1) ⭐ Task with an asterisk! ⭐ Load data from Google Sheets, which are located in the folder (table - Sheet1) using the library requests. Calculate the percentage of men among all visitors to the site, round the answer to integers.

In [1]:

```
import pandas as pd
import gspread
from oauth2client.service_account import ServiceAccountCredentials

# Define the URL to the Google Sheets document
url = 'https://docs.google.com/spreadsheets/d/1WZJ_HaJq_SjeUFCjjGPKbkye3lFaPr9pdYI_QwcdS5s/edit?usp=sharing'

# Define the scope and credentials for accessing Google Sheets API
scope = ['https://spreadsheets.google.com/feeds',
         'https://www.googleapis.com/auth/drive']
credentials = ServiceAccountCredentials.from_json_keyfile_name('manifest-verve-392606-1884b94a438f.json', scope)

# Authorize the credentials and open the Google Sheets document
gc = gspread.authorize(credentials)
doc = gc.open_by_url(url)

# Access a specific worksheet within the document
worksheet = doc.worksheet('table - Sheet1')

# Get all the values from the worksheet as a list of lists
data = worksheet.get_all_values()

data
```

Out[1]:

```
[['date', 'gender', 'users'],
 ['2020-06-14', 'Not specified', '150'],
 ['2020-06-14', 'male', '117'],
 ['2020-06-14', 'female', '21']]
```

In [2]:

```
headers = data.pop(0)
```

In [3]:

```
df = pd.DataFrame(data, columns=headers)
df
```

Out[3]:

	date	gender	users
0	2020-06-14	Not specified	150
1	2020-06-14	male	117
2	2020-06-14	female	21

In [4]:

```
df.users = df.users.astype(int)
df.dtypes
```

Out[4]:

```
date      object
gender    object
users      int32
dtype: object
```

In [5]:

```
int(round(df.query('gender == "male"').users / df.users.sum() * 100))
```

Out[5]:

```
41
```

2) Download the file with Yandex.Metric site attendance data (karpov_courses_test.csv). Using the np.where function, create a series that will record 1 if the visits were made by robots from a browser containing Chrome in the name and 0 in other cases. Put the resulting series into the chrome_robots variable.

May need: str.contains()

In [9]:

```
df = pd.read_csv('C:/Users/stask/Analitics_Karpov/Module7/Lesson/karpov_courses_test.csv')
df.head()
```

Out[9]:

	date	browser	user_type	users
0	2020-06-14	Google Chrome	People	139
1	2020-06-14	Chrome Mobile	People	35
2	2020-06-14	Google Chrome	Robots	28
3	2020-06-14	Yandex Browser	People	22
4	2020-06-14	Firefox	People	15

In [18]:

```
import numpy as np
```

In [20]:

```
# (df.user_type == "Robots") & (df.browser.str.contains('Chrome'))
np.where((df.user_type == "Robots") & (df.browser.str.contains('Chrome')), 1, 0)
```

Out[20]:

```
array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0])
```

In [24]:

```
chrome_robots = pd.Series(np.where((df.user_type == "Robots") & (df.browser.str.contains('Chrome')), 1, 0))
# df.user_type.map({'People':'Humans', 'Robots':'Bots'})
```

In [25]:

```
chrome_robots.head()
```

Out[25]:

```
0    0
1    0
2    1
3    0
4    0
dtype: int32
```

3) ⭐ Task with an asterisk!⭐ You have a dataset sample_ads.csv with data on clicks and impressions by ad, there are gaps in this dataset. Calculate the number of rows with no skips in the click_type column. Write the answer to the not_nan_count variable

```
ad_id - ad id
user_id - user id
click_type - click type (click on the ad title, click in the centre).
```

In [31]:

```
df = pd.read_csv('C:/Users/stask/Analitics_Karpov/Module7/Lesson/sample_ads.csv')
df.dtypes
```

Out[31]:

```
ad_id          int64
user_id       object
click_type    object
dtype: object
```

In [39]:

```
# not_nan_count = df['click_type'].dropna().shape[0]
(~df.click_type.isna()).sum()
```

Out[39]:

```
11
```

4) ⭐ Task with an asterisk!⭐ The lst list contains various items. Create a new_list and do the following:

- using the try except and loop construct, try to convert each value in the list to int
- if it succeeds - add the converted value to the new_list.

In [45]:

```
lst = ['0', '1', 'Hello', '57']
try:
    new_list = [int(i) for i in lst]
except:
    print('A problem with data in list')
```

```
A problem with data in list
```

5) ⭐ Task with an asterisk! ⭐ Now let's try to use the knowledge of using Telegram API. Send any message to yourself in a private message box, and in response to this step write the nickname of the bot on behalf of which you will send messages.

In [51]:

```
import requests
import json
from urllib.parse import urlencode


token = '6318778754:AAHkPoabEE-i_4w2u82wIRiS50sDbJPSJQ4'
chat_id = 342603321  # your chat id

message = 'Hello, Stas'  # text which you want to send

params = {'chat_id': chat_id, 'text': message}

base_url = f'https://api.telegram.org/bot{token}/'
url = base_url + 'sendMessage?' + urlencode(params)

resp = requests.get(url)
```