

Примеры работы с виджетами Jupyter Notebook: загрузка изображений

Алла Тамбовцева

Введение: загрузка и подготовка данных

В этом небольшом практикуме мы посмотрим, как, используя виджеты из библиотеки `ipywidgets`:

- запросить ввод у пользователя с помощью выпадающего меню и радио-кнопок;
- вывести на экран строки датафрейма, которые запросил пользователь, и изображения, которые им соответствуют.

Для начала импортируем библиотеки `pandas` и `ipywidgets` с сокращенными названиями:

In [1]:

```
import pandas as pd
import ipywidgets as widgets
```

В продолжение [занятия](https://nbviewer.org/github/allatambov/PyGoOn/blob/main/parsing/PARSING_CTD.ipynb) (https://nbviewer.org/github/allatambov/PyGoOn/blob/main/parsing/PARSING_CTD.ipynb) по парсингу загрузим данные из CSV-файла, который содержит информацию по актерам сказки «Не покидай...»:

In [2]:

```
final = pd.read_csv('final_NP.csv')
```

In [3]:

```
final.head()
```

Out[3]:

| | Unnamed: 0 | Актер | Роль | Число лайков | Число дизлайков | Рейтинг | Фото |
|---|------------|-------------------------|----------------|--------------|-----------------|---------|---------------------------------------|
| 0 | 0 | Лидия Федосеева-Шукшина | Королева Флора | 97 | 21 | 76 | jpeg/Королева(Федосеева-Шукшина).jpeg |
| 1 | 1 | Вячеслав Невинный | Король Теодор | 139 | 4 | 135 | jpeg/Король(Невинный).jpeg |
| 2 | 2 | Игорь Красавин | Патрик | 116 | 12 | 104 | jpeg/Патрик(Красавин).jpeg |
| 3 | 3 | Варвара Владимирова | Альбина | 129 | 7 | 122 | jpeg/Альбина(Владимирова).jpeg |
| 4 | 4 | Светлана Селезнёва | Марселла | 88 | 20 | 68 | jpeg/Марселла(Селезнёва).jpeg |

В столбце `Фото` сохранен относительный путь к файлу с фотографией актера/актрисы. Так как с обработкой изображений, загруженных по ссылке и не сохраненных на компьютере, часто возникают проблемы, мы будем работать с файлами, которые хранятся локально, в папке с названием `jpeg`.

Для этого скачайте себе [папку](https://www.dropbox.com/scl/fo/ypssd6c6sfmb08zejpt8t/h?dl=0&rlkey=vrqeuxfntzjy4fxkflfcrhnc) (<https://www.dropbox.com/scl/fo/ypssd6c6sfmb08zejpt8t/h?dl=0&rlkey=vrqeuxfntzjy4fxkflfcrhnc>) с изображениями (он небольшой, всего 20 кадров) как zip-архив, распакуйте полученный архив и поместите папку `jpeg` рядом с ipynb-файлом, в котором сейчас работаете.

Примечание. Если хочется узнать, как скачивать изображения с Github и сразу их открывать, посмотрите этот [пример кода](https://gist.github.com/mjdietzx/545fa2874b2688e9bcb71e2ee92cd5a0) (<https://gist.github.com/mjdietzx/545fa2874b2688e9bcb71e2ee92cd5a0>) от пользователя `mjdietzx`.

Чтобы понять логику дальнейшей работы с изображениями и проверить, что они отображаются, давайте для примера обработаем один файл. Для этого файл сначала нужно открыть – загрузить его как бинарный файл для чтения (`rb` – от *read binary*):

In [4]:

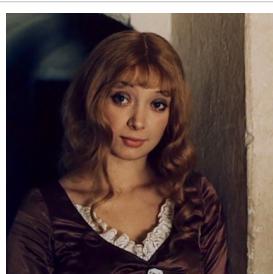
```
f = open('jpeg/Марселла(Селезнёва).jpeg', 'rb').read()
```

Если в отдельной ячейке запросить `f`, мы действительно увидим что-то странное, это будет содержимое файла в бинарном виде (как предупреждают в различных документациях, *not human-readable*). Но такой объект полезен – его можно подать на вход функции `Image` из библиотеки `ipywidgets`, и преобразовать в изображение, которое уже понятным образом выводится на экран:

In [5]:

```
# в виде png (качество получше)
# ширина 200 пунктов

image = widgets.Image(value = f, format = 'png', width = 200)
display(image)
```



Отлично! Все работает, данные есть, изображения открываются. Все готово к работе.

