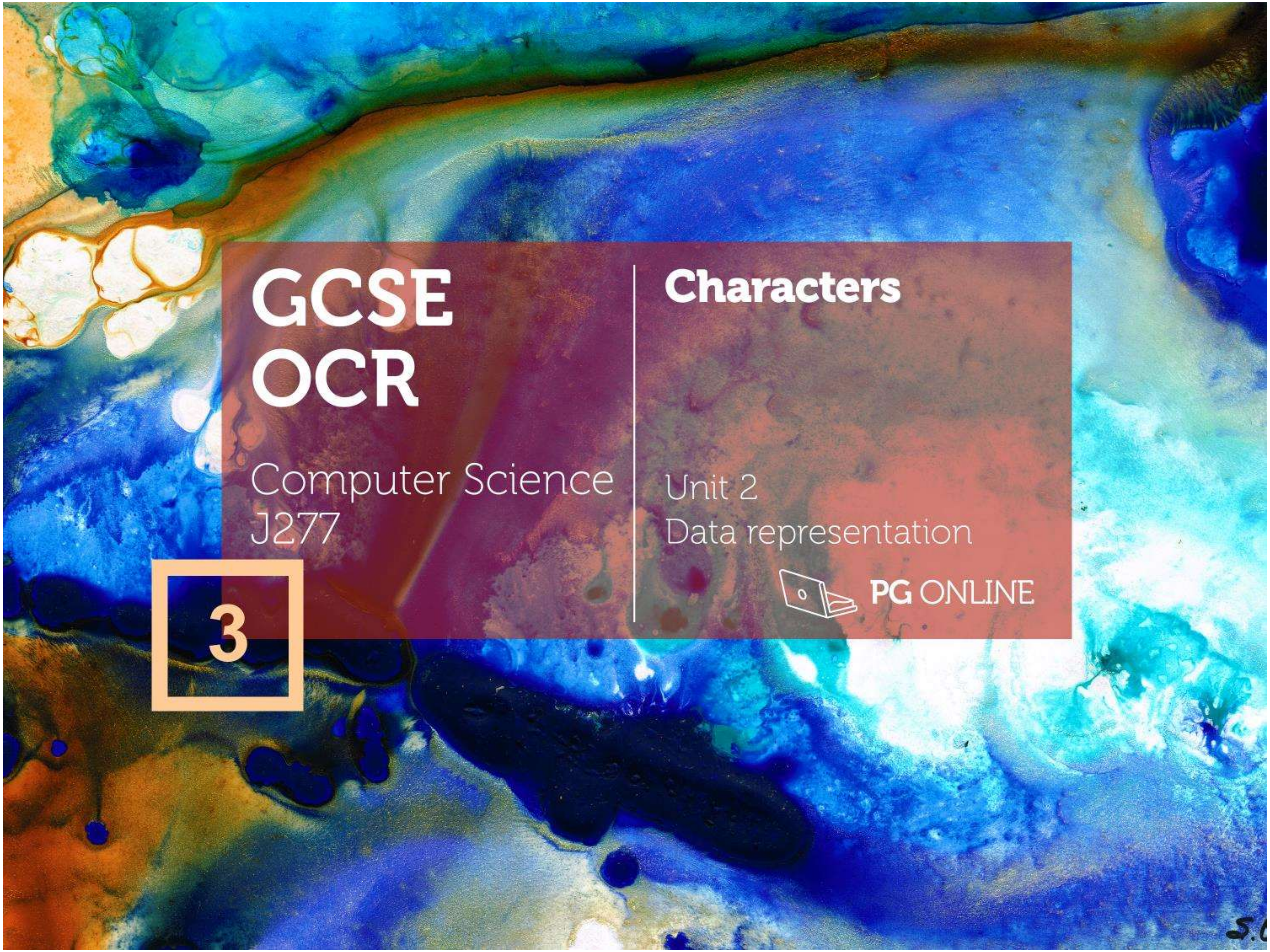


GCSE OCR

Computer Science
J277

3

Characters

Unit 2
Data representation

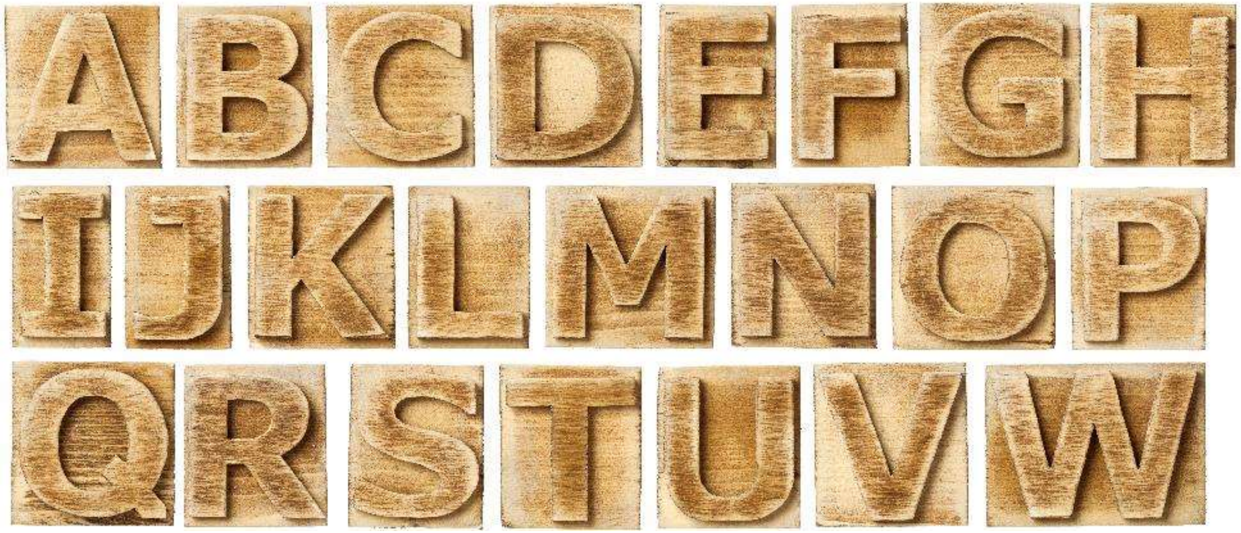
PG ONLINE

Objectives

- Understand the use of binary codes to represent characters
- Understand the term ‘character set’
- Explain the relationship between the number of bits per character in a character set, and the number of characters that can be represented using:
  - ASCII
  - Extended ASCII
  - Unicode

