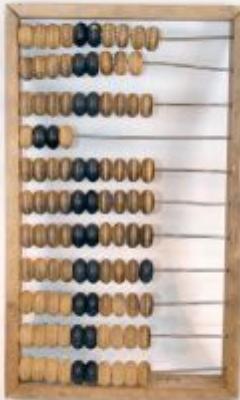


Завдання №5

Прикладна аналітика

Читання та запис файлів

1. Рядки коду



Один із способів вимірюти складність програми - підрахувати кількість [рядків коду](#) (LOC), не враховуючи порожні рядки та коментарі. Наприклад, програма типу:

```
# Say hello

name = input("What's your name? ")
print(f"hello, {name}")
```

має лише два рядки коду, а не чотири, оскільки перший рядок є коментарем, а другий - порожній (тобто просто пробіли).

Це не так вже й багато, тож, швидше за все, програма не така вже й складна. Звичайно, те, що одна програма (або навіть функція) має більше рядків коду, ніж інша, не обов'язково означає, що вона складніша. Наприклад, функція на зразок:

```
def is_even(n):
    if n % 2 == 0:
        return True
    else:
        return False
```

насправді не вдвічі складніша за таку функцію, як

```
def is_even(n):
    return n % 2 == 0
```

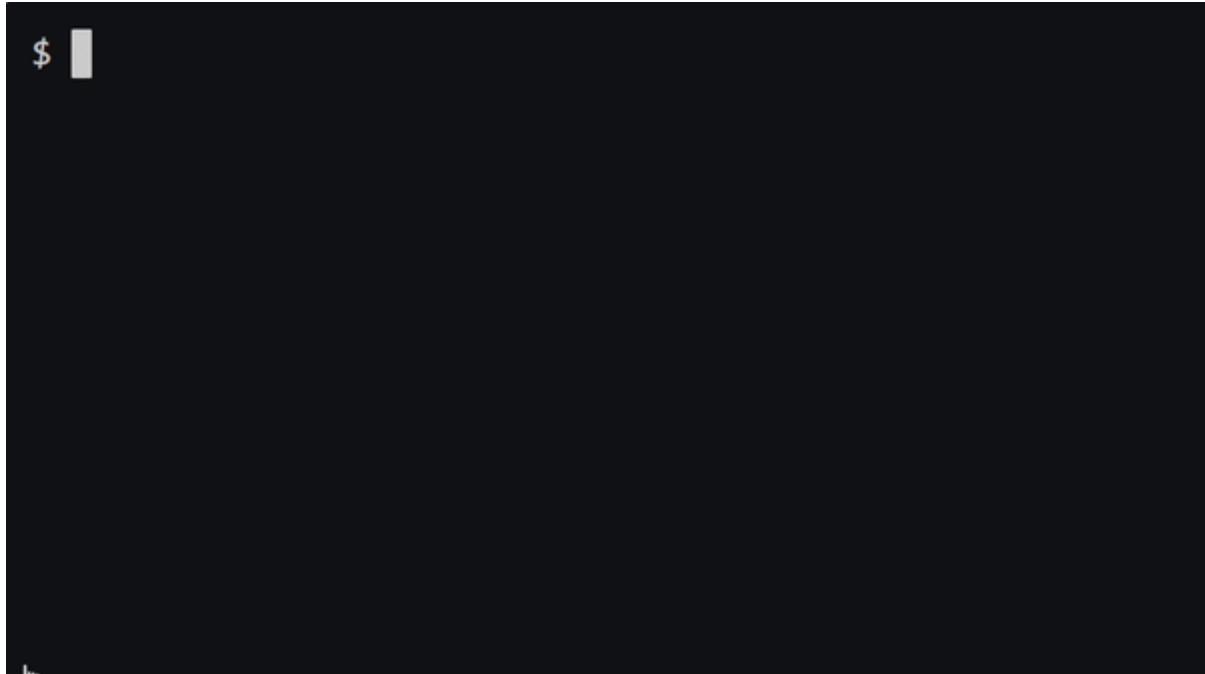
навіть незважаючи на те, що перша має (більш ніж) удвічі більше рядків коду. Насправді, перша може бути простішою, якщо її легше читати! Тож рядки коду слід сприймати з обережністю.