Добавление меню для выбора опций и вывод таблицы на экран

Сформулируем первую часть нашей глобальной задачи.

Пользователь должен иметь возможность выбрать, по какому принципу отсортировать актеров (число лайков или рейтинг, посчитанный как число лайков минус число дизлайков) и сколько строк он хочет увидеть на экране (все 20, топ 10, топ 5 или топ 3). Сортировка в любом случае производится по убыванию, так как логично начинать с актеров, чья игра понравилась большему количеству зрителей.

Создадим два виджета: выпадающее меню для выбора основания сортировки и радио-кнопки для выбора количества отображаемых строк. Для выпадающего меню нам понадобится функция `Dropdown()`. В аргументе `options` перечислим варианты ответа в виде списка, в аргументе `value` – значение по умолчанию, а в аргументе `description` – текст, который будет отображаться рядом с меню (про настройки внешнего вида меню, увеличение места для текста можно почитать [здесь](https://ipywidgets.readthedocs.io/en/stable/examples/Widget%20Styling.html) (<https://ipywidgets.readthedocs.io/en/stable/examples/Widget%20Styling.html>)):

In [6]:

```
rating_widget = widgets.Dropdown(  
    options = ['Число лайков', 'Рейтинг'],  
    value = "Число лайков",  
    description = 'Сортировка:',  
)
```

Для создания радио-кнопок воспользуемся функцией `RadioButtons()`. Логика работы здесь такая же, что и выше, но аргумент `options` мы запишем по-другому – в виде списка кортежей. В этом списке будут пары `метка-значение`, где метка – текст, который увидит пользователь, а значение – число, которое мы потом сможем использовать для отбора нужного количества строк.

In [7]:

```
mode_widget = widgets.RadioButtons(  
    options=[('Все', final.shape[0]), ('Топ 10', 10), ('Топ 5', 5), ('Топ 3', 3)],  
    value=10,  
    description='Формат:',  
)
```

Зачем такой формат записи нужен? Для того, чтобы упростить дальнейший код. Нам не придется дописывать условия с `if-else` и сообщать Python, что если выбрано значение Топ 10, в метод `.head()` нужно подставлять число 10, мы сможем сразу подставить значение, извлеченное из виджета.

Как эти значения, извлеченные из виджета, получить? Вывести сам виджет на экран и после выбора опций сохранить значения атрибута `value`. Протестируем:

In [8]:

```
# по умолчанию отмечены опции из value,  
# выберем Рейтинг и топ 5  
  
display(rating_widget)  
display(mode_widget)
```

Сортировка: Число лайков

Формат: Все
 Топ 10
 Топ 5
 Топ 3

In [9]:

```
rating_choice = rating_widget.value  
mode_choice = mode_widget.value
```

In [10]:

```
# сохранилось!  
  
print(rating_choice)  
print(mode_choice)
```

Рейтинг
5

Отлично! Теперь нам нужно учсть выбранные опции при выводе таблицы на экран. Для этого нам нужно:

- выполнить сортировку с помощью метода `.sort_values()`, сообщить, что сортировка производится по убыванию и подставить показатель сортировки из переменной `rating_choice`;
- для красоты изменить нумерацию строк – перезаписать содержимое атрибута `.index`, подставив туда последовательность чисел от 1 до 20 включительно (всего в датафрейме 20 строк);
- выбрать столбцы с Актер до Рейтинг включительно, чтобы не показывать неинформативные столбцы, и вывести необходимое число строк на экран, подставив в метод `.head()` значение из переменной `mode_choice`.

