



5.5 Brains

Name _____

Class _____

Date _____

What this lesson is about

From a loop to think(me) and me.memory, with a practice arena.

- The loop is outside
- Escape, as a loop
- The same thing as a brain

Questions

6 marks in all

1. How often does the arena call your think(me) function? [1 mark]
☐ A Ten times a second
☐ B Once, at the start of the game
☐ C Only when another robot comes close
☐ D As fast as the computer can
2. Why does a brain keep its state in me.memory rather than in a normal variable? [1 mark]
☐ A Local variables are forgotten when the function returns
☐ B me.memory is faster
☐ C Normal variables cannot hold words
☐ D The arena reads me.memory to score you
3. What does this program print? [1 mark]

```
class Me:
    def __init__(self):
        self.memory = {}

    def think(me):
        count = me.memory.get("count", 0)
        me.memory["count"] = count + 1

    def forgetful(me):
        count = 0
        count += 1
        return count

me = Me()
for i in range(3):
    think(me)
    n = forgetful(me)
    print(me.memory["count"])
    print(n)
```

4. On the very first tick, `me.memory` is empty. What does `me.memory.get("state", "go")` give? [1 mark]
- ☐ A `"go"`, the default
 - ☐ B `None`
 - ☐ C An error, because the key is missing
 - ☐ D An empty string

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

```
samples = [(20, 0), (25, 45), (60, 90), (80, 135), (70, 180), (30, 225), (20, 270)]
opens = [h for d, h in samples if d > 40]
target = opens[len(opens) // 2]
print(opens, target)
```

6. The escape program aims at the middle of the open headings, not the single longest reading. Why? [1 mark]
- ☐ A The longest reading points at a corner of the opening
 - ☐ B The longest reading is always wrong
 - ☐ C The middle is quicker to work out
 - ☐ D The distance sensor cannot read long distances

The task: escape the box

Three walls around you. Find the gap with the distance sensor, drive out and into the green zone, touching nothing.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

# as long as the thing ahead is further than 15 cm
while distance() > 15:
    # drive forward at 50 (keeps going until the next command)
    forward(50)
    # pause 0.1 s (the robot keeps doing what it was told)
    wait(0.1)
# all motors off
stop()
print("wall at", distance())
```

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



Do it on the robot

www.bugbotlab.com/learn/5-5-brains/

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

Challenges

1. Escape faster: stop looking as soon as you have seen a reading over 40 and the readings are falling again.
2. In the arena, add a third state: "shove", entered when another robot is within 10 cm, that drives at it for five ticks.
3. Write the escape as a brain (`think(me)` with `me.memory`) on paper. What changes?