Завдання: у файлі `lines.py` реалізуйте програму, яка очікує рівно один аргумент командного рядка, ім'я (або шлях) файлу Python, і виводить кількість рядків коду у цьому файлі, виключаючи коментарі та порожні рядки. Якщо користувач не вказав рівно один аргумент командного рядка, або якщо ім'я вказаного файлу не закінчується на `.py`, або якщо вказаного файлу не існує, програма повинна завершити роботу через `sys.exit`.

Вважайте, що будь-який рядок, який починається з `#`, перед яким необов'язково ставиться пробіл, є коментарем ([docstring](#) не слід вважати коментарем). Вважатимемо, що будь-який рядок, який містить лише пробіли, є порожнім.

Підказка

Демонстрація роботи програми:



2. Döner



Напевно одна з найпопулярніших страв сучасного стріт-фуду — це шаурма та її різновиди. Її можна зустріти біля кожного метро, ТРЦ або зупинки громадського транспорту.

Студенти, як правило, купують шаурму з курятини, але в меню також є з яловичиною, фалафель та готові набори як показано в CSV-файлі [chicken.csv](#):

```
Chicken Shawarma,Small,Large
Classic,120.00,190.00
Assorted,155.00,225.00
Spicy,150.00,220.00
Pineapple,150.00,220.00
Double cheese,150.00,220.00
```

Дивіться також файл [veal.csv](#) для отримання CSV-файлу шаурми з телятини.

Звичайно, CSV-файл - не найзручніший формат для перегляду. Красивіше може бути таблиця, відформатована у форматі [ASCII art](#), як ця:

	Small	Large
Chicken Shawarma	120.00	190.00
Classic	155.00	225.00
Assorted	150.00	220.00
Spicy	150.00	220.00
Pineapple	150.00	220.00
Double cheese	150.00	220.00

Завдання: у файлі з назвою [shawarma.py](#) реалізуйте програму, яка очікує рівно один аргумент командного рядка, ім'я (або шлях) CSV-файлу і виводить таблицю у форматі ASCII art за допомогою `tabulate`, пакета на PyPI за адресою <https://pypi.org/project/tabulate>.

Відформатуйте таблицю за допомогою `grid` формату бібліотеки. Якщо користувач не вказав жодного аргументу командного рядка, або якщо ім'я вказаного файлу не закінчується на `.csv`, або якщо вказаного файлу не існує, програма має завершити роботу за допомогою `sys.exit`.

Підказка

3. Очищення



Дані також часто потребують "очищення", наприклад, шляхом переформатування, щоб значення мали узгоджений, якщо не зручніший, формат. Розглянемо, наприклад, цей CSV-файл студентів¹, `before.csv`, нижче:

```
name,house
"Abbott, Hannah",Hufflepuff
"Bell, Katie",Gryffindor
"Bones, Susan",Hufflepuff
"Boot, Terry",Ravenclaw
"Brown, Lavender",Gryffindor
"Bulstrode, Millicent",Slytherin
"Chang, Cho",Ravenclaw
"Clearwater, Penelope",Ravenclaw
"Crabbe, Vincent",Slytherin
"Creevey, Colin",Gryffindor
"Creevey, Dennis",Gryffindor
"Diggory, Cedric",Hufflepuff
"Edgecombe, Marietta",Ravenclaw
"Finch-Fletchley, Justin",Hufflepuff
"Granger, Hermione",Gryffindor
```

Хоча кожен "рядок" у файлі має три значення (прізвище, ім'я та будинок), перші два об'єднано в один "стовпчик" (ім'я), взято в подвійні лапки, а прізвище та ім'я розділено комою та пробілом. Не ідеальний варіант, якщо [Гогвортс](#) хоче надіслати кожному учневі стандартного листа, оскільки було б дивно починати лист з цього:

Dear Potter, Harry,

А не, наприклад, з ним:

