ANSWERS

F10.6 The internet, DNS and the cloud

Networks · GCSE · OCR J277 1.3.1, Edexcel 1CP2 4.1.3 · about 20 min

What this lesson is about

The internet and the web, DNS, web servers and hosting, and the cloud.

- DNS: names to addresses
- Web servers and clients
- The cloud
- A DNS server on the mat

Questions

5 marks in all

1. What does DNS do? [1 mark]

- ☐ **A** Turns a domain name into an IP address
- ☐ **B** Encrypts web pages
- ☐ **C** Stores websites
- ☐ **D** Connects a LAN to the internet

> **Answer:** A. The browser then uses the IP address to contact the server.

2. Put the steps in order. [1 mark]

Number the lines 1 to 4 to put them in the right order.

___ `The user types a URL into the browser`

___ `The browser asks a DNS server for the IP address`

___ `The DNS server returns the IP address`

___ `The browser requests the page from the web server`

> **Answer:**
>
> ```
> The user types a URL into the browser
> The browser asks a DNS server for the IP address
> The DNS server returns the IP address
> The browser requests the page from the web server
> ```
>
> Name to address, then address to page.

3. What is the difference between the internet and the World Wide Web? [1 mark]

- ☐ **A** The internet is the network; the web is a service of pages that runs on it
- ☐ **B** They are the same thing
- ☐ **C** The web is the cables; the internet is the pages
- ☐ **D** The internet is only for email

> **Answer:** A. Email and games use the internet but are not the web.

4. What is web hosting? [1 mark]

- ☐ **A** Storing a website on a server that is always connected to the internet
- ☐ **B** Visiting a website
- ☐ **C** Designing a website
- ☐ **D** Searching the web

> **Answer:** A. Usually rented from a hosting company.

5. Which is a disadvantage of the cloud? [1 mark]

- ☐ **A** It needs an internet connection
- ☐ **B** Files can be reached from any device
- ☐ **C** The provider handles backups
- ☐ **D** Storage can grow as needed

> **Answer:** A. The others are advantages.

The task: look it up

Two robots are on the mat: `scout.bot` and `rover.bot`. Look up `scout.bot` with the DNS robot, then take the IP address from its reply. Send `to <address>: where` and the Scout answers `at <x>,<y>`. Drive to about 13 cm in front of it, into the green square, without touching anything. The helpers `ask`, `numbers_in`, `go_to` and `near` are ready to use.

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

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)):
        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

def wrapped(h):
    return (h + 180) % 360 - 180

def go_to(x, y, speed=60):
    # one step of driving towards (x, y), measured from the start; call it in a loop
    px, py = position()
    a = math.radians(wrapped(math.degrees(math.atan2(x - px, y - py)) - heading()))
    drive(speed * math.cos(a), speed * math.sin(a), wrapped(0 - heading()) * 3)

def near(x, y, cm=4):
    px, py = position()
    return math.hypot(x - px, y - py) < cm
```

The hint students can ask for: Ask the DNS robot first and pull the address out of its reply. Use that address to ask the Scout where it is, take the two numbers out of its answer, and drive to a point short of it so you stop in the square without touching.

A solution

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

def ask(question, seconds=1.0):
    send(question)
    for tick in range(int(seconds * 10)):
```

```
        wait(0.1)
        for sender, text in messages():
            return text
    return None

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

def wrapped(h):
    return (h + 180) % 360 - 180

def go_to(x, y, speed=60):
    px, py = position()
    a = math.radians(wrapped(math.degrees(math.atan2(x - px, y - py)) - heading()))
    drive(speed * math.cos(a), speed * math.sin(a), wrapped(0 - heading()) * 3)

def near(x, y, cm=4):
    px, py = position()
    return math.hypot(x - px, y - py) < cm

reply = ask("lookup scout.bot")
address = reply.split(" = ")[1]
x, y = numbers_in(ask("to " + address + ": where"))
while not near(x, y - 13):
    go_to(x, y - 13)
    wait(0.1)
stop()
```

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