Весь код сразу:

In [11]:

```
display(rating_widget)
display(mode_widget)
```

Сортировка: Число лайков

Формат: Все
 Топ 10
 Топ 5
 Топ 3

In [12]:

```
# сохраняем
rating_choice = rating_widget.value
mode_choice = mode_widget.value

# сортируем, выбираем и показываем
to_show = final.sort_values(rating_choice, ascending = False)
to_show = to_show.set_index(pd.Index(range(1, 21)))
head = to_show.loc[:, "Актер": "Рейтинг"].head(mode_choice)
display(head)
```

| | Актер | Роль | Число лайков | Число дизлайков | Рейтинг |
|----|---------------------|---------------------|--------------|-----------------|---------|
| 1 | Вячеслав Невинный | Король Теодор | 139 | 4 | 135 |
| 2 | Альберт Филозов | Канцлер граф Давиль | 129 | 4 | 125 |
| 3 | Варвара Владимирова | Альбина | 129 | 7 | 122 |
| 4 | Артём Тынкасов | Пенапью | 120 | 2 | 118 |
| 5 | Анатолий Рудаков | Капитан Удилак | 114 | 3 | 111 |
| 6 | Игорь Красавин | Патрик | 116 | 12 | 104 |
| 7 | Владимир Ставицкий | Жак Веснушка | 105 | 4 | 101 |
| 8 | Регина Разумा | Оттилия | 106 | 7 | 99 |
| 9 | Александр Денисов | Главарь разбойников | 93 | 3 | 90 |
| 10 | Елена Антонова | Марта Веснушка | 89 | 10 | 79 |

Вариант 1: выводим таблицу и изображения через Image()

Теперь в дополнение к таблице нам нужно добавить изображения. Как мы уже видели в небольшом примере выше, если нам просто нужна картинка фиксированной ширины/высоты, без подписей и прочих элементов, нам достаточно виджета с изображением, который создается с помощью `Image()`.

В нашем случае с таблицей картинок должно быть несколько – столько же, сколько строк захотел увидеть пользователь. Возникает явная необходимость задействовать цикл или списковое включение. Но для начала давайте заберем названия файлов, которые нам потребуются. Код для формирования таблицы выглядел так:

In [13]:

```
to_show = final.sort_values(rating_choice, ascending = False)
to_show = to_show.set_index(pd.Index(range(1, 21)))
head = to_show.loc[:, "Актер": "Рейтинг"].head(mode_choice)
```

Заберем из `to_show` значения из столбца `Фото`, но не все, а то число, которое сохранено в `mode_choice`, это будут пути к файлам:

In [14]:

```
names = to_show["Фото"][:mode_choice]
names
```

Out[14]:

```
1      jpeg/Король(Невинный).jpeg
2      jpeg/Канцлер(Филозов).jpeg
3      jpeg/Альбина(Владимирова).jpeg
4      jpeg/Пенапью(Тынкасов).jpeg
5      jpeg/Удилак(Рудаков).jpeg
6      jpeg/Патрик(Красавин).jpeg
7      jpeg/Жак(Ставицкий).jpeg
8      jpeg/Оттилия(Разумы).jpeg
9      jpeg/Разбойник(Денисов).jpeg
10     jpeg/Марта(Антонова).jpeg
Name: Фото, dtype: object
```

Теперь с помощью спискового включения откроем каждый файл, загрузим в бинарном формате и сохраним в список `images`:

In [15]:

```
images = [open(n, 'rb').read() for n in names]
```

Наконец, передадим эти бинарные файлы на вход функции `Image()` и создадим виджет с изображением для каждого файла:

In [16]:

```
im_widgets = [widgets.Image(value = i, format = 'png', width = 150) for i in images]
```

Чтобы эти виджеты выводились в структурированном виде, в один ряд, объединим их в более крупный виджет – в горизонтальный блок `HBox`:

In [17]:

```
sidebyside = widgets.HBox(im_widgets)
```

Посмотрим на результат:

In [18]:

```
display(sidebyside)
```



Отлично! Что приятно, если изображений много, автоматически добавится горизонтальный скроллинг. Давайте объединим все части кода и посмотрим на пример с добавленным скроллингом:

In [19]:

```
display(rating_widget)
display(mode_widget)
```

Сортировка: Число лайков

Формат: Все
 Топ 10
 Топ 5
 Топ 3

In [20]:

```
# сохраняем
rating_choice = rating_widget.value
mode_choice = mode_widget.value

# сортируем и выбираем
to_show = final.sort_values(rating_choice, ascending = False)
to_show = to_show.set_index(pd.Index(range(1, 21)))
head = to_show.loc[:, "Актер": "Рейтинг"].head(mode_choice)

# обрабатываем изображения
names = to_show["Фото"][:mode_choice]
images = [open(n, 'rb').read() for n in names]
im_widgets = [widgets.Image(value = i, format = 'png', width = 150) for i in images]
sidebyside = widgets.HBox(im_widgets)

# выводим результат
display(head)
display(sidebyside)
```

| | Актер | Роль | Число лайков | Число дизлайков | Рейтинг |
|----|-------------------------|---------------------|--------------|-----------------|---------|
| 1 | Вячеслав Невинный | Король Теодор | 139 | 4 | 135 |
| 2 | Варвара Владимирова | Альбина | 129 | 7 | 122 |
| 3 | Альберт Филозов | Канцлер граф Давиль | 129 | 4 | 125 |
| 4 | Артём Тынкасов | Пенаплю | 120 | 2 | 118 |
| 5 | Игорь Красавин | Патрик | 116 | 12 | 104 |
| 6 | Анатолий Рудаков | Капитан Удилак | 114 | 3 | 111 |
| 7 | Регина Разумова | Оттилия | 106 | 7 | 99 |
| 8 | Владимир Ставицкий | Жак Веснушка | 105 | 4 | 101 |
| 9 | Лидия Федосеева-Шукшина | Королева Флора | 97 | 21 | 76 |
| 10 | Александр Денисов | Главарь разбойников | 93 | 3 | 90 |
| 11 | Елена Антонова | Марта Веснушка | 89 | 10 | 79 |
| 12 | Светлана Селезнёва | Марселла | 88 | 20 | 68 |
| 13 | Юрий Багинян | Лысый лакей | 73 | 2 | 71 |
| 14 | Валентин Букин | Хозяин таверны | 64 | 5 | 59 |
| 15 | Леонид Нечаев | Точильщик ножей | 51 | 1 | 50 |
| 16 | Анатолий Голуб | Студент | 37 | 1 | 36 |
| 17 | Ростислав Рахт | Шарманщик | 24 | 0 | 24 |
| 18 | Владимир Корпусь | Жандарм | 21 | 0 | 21 |
| 19 | Владимир Зубенко | Музыкант | 20 | 0 | 20 |
| 20 | Валерий Мороз | Дворецкий | 19 | 0 | 19 |



Вариант 2: выводим таблицу и изображения через HTML()

Теперь посмотрим, как будет выглядеть решение той же задачи, но с помощью виджета с HTML. Содержимое этого виджета можно считать миниатюрным вариантом HTML-страницы, на которой можно разместить не только картинку, а вообще, что угодно, главное, знать, как это сделать на языке HTML.

Снова начнем с одной картинки. Вспомним, что для добавления изображений в HTML-файл используется тэг ``, а в атрибут `src` добавляется источник изображения – ссылка на файл:

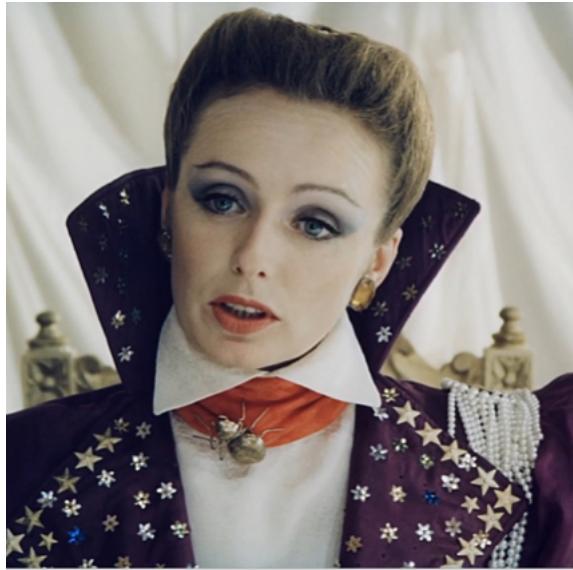
```

```

Поместим строку такого вида в функцию HTML и создадим виджет с изображением (аккуратно с кавычками, должны быть разного вида вокруг всей строки и вокруг названия файла):