Dear Harry,

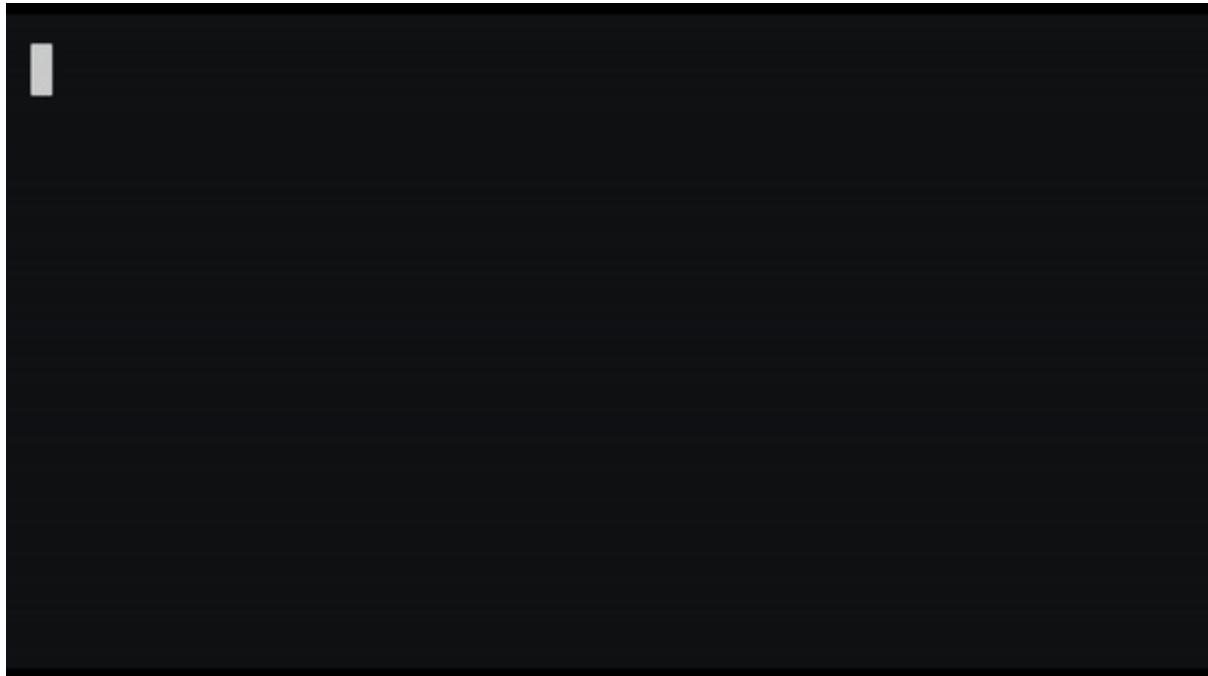
Завдання: У файлі з назвою `scourgify.py` реалізуйте програму, яка:

- Очікує від користувача два аргументи командного рядка:
 - ім'я існуючого CSV-файлу для читання на вході, стовпці якого мають бути, у порядку, `name` та `house`, та
 - ім'я нового CSV-файлу для запису у вихідний файл, стовпці якого мають бути за порядком: `first`, `last` та `house`.
- Перетворює вхідні дані у вихідні, розбиваючи кожне `name` на `first` (ім'я) та `last` (прізвище). Припустимо, що кожен студент має як ім'я, так і прізвище.

Якщо користувач не введе рівно два аргументи командного рядка, або якщо перший аргумент не може бути прочитаний, програма повинна завершити роботу через `sys.exit` з повідомленням про помилку.

Підказка

Демонстрація роботи програми:



4. FIT-shirt



Завдання: у файлі з назвою `shirt.py` реалізуйте програму, яка очікує рівно два аргументи командного рядка:

- у `sys.argv[1]` - ім'я (або шлях) до файлу JPEG або PNG, який потрібно прочитати (тобто відкрити) на вході
- у `sys.argv[2]` - ім'я (або шлях) до файлу JPEG або PNG, який потрібно записати (тобто зберегти) на виході.

