# Worksheet 4 Regular expressions

1. **A simple Regular Expression**

Try the following code.   
  
If the user enters a name starting with an uppercase letter the program should say “That looks OK”. If the user enters any other name, the program should say “Invalid, no capital”.

import re  
 name = input(“Enter your name: ”)  
 valid = re.match(“[A-Z]”,name)  
 if valid:  
 print(“That looks OK”)  
 else:  
 print(“Invalid, no capital!”)

**Extension**: Try adding a loop so that the user will keep being asked to re-enter their name until they do it right

Now alter the program so that it asks for the user’s home town and checks that it starts with a capital letter.

1. **Checking for lower case letters or digits**

(a) Try the following code.   
  
If the user enters an email address that starts with a lower case letter the program should say “That looks OK”. If the user enters an email address that starts with an upper case letter, number or punctuation mark then it should say “Erm, try again!”.

import re  
 email = input(“Enter your email address: ”)  
 valid = re.match(“[a-z]”,email)  
 if valid:  
 print(“That looks OK”)  
 else:  
 print(“Erm, try again!”)

(b) Now write a program that asks for an input that starts with a number (perhaps a phone number).

|  |  |
| --- | --- |
| **Expression** | **Description** |
| [A-R] | Uppercase letter from A to R |
| [A-Z]{3} | Exactly three uppercase letters from A to Z |
| [A-Z]{1,2} | One or two uppercase letters from A to Z |
| [0-9]{3,} | Three or more digits |
| [0-5] | Digit between 0 and 5 |
| \d | Digit |
| [a-z]+ | One or more lowercase characters |
| [a-z]\* | Zero or more lowercase characters |
| [a-z]? | 0 or 1 lowercase characters |
| \s | space (e.g. [A-Z]\s\d{1,2} will find B 56 and B 1 valid, B52 and BN 3 invalid |
| A|B | A or B |

1. **A complex Regular Expression**

This program accepts a postcode that starts with **two** capital letters and then a number.

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!")

(a) Alter the program so that it accepts postcodes with one or two letters and one or two numbers at the start, followed by a space and one number followed by two letters. Test your programs with each of these postcodes:

TS16 1DA - should be VALID (two letters and two numbers at the start)  
DT2 7EW - should be VALID (two letters and one number at the start)  
W1 8BL - should be VALID (one letter and one number at the start)  
C17 9DF - should be VALID (one letter and two numbers at the start)  
E 9GL - should be INVALID (no number in the first part)  
N32 178 - should be INVALID (no letters in the second part)

EC1A 1CD - will be judged invalid, although this is a valid postcode.

The validation routine still needs some work! See question 4.

(b) Complete the following program to allow the user to test as many postcodes as they like, stopping when they press **Enter** instead of a postcode.

**Note:** The validation rule is simplified in this program, it just checks that the postcode starts with 2 uppercase letters followed by a number.

#Worksheet 1 Qu 3: Program tests postcodes according to the rule,

#"starts with 2 capital letters then a number"

import re

print ("Test as many postcodes as you like.. press Enter to end.")

anotherGo = True

while anotherGo:

postcode = input("Please input your postcode: ")

valid = **# INSERT LINE HERE**

if len(postcode)==0:

**# INSERT LINE HERE**

else:

if **# INSERT CONDITION HERE**

print("valid postcode")

else:

print("invalid postcode")

**4. Extension: A regular expression to validate any postcode**

Use the reference table of expressions and descriptions above question 3 to alter your postcode validation so that it accepts all valid postcodes and rejects all invalid postcodes. Valid postcodes have one of the following formats (L= letter, N = number):

|  |  |
| --- | --- |
| **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 |

Write here the regular expression that you used.