In [21]:

```
s = ''  
w = widgets.HTML(s)  
display(w)
```



Если мы используем виджет HTML, изображение загружается «как есть», в случае, если его ширина меньше ширины страницы, и ужимается до ширины страницы, если оно слишком большое. Здесь картинка открылась в своем размере, 425 пунктов.

Итак, как можно догадаться, дальше план такой:

- создаем строку с кодом HTML для изображения, куда будут подставляться названия файлов, форматирование строк нам в помощь;
- через списковые включения формируем список со строками с кодом HTML;
- через списковые включения формируем список с виджетами HTML;
- объединяем все в HBox и выводим на экран.

In [22]:

```
display(rating_widget)  
display(mode_widget)
```

Сортировка: Число лайков

Формат: Все
 Top 10
 Top 5
 Top 3

In [23]:

```
rating_choice = rating_widget.value
mode_choice = mode_widget.value

to_show = final.sort_values(rating_choice, ascending = False)
to_show.set_index(pd.Index(range(1, 21)), inplace = True)
head = to_show.loc[:, "Актер": "Рейтинг"].head(mode_choice)

names = to_show[ "Фото" ][ 0:mode_choice]

# форматирование строк – вместо %s подставляется название файла в
images = ["<img src='%s'>"%n for n in names]
pages = [widgets.HTML(i) for i in images]
sidebyside = widgets.HBox(pages)

display(head)
display(sidebyside)
```

| | Актер | Роль | Число лайков | Число дизлайков | Рейтинг |
|----|-------------------------|---------------------|--------------|-----------------|---------|
| 1 | Вячеслав Невинный | Король Теодор | 139 | 4 | 135 |
| 2 | Варвара Владимирова | Альбина | 129 | 7 | 122 |
| 3 | Альберт Филозов | Канцлер граф Давиль | 129 | 4 | 125 |
| 4 | Артём Тынкасов | Пенаплю | 120 | 2 | 118 |
| 5 | Игорь Красавин | Патрик | 116 | 12 | 104 |
| 6 | Анатолий Рудаков | Капитан Удилак | 114 | 3 | 111 |
| 7 | Регина Разумя | Оттилия | 106 | 7 | 99 |
| 8 | Владимир Ставицкий | Жак Веснушка | 105 | 4 | 101 |
| 9 | Лидия Федосеева-Шукшина | Королева Флора | 97 | 21 | 76 |
| 10 | Александр Денисов | Главарь разбойников | 93 | 3 | 90 |



Не самое удачное решение задачи – картинки получаются довольно маленькими (попробуйте выбрать все 20 строк, будет совсем несерьезно). Поэтому давайте усовершенствуем наш код: во-первых, добавим отдельную обработку для каждого выбранного числа строк, а во-вторых, добавим подписи к каждому изображению.

Вариант 3: выводим таблицу и изображения с подписями через HTML, с учетом их количества

Создадим шаблон фрагмента кода HTML, в котором будет находиться изображение, заголовок с именем актера/актрисы и строка с указанием роли:

In [24]:

```
# <h5> заголовок пятого уровня
# <p> просто текст

template = """
<img src='%s'>
<h5>%s</h5>
<p>%s</p>
"""
```

Это обычная строка типа *string*, вместо первого `%s` мы будем подставлять ссылки на изображения, вместо второго – имена актеров, а вместо третьего – их роли. Давайте пока рассмотрим случай, когда выбраны не все строки, если выбраны все 20, на экран будет выводиться только таблица, без картинок (оператор `pass` поможет).

