regular expressions: regexp

> A regular expression, regex or regexp is a sequence of characters that define a search pattern. Usually this pattern is used by string searching algorithms for "find" or "find and replace" operations on strings, or for input validation.

- Open regexone.com/lesson and go through **lessons 1 to 5**.
- Open your terminal and clone the repository: `pythonclubtmtl/learning_python3`
- We are going to use the grep command to find all occurrences of the word `python` in the file `002-using-pythonshell.md`. In the `learning_python3` folder from your terminal: `grep "python" 002-using-pythonshell.md`
- In the whole repository: `grep -r "python" <path>`
  - Reminder `.` means here (in the current folder)
- Use regular expressions instead of a string to find **all occurrences of double digit numbers** (ex: `42`, `51`,...)
- Use regular expressions to find **all occurrences of any letter followed by a single digit** (ex: `k3`)

regexp in python: try it

Let's find all double digits from a string (python console):

```
# regex pacakge
>>> import re
# We're looking for double digits only
>>> regex = r"[0-9][0-9]"
# some random text
>>> text = "Hi 42, it's me, 24"
# Get strings that fit regex (occurrences)
>>> matches = re.findall(regex, text)
>>> matches
# Get all occurrences with position
>>> matches = re.finditer(regex, text)
>>> for match in matches:
#       occurrence    , char # start , char # end
...     match.group(0), match.start(), match.end()
...
```

regexp: you do it now

- Open your "baby name parser" script from the previous session
- Modify your script to return:
  - All female names that contain (anywhere) the letter `c` followed by any letter, then the letter `a` (`c*a***` or `**c*a**` or `******c*a` or `...`), then find the most popular one

> Note: re.findall(regex, text, re.IGNORECASE) will make re case insensitive (as if all characters from the text are lower case.

- Next time, we will tokenize a corpus and work toward getting its keywords.