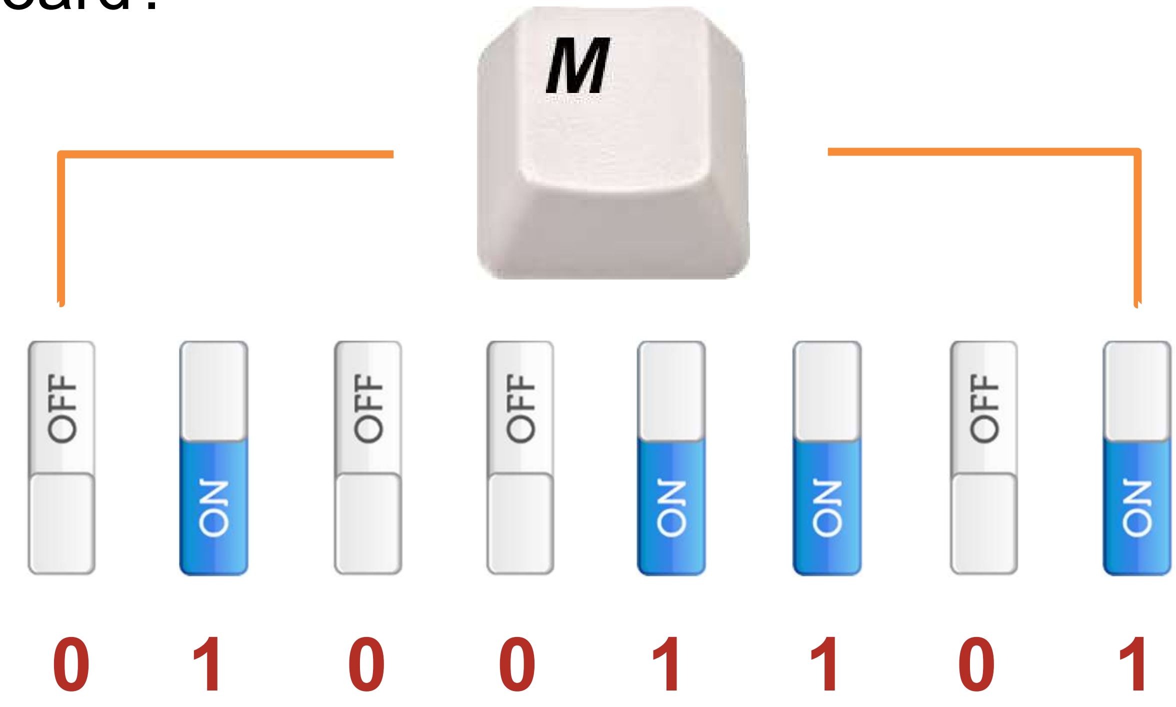
Starter

- A computers memory and storage only hold binary 1s and 0s
  - How might it be possible to store letters with only binary?



Representing text characters

- If a computer understands only 1s and 0s, what happens when the ‘M’ key is pressed on the keyboard?



Representing characters in binary

- Every character on the keyboard is represented by a binary value
  - Uppercase letters (capitals) have different values from lowercase characters
  - Punctuation symbols have their own character code
- How many characters are there on a standard keyboard?
  - How many bits would be required to represent this many combinations?

Characters in binary

Answers

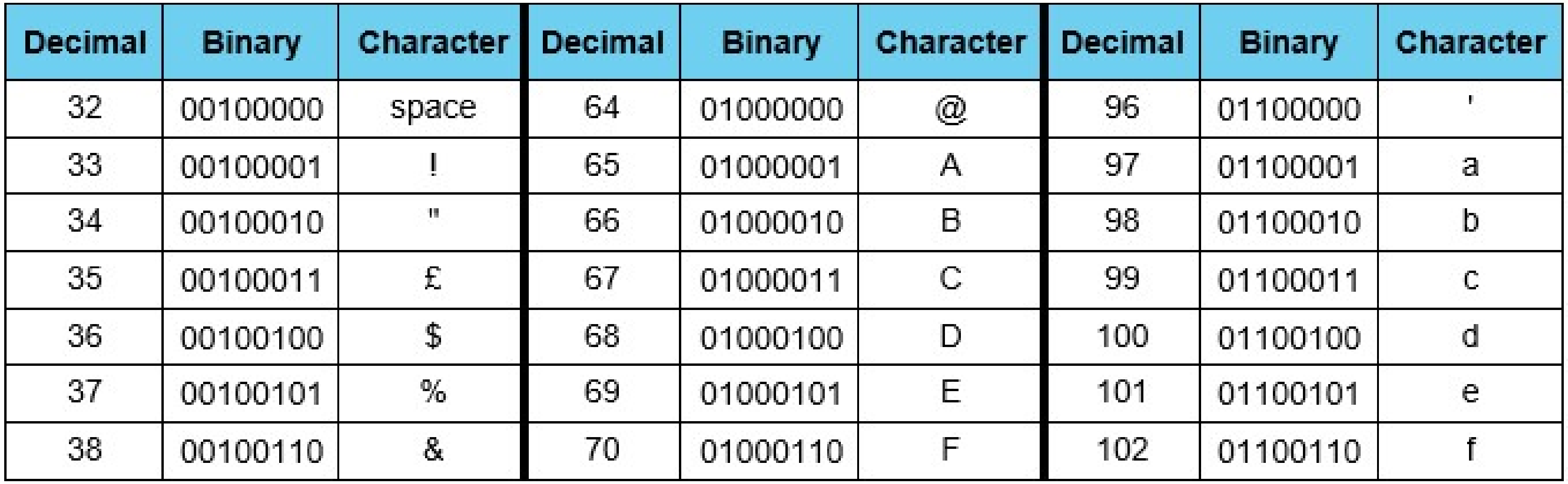
- A keyboard needs to contain
  - 26 lowercase letters
  - 26 uppercase letters
  - 10 numbers
  - (around) 36 other characters
- There are around 98 unique characters that are available on a keyboard
  - 6 bits give 64 different combinations – this isn't enough
  - 7 bits give 128 different combinations which can represent 128 different characters

The ASCII code

- **ASCII** (American Standard Code for Information Interchange) has become the standard code, used worldwide
  - It was originally developed in the 1960s for representing the English alphabet
  - It encodes 128 characters into 7-bit binary codes
- Characters include numbers 0 to 9, uppercase and lowercase letters A-Z, a-z, punctuation symbols and the space character

The ASCII character set

- What happens if you press ***ALT+65*** on a keyboard?
  - What character is represented by 0100000 (32)?
- What is the ASCII character for the number 7? Is this the same as the binary value for 7?
  - Why not? What is happening? What does this mean?



