



A11.8 Capturing, managing and exchanging data

Databases and big data · A level · OCR H446 1.3.2, AQA 7517 4.9.4.10,
Eduqas A500QS 2.4 · about 55 min

Name _____

Class _____

Date _____

What this lesson is about

How data gets in, how it is selected and managed, and how systems swap it as CSV, JSON and XML. The robot exports its own readings.

- Capturing data
- Managing data
- Selecting data
- Exchanging data

Questions

6 marks in all

- Exam answer sheets are scanned and the positions of shaded boxes are read. Which capture method is this? [1 mark]
 - ☐ A Optical mark recognition (OMR)
 - ☐ B Optical character recognition (OCR)
 - ☐ C RFID
 - ☐ D A data logger
- A run time of 7.2 s is typed as 2.7 s. Which check could catch this? [1 mark]
 - ☐ A Verification, such as double entry
 - ☐ B A range check
 - ☐ C A presence check
 - ☐ D A format check
- What does a data dictionary hold? [1 mark]
 - ☐ A Data about the data: tables, fields, types, lengths, keys and validation rules
 - ☐ B Every record in the database, sorted
 - ☐ C A list of the words used in text fields
 - ☐ D Backups of deleted records
- Which are true of JSON compared with XML? [1 mark]

Tick every answer that is true.

 - ☐ A JSON is usually more compact
 - ☐ B JSON has types such as numbers, true and false
 - ☐ C Only XML can represent nested data
 - ☐ D JSON requires every value to be wrapped in opening and closing tags

5. What does this print?

[1 mark]

```
import json
text = '{"robot": "Ada", "readings": [61, 51, 41], "ok": true}'
data = json.loads(text)
print(data["robot"], len(data["readings"]), data["readings"][-1] + 1, data["ok"])
```

6. What does this print?

[1 mark]

```
import csv, io
rows = list(csv.reader(io.StringIO("name,cm\nAda,42\nBolt,38\n")))
print(rows[1])
print(rows[1][1] + rows[2][1])
```

The task: three formats, one set of readings

The robot starts facing a wall. Take four readings, numbered `step` 0 to 3. For each: record `step` (a whole number), `y` (the second value of `position()`, rounded to 1 decimal place) and `cm` (the value of `distance()`), then drive forward 10 cm at speed 40 before the next reading (no drive after step 3). Then, from that one list of readings: 1. Write `readings.csv`: the header line `step,y,cm`, then one line per reading, such as `1,10.0,51.0`. 2. Write `readings.json` with `json.dump`: a list of four objects, each with the keys `step`, `y` and `cm`. 3. Read `readings.json` back with `json.load` and print `json readings: <n>`, the number of objects in it. 4. Print the readings as XML: a line `<readings>`, then one line per reading exactly in the form `<reading step="1" y="10.0" cm="51.0"/>` (indenting is allowed), then `</readings>`.

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

import json

readings = []
```

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



Do it on the robot

www.bugbotlab.com/learn/a11-8-capturing-and-exchanging-data/

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

Challenges

1. Add a `seconds` field, the time of each reading from `clock()`. Which of your three outputs needed changing, and how?
2. Nest the readings inside a JSON object that also records the robot's name and the date. Could CSV hold the same thing?
3. A robot's name is `Ada`, the `second`. What goes wrong in a CSV file written by hand, and how does the `csv` module deal with it?