

GCSE

Practical
programming
skills in Python

Regular expressions

Topic 4



PG ONLINE

4

Objectives

- Review the purpose of **validation**, one of the programming techniques that may be helpful to complete the controlled assessment tasks
- Understand the purpose of a **regular expression**
- Be able to use a regular expression to validate an input

Validation

- Validation is the process of checking data as it is input to make sure it meets certain rules
- You've probably already done some validation with simple numbers:
 - e.g. pupil age must be less than 16
 - e.g. pupil height cannot be more than 2m

Validating patterns

- A **Regular Expression** is a tool you can use to check that an input from a user matches a particular pattern
 - e.g. a credit card number must be 16 digits long
 - e.g. a phone number must start with a 0
 - e.g. a postcode must go LLNN NLL
(where L is a letter, N is a number)
- Can you think of some other input data that always follows a certain pattern?

A simple regular expression

- Try the following code:

```
import re
name = input("Enter your name: ")

valid = re.match("[A-Z]", name)

if valid:
    print("That looks OK")

else:
    print("Invalid, no capital")
```

A simple regular expression

- The program on the previous slide will check to see if the name the user entered starts with an uppercase letter (`[A-Z]`)
- If the user typed a name that started with a lowercase letter then the user would see the message:

`“Invalid, no capital”`

Regular expression syntax

- This line loads the regular expression library:

```
import re
```

- This line checks to see if the first characters of the name matches the pattern (`[A-Z]`)

```
valid = re.match("[A-Z]", name)
```

Uppercase and lowercase

- You can check that whatever the user types in should start with a lowercase letter (`[a-z]` or an uppercase letter `[A-Z]`)
- You can also check for digits using `[0-9]`

Worksheet 4

- Complete **Questions 1 and 2**



Postcodes

- Postcodes have to fit a certain pattern
 - UK postcodes are between 6 and 8 characters long
 - All UK postcodes include letters and numbers
 - The first 2 letters indicate the geographical area of the postcode:
 - W1 8BL is in West London
 - CB7 8LY is in Cambridge
 - NE15 6BN is in Newcastle
- What other rules does a postcode follow?

Postcode formats

- All postcodes are in one of the following formats:

Format	Example
LN NLL	M1 1AA
LNN NLL	M60 1NW
LLN NLL	CR2 6XH
LLNN NLL	DN55 1PT
LNL NLL	W1A 1HQ
LLNL NLL	EC1A 1BB

- We will build up our validation rule step by step

Matching codes

- Try this code:

```
import re
code = input("Enter your postcode: ")
valid = re.match("[A-Z][A-Z][0-9]", code)
if valid:
    print("That looks OK")
else:
    print("Erm, try again! ")
```

Matching codes

- This program will check that the data entered by the user goes Letter-Letter-Number

i.e. `[A-Z][A-Z][0-9]`

- It is important **not** to add any spaces as the regular expression will try to match the pattern exactly

Regular expression codes

- We can add more detail to our rule by simply adding more characters.
- `[A-Z]` – A capital letter
- `[a-z]` – A lower case letter
- `[0-9]` – A digit
- `m` – A specific letter (in this case, 'm')
- `[a-zA-Z]` – Any letter (upper or lower case)

Allowing multiple characters

- Adding a '+' symbol lets you allow multiple instances of that character
- `[A-Z]+` means at least 1 upper case letter:
e.g. "A", "ASDF" or "QWERTYUIOP"
- `[A-Z]+[0-9]` means at least one letter and then a number:
e.g. "U2" or "SHED7"
- `[A-Z]+[0-9]+` means at least one letter and at least one number:
e.g. "U2", "SHED7", "UB40", "LEVEL42" or "HAIRCUT100"

Allowing multiple characters

- Try this code:

```
import re

code= input("Enter your postcode: ")

valid = re.match("[A-Z]+[0-9]",code)

if valid:

    print("That looks OK")

else:

    print("Erm, try again! ")
```

Worksheet 4

- Complete **Question 3**



Plenary

- Which of these postcodes would be allowed with the following rule?

$[A-Z]^+[0-9]^+ [0-9][A-Z]^+$

Postcode	Accepted?
TSB16 1DA	
Cb23 9GF	
W18BL	
NH54 1QNM	
T16 S1DA	

Plenary

- Which of these postcodes would be allowed with the following rule?

$[A-Z]^+[0-9]^+ [0-9][A-Z]^+$

Postcode	Accepted?
TSB16 1DA	Yes
Cb23 9GF	No: Lower case letter
W18BL	No: No space
NH54 1QNM	Yes
T16 S1DA	No: A letter after space

Worksheet 4

- Have a look at the extension activity **Question 4**
 - Write the program to fully test a postcode for validity

Copyright

© 2017 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.