Reemplazar grupos con expresiones regulares en un DataFrame de Pandas

In [1]:
```
import pandas as pd

cols = ['Date', 'Time', 'Latitude', 'Longitude', 'Depth', 'Magnitude Type']
df_e = pd.read_csv(f'..\data/earthquakes_1965_2016_database.csv.zip', dtype=str)[cols]

df_e
```

Out[1]:

	Date	Time	Latitude	Longitude	Depth	Magnitude Type
0	01/02/1965	13:44:18	19.246	145.616	131.6	MW
1	01/04/1965	11:29:49	1.863	127.352	80	MW
2	01/05/1965	18:05:58	-20.579	-173.972	20	MW
3	01/08/1965	18:49:43	-59.076	-23.557	15	MW
4	01/09/1965	13:32:50	11.938	126.427	15	MW
...	...	...	...	...	...	...
23407	12/28/2016	08:22:12	38.3917	-118.8941	12.3	ML
23408	12/28/2016	09:13:47	38.3777	-118.8957	8.8	ML
23409	12/28/2016	12:38:51	36.9179	140.4262	10	MWW
23410	12/29/2016	22:30:19	-9.0283	118.6639	79	MWW
23411	12/30/2016	20:08:28	37.3973	141.4103	11.94	MB

23412 rows × 6 columns

Cómo hacer coincidir y reemplazar grupos con expresiones regulares; patrones de fecha

In [2]:
```
df_e['Date']
```

Out[2]:
```
0        01/02/1965
1        01/04/1965
2        01/05/1965
3        01/08/1965
4        01/09/1965
            ...
23407    12/28/2016
23408    12/28/2016
23409    12/28/2016
23410    12/29/2016
23411    12/30/2016
Name: Date, Length: 23412, dtype: object
```

In [3]:
```
df_e['Date'].str.replace(r'(\d{2})/(\d{2})/\d{2}(\d{2})', r"\2 \1 '\3", regex=True)
```

Out[3]:
```
0        02 01 '65
1        04 01 '65
2        05 01 '65
3        08 01 '65
4        09 01 '65
           ...
23407    28 12 '16
23408    28 12 '16
23409    28 12 '16
23410    29 12 '16
23411    30 12 '16
Name: Date, Length: 23412, dtype: object
```

In [4]:
```
df_e['Date'].str.replace(r'(\d{2})/(\d{2})/(\d{4})', r"\3-\2-\1", regex=True)
```

Out[4]:
```
0        1965-02-01
1        1965-04-01
2        1965-05-01
3        1965-08-01
4        1965-09-01
            ...
23407    2016-28-12
23408    2016-28-12
23409    2016-28-12
23410    2016-29-12
23411    2016-30-12
Name: Date, Length: 23412, dtype: object
```

Cómo hacer coincidir y reemplazar grupos con expresiones regulares

In [5]:
```
import pandas as pd

df = pd.read_csv(f'..\data/internshala_dataset_raw.csv')
df.head()
```

Out[5]:

	internship	company_name	skills	perks	location	duration	stipend	applicants	ifSkillsorPerksMissingUseThis
0	Software Testing	Times Internet	Software Testing	Certificate, 5 days a week	Noida	6 Months	8000 /month	119 applicants	Software Testing, Certificate\n5 days a week
1	Technical Operations - Networking And Monitoring	Paytm Payments Bank	Java, SQL, Unix, Oracle, MS SQL Server, Hibern...	Certificate, Letter of recommendation, 5 days ...	Noida	6 Months	10000 /month	194 applicants	Java\nSQL\nUnix\nOracle\nMS SQL Server\nHibern...
2	Software Project Management	IIT Bombay	English Proficiency (Spoken), English Proficie...	Certificate, Letter of recommendation, Flexibl...	Work From Home	6 Months	1000-2000 /month	113 applicants	English Proficiency (Spoken)\nEnglish Proficie...
3	Web Development	IIT Bombay	HTML, CSS, Flask, Python, Django	Certificate, Letter of recommendation, Flexibl...	Work From Home	6 Months	1000-2000 /month	183 applicants	HTML\nCSS\nFlask\nPython\nDjango, Certificate\...
4	Front End Development	IIT Bombay	HTML, CSS, JavaScript, ReactJS, Redux	Certificate, Letter of recommendation, Flexibl...	Work From Home	6 Months	1000-2000 /month	205 applicants	HTML\nCSS\nJavaScript\nReactJS\nRedux, Certifi...

Reemplazar con captura de grupo

In [6]:
```
df['applicants'].replace(to_replace=r"([0-9,\.]+)(.*)", value=r"\1", regex=True)
```

Out[6]:
```
0                        119
1                        194
2                        113
3                        183
4                        205
                ...
2562                      30
2563    Be an early applicant
2564    Be an early applicant
2565    Be an early applicant
2566    Be an early applicant
Name: applicants, Length: 2567, dtype: object
```

Esto significa lo siguiente:

- (la apertura de un parentesis indica que comienza un "grupo de captura". Lo que haya dentro de los paréntesis será "ca pturado" cuando la expresión regular encaje con algo de lo que había en la línea. Esa "captura" estará disponible para us arse en la cadena de sustitución, donde se ponga \1.

- El carácter \ es especial e indica que para interpretarlo hay que seguir mirando el siguiente carácter.

