



9.3 Protocols

Talking to each other · Robot club · about 20 min

Name _____

Class _____

Date _____

What this lesson is about

Agreeing what a message looks like; splitting it into words and numbers.

- Asking and waiting
- Taking a message apart
- Strict out, forgiving in

Questions

7 marks in all

1. What does this program print?

[1 mark]

```
words = "forward 30".split()
command, amount = words[0], float(words[1])
print(command, amount * 2)
```

2. What does this program print?

[1 mark]

```
def numbers_in(text):
    out = []
    for word in text.replace(",", " ").split():
        try:
            out.append(float(word))
        except ValueError:
            pass
    return out

print(numbers_in("ball at -12,40 ok"))
```

3. What does this program print?

[1 mark]

```
text = " Forward 30 "
words = text.lower().strip().split()
print(words)
```

4. Why does `ask` give up after a second instead of waiting for a reply forever?

[1 mark]

- ☐ A Radios miss things, and a program waiting for a reply that never comes is stuck
- ☐ B The Guide only speaks for one second
- ☐ C `messages()` stops working after a second
- ☐ D To save battery

5. A message arrives saying `jump 10`, which your program does not understand. What should it do? [1 mark]
- ☐ A Ignore it and carry on
 - ☐ B Stop with an error so you notice
 - ☐ C Treat it as forward 10
 - ☐ D Send it back to whoever sent it

6. `ask("next")` returns `None`. What does that mean? [1 mark]
- ☐ A No reply arrived within the time allowed
 - ☐ B The Guide said none
 - ☐ C The robot has reached the green zone
 - ☐ D The message was sent to nobody

7. What is the word for the agreement, made in advance, about what messages look like? [1 mark]

The task: follow instructions

Ask the Guide for instructions one at a time, do each one, and stop at `done` . You will end up in the green zone.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()
def ask(question, seconds=1.0):
    # send a question and return the first reply that arrives within `seconds`, or None
    send(question)
    for tick in range(int(seconds * 10)):
        # pause 0.1 s (the robot keeps doing what it was told)
        wait(0.1)
        for sender, text in messages():
            return text
    return None

def numbers_in(text):
    # every number in a message, as floats: "at 25,55" -> [25.0, 55.0]
    out = []
    for word in text.replace(",", " ").split():
        try:
            out.append(float(word))
        except ValueError:
            pass
    return out
reply = ask('next')
print('guide says:', reply)
```

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



Do it on the robot

www.bugbotlab.com/learn/9-3-protocols/

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

Challenges

1. Handle `left <cm>` and `backward <cm>` too, even though this Guide never sends them.
2. Handle `turn <degrees>`.
3. If `ask` returns `None` , try again, up to three times, then give up and print why.