Потім програма повинна накласти [`shirt.png`](#) (з прозорим фоном) на вхідні дані після зміни розміру та обрізання вхідних даних до однакового розміру, зберігши результат як вихідні дані.

Відкрийте вхідні дані за допомогою [`Image.open`](#), змініть розмір і обріжте вхідні дані за допомогою [`ImageOps.fit`](#), використовуючи значення за замовчуванням для параметрів `method`, `bleed` і `centering`, накладіть футболку за допомогою [`Image.paste`](#) і збережіть результат за допомогою [`Image.save`](#).

Програма повинна завершити роботу через `sys.exit` у випадку:

- якщо користувач не вказав рівно два аргументи командного рядка,
- якщо назви вхідних і вихідних даних не закінчуються на `.jpg`, `.jpeg` або `.png`, незалежно від регістру,

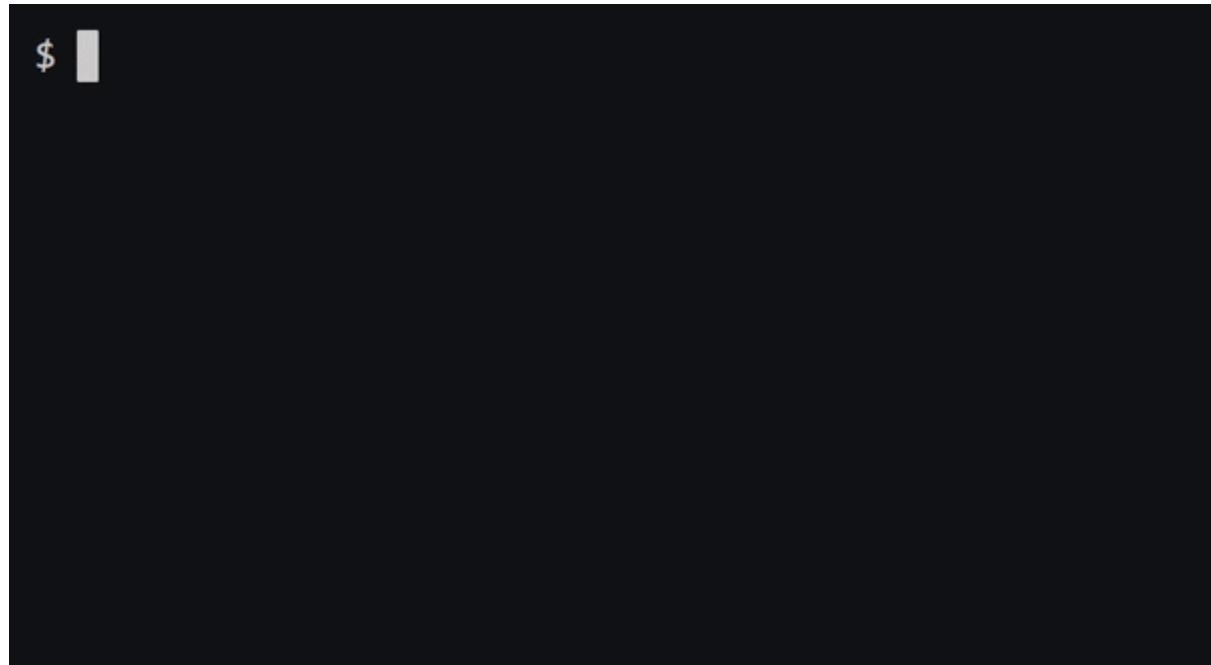
- якщо ім'я вхідних даних не має такого ж розширення, як ім'я вихідних даних, або
- якщо вказаний вхідний файл не існує.

Припустимо, що вхідними даними буде фотографія людини, яка позує у правильній позі, як на цих демонстраційних прикладах, так що після зміни розміру та обрізання сорочка буде ідеально сидіти на ній.

Якщо ви хочете запустити свою програму на власній фотографії, спочатку перетягніть фотографію до файлового провідника VS Code у ту саму папку, що й `shirt.py`. Не потрібно надсилати жодних фотографій разом з кодом.