Decimal	Binary	Character	Decimal	Binary	Character	Decimal	Binary	Character
32	00100000	space	64	01000000	@	96	01100000	'
33	00100001	!	65	01000001	A	97	01100001	a
34	00100010	"	66	01000010	B	98	01100010	b
35	00100011	£	67	01000011	C	99	01100011	c
36	00100100	$	68	01000100	D	100	01100100	d
37	00100101	%	69	01000101	E	101	01100101	e
38	00100110	&	70	01000110	F	102	01100110	f

ASCII groups and sequences

- Character codes are commonly grouped and run in sequence
  - Numeric characters 0 to 9 run consecutively from 48 to 57 on the ASCII table
- A-Z characters are from 65-90 or 01000001 to 01011010
  - What range does lowercase characters a-z use?
  - If you know Capital A is 65 or 01000001, what is Capital E?

ASCII character set

Answers

- ASCII character 32 (010 0000) represents a space
- The ASCII character code for '7' is 55
  - 55 (011 0111) is the ASCII character code that represents the character '7'
  - In programming this is very different to the integer 7 which is represented by 0000 0111 (7)
- Lowercase characters a-z use 97-122
- If A is 65 (0100 0001) then E is 69 (0100 0101)

7- and 8-bit ASCII

- Numerous different codes for representing characters have been created, but ASCII is commonly used on PCs
- Originally only seven bits were used, but now an eighth bit is used allowing for many more characters such as ©, ® etc.
  - How many different characters can be encoded using 7 bits, 8 bits or 16 bits?

Character codes

Answers

- A 7-bit character code (like ASCII) has 128 different characters that can be encoded
  - An 8-bit character code (like extended ASCII) has 256 different characters that can be encoded
  - A 16-bit character code has 65 536 different characters that can be encoded

Using the eighth bit

- Sometimes it is useful to be able to type special characters like á, à, ®
- Here are the codes for some of them:

©	Alt+0169
®	Alt+0174
á	Alt+0225
à	Alt+0224
â	Alt+0226
ä	Alt+0228

- Try out these different character codes

Worksheet 3

- Complete **Task 1** and **Task 2** on **Worksheet 3**

Programming with text and numbers

- The ASCII code for ‘7’ is 011 0111
    - The binary code for the digit 7 is 0000 0111
- When you write a program in Python, for example, you have to specify whether a variable is text or integer
    - You cannot do arithmetic with characters
    - If the character represents a number it must first be converted to an integer before any arithmetic can be carried out

Working with string input

- In Python, two strings can be concatenated, or joined together, using the + symbol

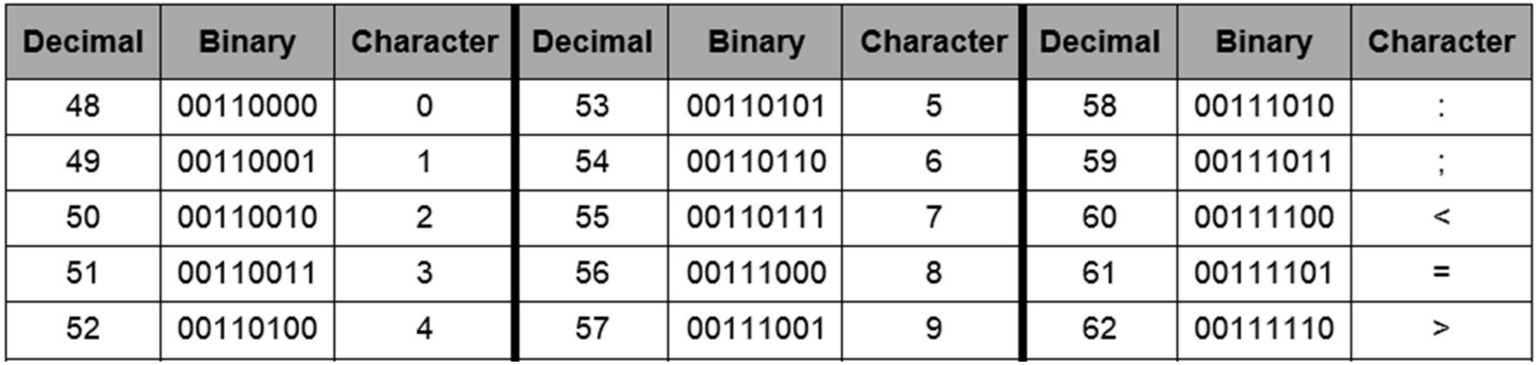
```
firstname = input("Please input your first name: ")
secondname = input("Please input your second name: ")
fullname = firstname + " " + secondname
print("Your full name is " + fullname)
```

