.
2. Give `leg` a second parameter for the speed: `leg(40, 30)`.
3. Write `triangle(cm)`, and work out the angle inside the function rather than in your head.