In [25]:

```
display(rating_widget)
display(mode_widget)
```

Сортировка: Число лайков

- Формат: Все
 Top 10
 Top 5
 Top 3

In [26]:

```
# сохраняем
rating_choice = rating_widget.value
mode_choice = mode_widget.value

# сортируем, отбираем и показываем таблицу
to_show = final.sort_values(rating_choice, ascending = False)
to_show.set_index(pd.Index(range(1, 21)), inplace = True)
head = to_show.loc[:, "Актер": "Рейтинг"].head(mode_choice)
display(head)

# получаем перечни путей к файлам, имен актеров и ролей
files = to_show[ "Фото" ][ 0 : mode_choice ]
actors = to_show[ "Актер" ][ 0 : mode_choice ]
roles = to_show[ "Роль" ][ 0 : mode_choice ]

# создаем список кортежей вида (файл, актер, роль) через zip
# подставляем в template первое, второе и третье значение
# из каждого такого кортежа

images = [template%(file, actor, role) for file, actor, role in zip(files, actors, roles)]
pages = [widgets.HTML(i) for i in images]

# если нужно 3 или 5 картинок, просто выводим их одним рядом, размер разумный
if mode_choice == 3 or mode_choice == 5:
    sidebyside = widgets.HBox(pages)
    display(sidebyside)

# если 10 картинок, разбиваем все на 2 ряда
elif mode_choice == 10:
    sidebyside1 = widgets.HBox(pages[ 0 : 5 ])
    sidebyside2 = widgets.HBox(pages[ 5 : ])
    display(sidebyside1)
    display(sidebyside2)

else:
    pass
```

| | Актер | Роль | Число лайков | Число дизлайков | Рейтинг |
|----|---------------------|---------------------|--------------|-----------------|---------|
| 1 | Вячеслав Невинный | Король Теодор | 139 | 4 | 135 |
| 2 | Альберт Филозов | Канцлер граф Давиль | 129 | 4 | 125 |
| 3 | Варвара Владимирова | Альбина | 129 | 7 | 122 |
| 4 | Артём Тынкасов | Пенаплю | 120 | 2 | 118 |
| 5 | Анатолий Рудаков | Капитан Удилак | 114 | 3 | 111 |
| 6 | Игорь Красавин | Патрик | 116 | 12 | 104 |
| 7 | Владимир Ставицкий | Жак Веснушка | 105 | 4 | 101 |
| 8 | Регина Разумा | Оттилия | 106 | 7 | 99 |
| 9 | Александр Денисов | Главарь разбойников | 93 | 3 | 90 |
| 10 | Елена Антонова | Марта Веснушка | 89 | 10 | 79 |



Вячеслав Невинный



Альберт Филозов



Варвара Владимирова



Артём Тынкасов



Анатолий Рудаков

Король Теодор

Канцлер граф Давиль

Альбина

Пенаплю

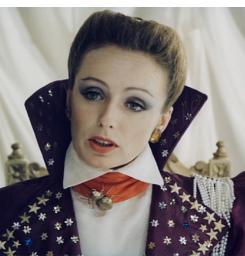
Капитан Удилак



Игорь Красавин



Владимир Ставицкий



Регина Разумा



Александр Денисов



Елена Антонова

Патрик

Жак Веснушка

Оттилия

Главарь разбойников

Марта Веснушка

Получилось!

Наконец, давайте обработаем случай с 20 картинками. Для этого сделаем заголовок поменьше и разобьем каждую строку с именем и фамилией на две, иначе у кого-то подпись к картинке не поместится на одну строку и картинка «съедет» вниз:

In [27]:

```
# h6 вместо h5
# </br> разрыв строки — для разбиения
# внутри одного заголовка

template20 = """


###### %s</br>%s</h6> %s """


```

In [28]:

```
display(rating_widget)
display(mode_widget)
```

Сортировка:

