



F3.7 Random numbers

Strings, lists and records · GCSE · OCR J277 2.2.3, AQA 8525 3.2.9, Edexcel

1CP2 6.6.1 · about 12 min

What this lesson is about

randint, choice and seeds: dice, random tunes and a robot that wanders.

- randint
- Other random choices
- A robot that wanders
- Checking your odds
- Random, but the same every time

Questions

5 marks in all

1. Which values can `random.randint(1, 6)` give?

[1 mark]

- ☐ A 1, 2, 3, 4, 5 or 6
- ☐ B 1 to 5
- ☐ C 0 to 6
- ☐ D Any real number from 1 to 6

Answer: A. randint includes both ends, unlike range.

2. What does `random.choice(["red", "green", "blue"])` give?

[1 mark]

- ☐ A One of the three colours, picked at random
- ☐ B All three in a random order
- ☐ C The number 3
- ☐ D Always red

Answer: A. choice picks one item from a list.

3. What does setting `random.seed(42)` before generating numbers do?

[1 mark]

- ☐ A Makes the same sequence of numbers come out every run
- ☐ B Makes the numbers more random
- ☐ C Limits numbers to 42
- ☐ D Stops random numbers

Answer: A. Computers make pseudo-random numbers with a formula; the same seed gives the same sequence, which helps testing.

4. Write the AQA pseudo-code call for a random whole number from 1 to 10.

[1 mark]

Answer: `RANDOM_INT(1, 10)`. AQA's `RANDOM_INT` includes both ends.

5. A dice program counts 600 rolls. About how many sixes should it see?

[1 mark]

- ☐ A About 100, but rarely exactly 100
- ☐ B Exactly 100
- ☐ C About 600
- ☐ D About 6

Answer: A. Each face has a 1 in 6 chance, so around 100, with some variation each run.

The task: dice drive

Roll a dice three times with `random.randint(1, 6)`. For each roll, print `roll: <n>` and drive forward `<n> * 5` cm. At the end print `total: <cm>` with the total distance, worked out from the rolls. This task fixes its random numbers, so you get the same rolls each run.

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

import random

total = 0
```

The hint students can ask for: Roll inside the loop, print the roll, then drive a distance worked out from it. Keep a running total as you go.

A solution

```
from bugbot import *
connect()
import random

total = 0
for i in range(3):
    roll = random.randint(1, 6)
    print("roll:", roll)
    forward(50, distance=roll * 5)
    total = total + roll * 5
print("total:", total)
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.