-) cierra el grupo de captura.

Por ejemplo, en la cadena "Esto es una prueba", el primero sería en la 'o' de Esto, pues esa o es un carácter no-blanco seguido de dos o más espacios. La expresión encajaría con las subcadenas marcadas en las siguiente figura:

```
Esto  es  una  prueba
   ^^^^^^ ^^^  ^^^^^
```

En el primer match tendría seis caracteres: 'o' (una o y cinco espacios) y el grupo de captura asociado a ese match sería la 'o'.

El siguiente parámetro de re.sub() vale r"\1 ", que significa: Sustituir el match que se haya obtenido por una cadena fo rmada por el contenido del grupo de captura y un espacio. Por tanto 'o ' será sustituido por 'o ' (la o sería el \1). Análogamente 's ' será sustituido por 's ', etc. De este modo se sustituyen varios espacios por uno solo, pero sólo si esos espacios iban después de algo que no era espacio.

Cómo hacer coincidir y reemplazar grupos con expresiones regulares: patrones de cadena

In [7]:
```
df['location']
```

Out[7]:
```
0               Noida
1               Noida
2      Work From Home
3      Work From Home
4      Work From Home
             ...
2562         Lucknow
2563           Delhi
2564    Work From Home
2565         Nainital
2566         Nainital
Name: location, Length: 2567, dtype: object
```

In [8]:
```
df['location'].str.replace(r'(.*) (.*) (.*)', r"\3", regex=True)
```

Out[8]:
```
0          Noida
1          Noida
2           Home
3           Home
4           Home
          ...
2562     Lucknow
2563       Delhi
2564        Home
2565    Nainital
2566    Nainital
Name: location, Length: 2567, dtype: object
```

In [9]:
```
df['skills']
```

Out[9]:
```
0                                    Software Testing
1       Java, SQL, Unix, Oracle, MS SQL Server, Hibern...
2       English Proficiency (Spoken), English Proficie...
3                         HTML, CSS, Flask, Python, Django
4                    HTML, CSS, JavaScript, ReactJS, Redux
                              ...
2562    AutoCAD, Autodesk Inventor, Arduino, Circuit D...
2563                                                  NaN
2564                                                  NaN
2565                                                  NaN
2566                                                  NaN
Name: skills, Length: 2567, dtype: object
```

In [10]:
```
df['skills'].str.replace(r'(.*)(\(.*\))(.*)', r"\1\3", regex=True)
```

Out[10]:
```
0                                    Software Testing
1       Java, SQL, Unix, Oracle, MS SQL Server, Hibern...
2       English Proficiency (Spoken), English Proficie...
3                         HTML, CSS, Flask, Python, Django
4                    HTML, CSS, JavaScript, ReactJS, Redux
                              ...
2562    AutoCAD, Autodesk Inventor, Arduino, Circuit D...
2563                                                  NaN
2564                                                  NaN
2565                                                  NaN
2566                                                  NaN
Name: skills, Length: 2567, dtype: object
```

In [11]:
```
df['skills'].str.replace(r'(.*)(\(.*\))(.*)', r"\2", regex=True)
```

Out[11]:
```
0                            Software Testing
1                                     (Java)
2                                   (Written)
3              HTML, CSS, Flask, Python, Django
4         HTML, CSS, JavaScript, ReactJS, Redux
                          ...
2562                                   (IoT)
2563                                     NaN
2564                                     NaN
2565                                     NaN
2566                                     NaN
Name: skills, Length: 2567, dtype: object
```

In [12]:
```
df['skills'].str.replace(r'(.*?)(\(.*?\))(.*)', r"\2", regex=True)
```

Out[12]:
```
0                            Software Testing
1                                     (Java)
2                                   (Spoken)
3              HTML, CSS, Flask, Python, Django
4         HTML, CSS, JavaScript, ReactJS, Redux
                          ...
2562                                   (IoT)
2563                                     NaN
2564                                     NaN
2565                                     NaN
2566                                     NaN
Name: skills, Length: 2567, dtype: object
```

Cómo hacer coincidir y reemplazar grupos con expresiones regulares: patrones numéricos

In [13]:
```
1 df['stipend']
```

Out[13]:
```
0            8000 /month
1           10000 /month
2       1000-2000 /month
3       1000-2000 /month
4       1000-2000 /month
              ...
2562    2000-5000 /month
2563         5000 /month
2564        Not provided
2565        10000 /month
2566        10000 /month
Name: stipend, Length: 2567, dtype: object
```

In [14]:
```
1 df['stipend'].str.replace(r'(\d+)-(\d+)(.*)', r"\2\3", regex=True)
```

Out[14]:
```
0        8000 /month
1       10000 /month
2        2000 /month
3        2000 /month
4        2000 /month
            ...
2562     5000 /month
2563     5000 /month
2564    Not provided
2565    10000 /month
2566    10000 /month
Name: stipend, Length: 2567, dtype: object
```

In [15]:
```
1 df['stipend'].str.replace(r'(\d+)-(\d+)(.*)', r"\1\3", regex=True)
```

Out[15]:
```
0        8000 /month
1       10000 /month
2        1000 /month
3        1000 /month
4        1000 /month
            ...
2562     2000 /month
2563     5000 /month
2564    Not provided
2565    10000 /month
2566    10000 /month
Name: stipend, Length: 2567, dtype: object
```