Формат: Все
 Топ 10
 Топ 5
 Топ 3

In [29]:

```
rating_choice = rating_widget.value
mode_choice = mode_widget.value

# сортируем, отбираем и показываем таблицу
to_show = final.sort_values(rating_choice, ascending = False)
to_show.set_index(pd.Index(range(1, 21)), inplace = True)
head = to_show.loc[:, "Актер": "Рейтинг"].head(mode_choice)
display(head)

# получаем перечни путей к файлам, имен актеров и ролей
files = to_show["Фото"][:mode_choice]
actors = to_show["Актер"][:mode_choice]
roles = to_show["Роль"][:mode_choice]

# разбиваем по пробелу и забираем отдельно имена и фамилии
names = [a.split()[0] for a in actors]
surnames = [a.split()[1] for a in actors]

# та же логика с zip()
# только здесь уже 4 элемента, имя и фамилия отдельно
images = [template20%file, name, surname, role) for file, name, surname,
          role in zip(files, names, surnames, roles)]
pages = [widgets.HTML(i) for i in images]

if mode_choice == 3 or mode_choice == 5:
    sidebyside = widgets.HBox(pages)
    display(sidebyside)

elif mode_choice == 10:
    sidebyside1 = widgets.HBox(pages[0:5])
    sidebyside2 = widgets.HBox(pages[5:])
    display(sidebyside1)
    display(sidebyside2)

else:
    sidebyside1 = widgets.HBox(pages[0:5])
    sidebyside2 = widgets.HBox(pages[5:10])
    sidebyside3 = widgets.HBox(pages[10:15])
    sidebyside4 = widgets.HBox(pages[15:])
    display(sidebyside1)
    display(sidebyside2)
    display(sidebyside3)
    display(sidebyside4)
```

| | Актер | Роль | Число лайков | Число дизлайков | Рейтинг |
|----|-------------------------|---------------------|--------------|-----------------|---------|
| 1 | Вячеслав Невинный | Король Теодор | 139 | 4 | 135 |
| 2 | Варвара Владимирова | Альбина | 129 | 7 | 122 |
| 3 | Альберт Филозов | Канцлер граф Давиль | 129 | 4 | 125 |
| 4 | Артём Тынкасов | Пенапью | 120 | 2 | 118 |
| 5 | Игорь Красавин | Патрик | 116 | 12 | 104 |
| 6 | Анатолий Рудаков | Капитан Удилак | 114 | 3 | 111 |
| 7 | Регина Разумा | Оттилия | 106 | 7 | 99 |
| 8 | Владимир Ставицкий | Жак Веснушка | 105 | 4 | 101 |
| 9 | Лидия Федосеева-Шукшина | Королева Флора | 97 | 21 | 76 |
| 10 | Александр Денисов | Главарь разбойников | 93 | 3 | 90 |
| 11 | Елена Антонова | Марта Веснушка | 89 | 10 | 79 |
| 12 | Светлана Селезнёва | Марселя | 88 | 20 | 68 |
| 13 | Юрий Багинян | Лысый лакей | 73 | 2 | 71 |
| 14 | Валентин Букин | Хозяин таверны | 64 | 5 | 59 |
| 15 | Леонид Нечаев | Точильщик ножей | 51 | 1 | 50 |
| 16 | Анатолий Голуб | Студент | 37 | 1 | 36 |
| 17 | Ростислав Рахт | Шарманщик | 24 | 0 | 24 |
| 18 | Владимир Корпусь | Жандарм | 21 | 0 | 21 |
| 19 | Владимир Зубенко | Музыкант | 20 | 0 | 20 |
| 20 | Валерий Мороз | Дворецкий | 19 | 0 | 19 |



Вячеслав
Невинный

Король Теодор

Варвара
Владимирова

Альбина

Альберт
Филозов

Канцлер граф Давиль

Артём
Тынкасов

Пенапль

Игорь
Красавин

Патрик



Анатолий
Рудаков

Капитан Удилак

Регина
Разума

Оттилия

Владимир
Ставицкий

Жак Веснушка

Лидия
Федосеева-Шукшина

Королева Флора

Александр
Денисов

Главарь разбойников



Елена
Антонова

Марта Веснушка

Светлана
Селезнёва

Марселла

Юрий
Багинян

Лысый лакей

Валентин
Букин

Хозяин таверны

Леонид
Нечаев

Точильщик ножей



Анатолий
Горяинов

Ростислав
Довут

Владимир
Коробец

Владимир
Зильberman

Валерий
Мамонов