Підказка

Демонстрація роботи програми:



До

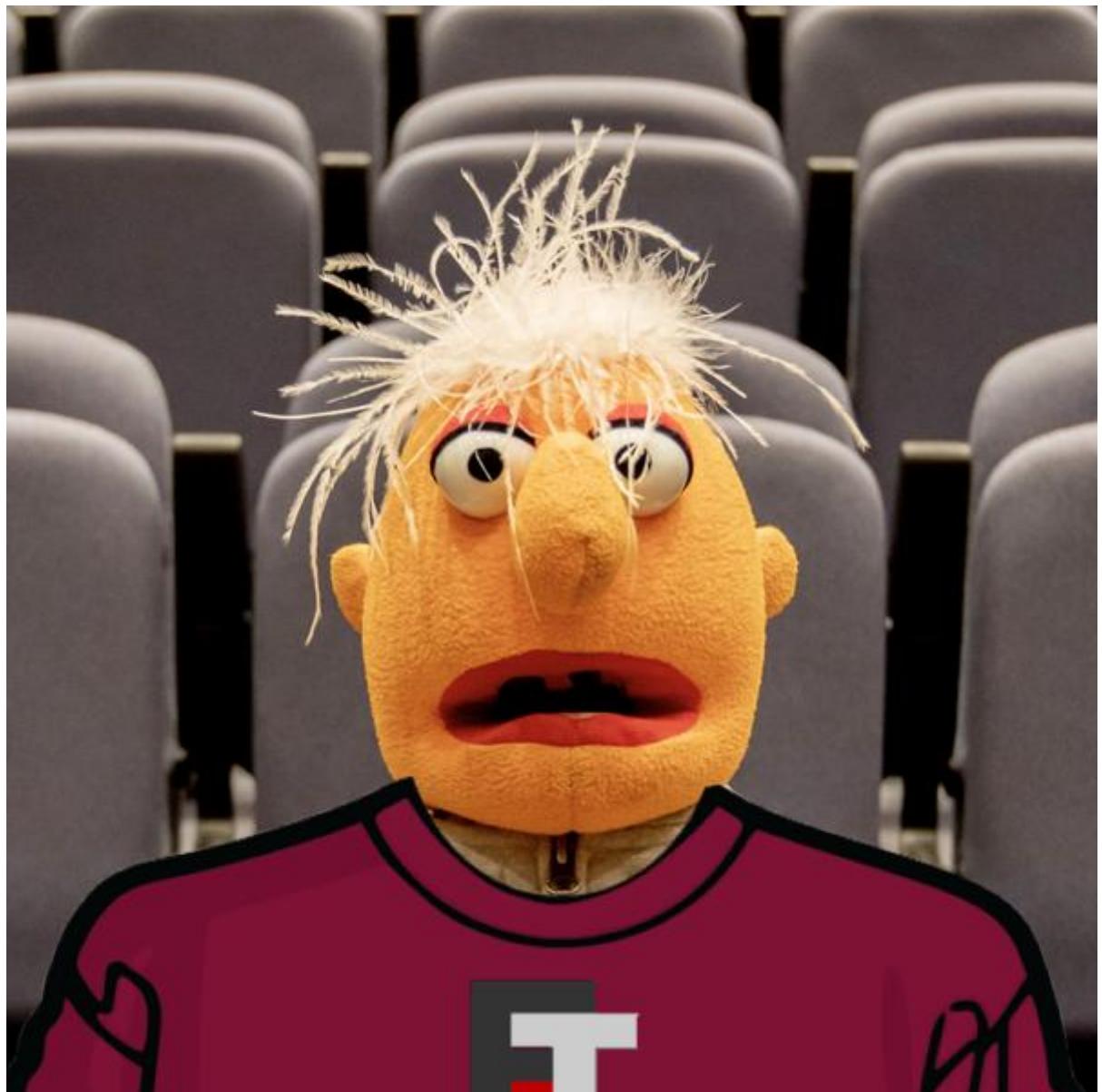


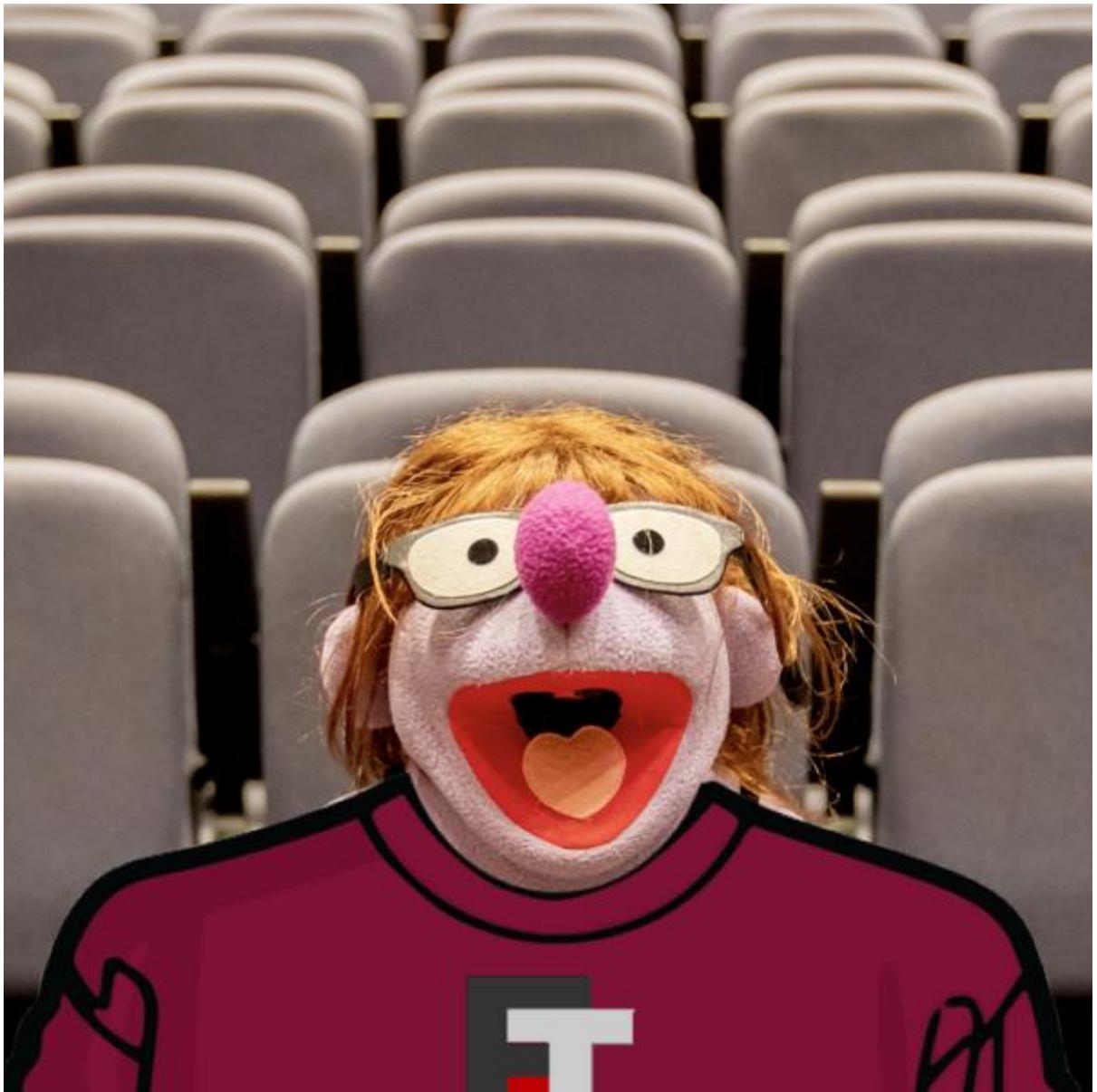




Після







Footnotes

1. Ці дані були отримані з https://en.wikipedia.org/wiki/List_of_Harry_Potter_characters. ↪