



F3.6 Records

Strings, lists and records · GCSE · OCR J277 2.2.3, AQA 8525 3.2.6, Edexcel

1CP2 6.3.1 · about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Grouping related fields: a sensor reading as a record, and a list of records as a table.

- A record as a dictionary
- Searching a table of records
- A list of records
- Records as their own type

Questions

5 marks in all

1. What makes a record different from an array?

[1 mark]

- ☐ **A** A record's fields can hold different data types, reached by name
- ☐ **B** A record can only hold numbers
- ☐ **C** A record is always sorted
- ☐ **D** An array can hold different types, a record cannot

2. What does this program print?

[1 mark]

```
reading = {"heading": 90, "distance": 51.0}
reading["heading"] = 180
print(reading["heading"], reading["distance"])
```

3. What does this program print?

[1 mark]

```
log = [{"h": 0, "d": 31}, {"h": 90, "d": 51}, {"h": 180, "d": 15}]
best = log[0]
for r in log:
    if r["d"] > best["d"]:
        best = r
print(best["h"])
```

4. A robot log stores a time, a heading, a distance and a note for each reading. Which data types suit the fields?

[1 mark]

Tick every answer that is true.

- ☐ **A** time: real
- ☐ **B** heading: integer
- ☐ **C** distance: real
- ☐ **D** note: string

5. In a table of records, what is each row?

[1 mark]

- ☐ A One record
- ☐ B One field
- ☐ C One data type
- ☐ D One array index

The task: reading records

The robot is boxed in on three sides. Look in four directions, 90 degrees apart, storing each look as a record with the fields `heading` and `distance` in a list. Then print one line per record as `heading 0: 31.0 cm`, and finally print `closest heading: <heading>` for the record with the smallest distance.

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

log = []
for i in range(4):
    turn_right(30, angle=90)
```

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



Do it on the robot

www.bugbotlab.com/learn/f3-6-records/

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

Challenges

1. Add a `time` field to each record using `clock()`.
2. Print the log as a table, with the headings lined up using f-string widths like `{r['distance']:6}`.
3. Store three names and scores as records and print the name with the highest score.