- If you enter **Mike** for a first name and **Bell** for a second name, the computer will display

```
Your full name is Mike Bell
```

ASCII representation of numbers

- Try typing ***ALT + 55***
- What is the binary representation of the ASCII character 7? Is this the same as the binary value for 7?
  - Why not? What does this mean?



Decimal	Binary	Character	Decimal	Binary	Character	Decimal	Binary	Character
48	00110000	0	53	00110101	5	58	00111010	:
49	00110001	1	54	00110110	6	59	00111011	;
50	00110010	2	55	00110111	7	60	00111100	<
51	00110011	3	56	00111000	8	61	00111101	=
52	00110100	4	57	00111001	9	62	00111110	>

Converting ASCII to pure binary

- Clearly, we cannot do arithmetic with ASCII characters
- Programming languages deal with the input of numbers in different ways
- In some languages, variables have to be declared as type char, string, integer, real etc. at the beginning of the program
  - In other languages such as Python, all data is input as string, and if it is to be regarded as an integer, it has to be converted using an inbuilt function

e.g.

```
xString = input (“Enter an integer: ”)
x = int(xString)
```

Using different alphabets

- To represent other characters for different languages, a new code allowing for many more characters is needed
  - **Unicode** was developed to use 16 bits 65 536 possible combinations
  - The 32 bit version gives 4 294 967 296 (over 4 billion) possible combinations

Unicode

- In Japanese, konnichiwa is used as a greeting meaning “good day”
- In Unicode this is written as three 16-bit characters

今日は

- How many bytes does the English ‘good day’ require in ASCII?
- How many bytes does the Japanese require in Unicode?

Unicode

Answers

- ‘good day’ requires 8 bytes to store
- 今日は requires 6 bytes to store (3 characters x 2 bytes)
- Unicode is also used to store emoji
  - ‘e’ is Japanese for picture
  - ‘moji’ is Japanese for character or alphabet

Smiling face with sunglasses
Unicode: 1F60E

Worksheet 3

- Complete **Task 3** on **Worksheet 3**

Plenary

- Work in a pair to answer the following questions
  - How many bits are in extended ASCII?
  - How many characters does this allow for?
  - How many bytes are in Unicode?
  - If ‘f’ has the ASCII code 102, what is the ASCII code for ‘g’?
  - How many bytes are needed to store “Hello everyone.”?

Plenary

Answers

- How many bits are in extended ASCII? 8 bits
- How many characters does this allow for? 256
- How many bytes are in Unicode?
  16-bit has 2 bytes, 32-bit has 4 bytes
- If ‘f’ has the ASCII code 102, what is the ASCII code for ‘g’?
  103
- How many bytes are needed to store “Hello everyone.”?
  15 letters (remember space and the full stop)

Copyright

© 2020 PG Online Limited

The contents of this unit are protected by copyright.

This unit and all the worksheets, PowerPoint presentations, teaching guides and other associated files distributed with it are supplied to you by PG Online Limited under licence and may be used and copied by you only in accordance with the terms of the licence. Except as expressly permitted by the licence, no part of the materials distributed with this unit may be used, reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic or otherwise, without the prior written permission of PG Online Limited.

Licence agreement

This is a legal agreement between you, the end user, and PG Online Limited. This unit and all the worksheets, PowerPoint presentations, teaching guides and other associated files distributed with it is licensed, not sold, to you by PG Online Limited for use under the terms of the licence.

The materials distributed with this unit may be freely copied and used by members of a single institution on a single site only. You are not permitted to share in any way any of the materials or part of the materials with any third party, including users on another site or individuals who are members of a separate institution. You acknowledge that the materials must remain with you, the licencing institution, and no part of the materials may be transferred to another institution. You also agree not to procure, authorise, encourage, facilitate or enable any third party to reproduce these materials in whole or in part without the prior permission of PG Online Limited.

PG ONLINE