

In [1]:

```
import pandas as pd # import biblioteki
```

In [2]:

```
"""plik 'pokemon_data.csv' proszę zapisać w tym samym katalogu w którym będzie zapisany
wasz obecny projekt"""
data=pd.read_csv("pokemon_data.csv")# importowanie pliku i przypisanie go do zmiennej
                                     #read_csv("Tu należy podać dokładny adres pliku")
                                     #W tym przypadku plik jest zapisany w tym samym katalog
u co projekt nad który teraz pracujemy

'''komenda data.head() pozwala podejrzeć jak wyglądają dane. Zostaną wyświetlane pierw
sze 5 wierszy
można to zmienić: np data.head(20)'''

data.head()
```

Out[2]:

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Leg.
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	

Wyciąganie informacji o danych

In [3]:

```
"""ilość wierszy i kolumn"""
data.shape
```

Out[3]:

(800, 12)

In [4]:

```
"""ilość wierszy"""
len(data)
```

Out[4]:

800

In [5]:

```
"""numery wierszy"""
data.index
```

Out[5]:

```
RangeIndex(start=0, stop=800, step=1)
```

In [6]:

```
"""poniższa komenda wypisuje typy danych w poszczególnych kolumnach """
data.dtypes
```

Out[6]:

```
#          int64
Name      object
Type 1    object
Type 2    object
HP        int64
Attack    int64
Defense   int64
Sp. Atk   int64
Sp. Def   int64
Speed     int64
Generation int64
Legendary bool
dtype: object
```

In [7]:

```
data.describe()
```

Out[7]:

	#	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed
count	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000	800.000000
mean	362.813750	69.258750	79.001250	73.842500	72.820000	71.902500	68.277500
std	208.343798	25.534669	32.457366	31.183501	32.722294	27.828916	29.060474
min	1.000000	1.000000	5.000000	5.000000	10.000000	20.000000	5.000000
25%	184.750000	50.000000	55.000000	50.000000	49.750000	50.000000	45.000000
50%	364.500000	65.000000	75.000000	70.000000	65.000000	70.000000	65.000000
75%	539.250000	80.000000	100.000000	90.000000	95.000000	90.000000	90.000000
max	721.000000	255.000000	190.000000	230.000000	194.000000	230.000000	180.000000

In [8]:

data.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 800 entries, 0 to 799
Data columns (total 12 columns):
#           800 non-null int64
Name       800 non-null object
Type 1     800 non-null object
Type 2     414 non-null object
HP         800 non-null int64
Attack     800 non-null int64
Defense    800 non-null int64
Sp. Atk    800 non-null int64
Sp. Def    800 non-null int64
Speed      800 non-null int64
Generation 800 non-null int64
Legendary  800 non-null bool
dtypes: bool(1), int64(8), object(3)
memory usage: 69.6+ KB
```

In [9]:

```
"""otrzymujemy nawzy kolumn"""
data.columns
```

Out[9]:

```
Index(['#', 'Name', 'Type 1', 'Type 2', 'HP', 'Attack', 'Defense', 'Sp. Atk',
      'Sp. Def', 'Speed', 'Generation', 'Legendary'],
      dtype='object')
```

In [10]:

```
"""wyciągnięcie samych danych bez numerów wierszy i nazw kolumn"""
data.values
```

Out[10]:

```
array([[1, 'Bulbasaur', 'Grass', ..., 45, 1, False],
       [2, 'Ivysaur', 'Grass', ..., 60, 1, False],
       [3, 'Venusaur', 'Grass', ..., 80, 1, False],
       ...,
       [720, 'HoopaHoopa Confined', 'Psychic', ..., 70, 6, True],
       [720, 'HoopaHoopa Unbound', 'Psychic', ..., 80, 6, True],
       [721, 'Volcanion', 'Fire', ..., 70, 6, True]], dtype=object)
```

In [11]:

```
"""opisy kolumn, numery wierszy i czyste dane, można przypisać do zmiennych"""
columny=data.columns
nr_wierszy=data.index
dane=data.values
```

In [12]:

```
print (columnny)
"""poniżej przedział od elementu 0 do 3 """
print(columnny[0:3])
```

```
Index(['#', 'Name', 'Type 1', 'Type 2', 'HP', 'Attack', 'Defense', 'Sp. At
k',
      'Sp. Def', 'Speed', 'Generation', 'Legendary'],
      dtype='object')
Index(['#', 'Name', 'Type 1'], dtype='object')
```

In [13]:

```
"""zmiana nazw kolumn"""
opis_kolumn=[]#tworzymy listę z nowymi nazwami kolumn
kolumny=data.columns
for i in range(len(kolumny)):#pętla wykona się tyle razy ile mamy kolumn w naszych dany
ch
    nowa_nazwa="pokemon "+str(i)#tworzymy nową nazwę dla kolumny i
    opis_kolumn.append(nowa_nazwa)#dodajemy nową nazwę do listy
print (opis_kolumn)
```

```
['pokemon 0', 'pokemon 1', 'pokemon 2', 'pokemon 3', 'pokemon 4', 'pokemon
5', 'pokemon 6', 'pokemon 7', 'pokemon 8', 'pokemon 9', 'pokemon 10', 'pok
emon 11']
```

In [14]:

```
"""teraz możemy zamienić nazwy kolumn w naszych danych"""
data.columns=opis_kolumn
data.head()
```

Out[14]:

	pokemon 0	pokemon 1	pokemon 2	pokemon 3	pokemon 4	pokemon 5	pokemon 6	pokemon 7
0	1	Bulbasaur	Grass	Poison	45	49	49	65
1	2	Ivysaur	Grass	Poison	60	62	63	80
2	3	Venusaur	Grass	Poison	80	82	83	100
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122
4	4	Charmander	Fire	NaN	39	52	43	60

In [15]:

```
"""podobnie możemy zrobić z opisami wierszy wierszami """
#to jest twojej zadanie
#zmień indeksy wierszy na """wiersz 0, wiersz 1"""
```

Out[15]:

```
'podobnie możemy zrobić z opisami wierszy wierszami '
```

In [16]:

```
"""wczytamy ponownie dane """
data=pd.read_csv("pokemon_data.csv")#przywróciliśmy pierwotne dane
data.head()
```

Out[16]:

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Leg.
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	

odczytywanie danych

In [17]:

```
""" odszukiwanie konkretnych wierszy i kolumn"""
#czytanie konkretnej kolumny
data.Name.head()#podajemy nazwę kolumny. ".head()" -> powoduje wyświetlenie tylko 5 wierszy
```

Out[17]:

```
0      Bulbasaur
1      Ivysaur
2      Venusaur
3  VenusaurMega Venusaur
4      Charmander
Name: Name, dtype: object
```

In [18]:

```
data[['Name', 'Type 1', 'HP']].head()#podajemy nazwy kolumn w liście poprzez dodatkowe klamry"[ ]"
```

Out[18]:

	Name	Type 1	HP
0	Bulbasaur	Grass	45
1	Ivysaur	Grass	60
2	Venusaur	Grass	80
3	VenusaurMega Venusaur	Grass	80
4	Charmander	Fire	39

In [19]:

```

"""metoda 'loc' polega na podawaniu nazw kolumn i indeksów wierszy"""
wiersze=range(0,50,2)
kolumny=['Name','Type 1','HP','Speed']
new_data=data.loc[wiersze,kolumny]
print(new_data.head())
new_data.shape# widać że nowe dane mają zdecydowanie mniejszy rozmiar (25,4)

```

	Name	Type 1	HP	Speed
0	Bulbasaur	Grass	45	45
2	Venusaur	Grass	80	80
4	Charmander	Fire	39	65
6	Charizard	Fire	78	100
8	CharizardMega	Charizard Y	Fire	78

Out[19]:

(25, 4)

In [20]:

```

"""wybieranie przedziałów"""
data.loc[0:50,'Name':'HP']#wiersze od 0 do 50 kolumny od "Name" do "HP"
data.loc[:,2,'Name':'HP':2].head()#co drógi wiersz, kolumny od "Name" do "HP" wybrane c
o dróga

```

Out[20]:

	Name	Type 2
0	Bulbasaur	Poison
2	Venusaur	Poison
4	Charmander	NaN
6	Charizard	Flying
8	CharizardMega	Charizard Y

In [21]:

```

"""drugi sposób wybierania wierszy metoda 'iloc' podajemy konkretne współrzędne """
#czytanie konkretnego wiersza
data.iloc[1:4] #trzy wiersze od 1 do 4
data.iloc[:-4]#wszystkie oprócz 4 ostatnich
data.iloc[5:] #od piątego do ostatniego
#odczytywanie konkretnej lokacji
data.iloc[4,1]#podaje konkretną współrzędną

```

Out[21]:

'Charmander'

wprowadzenie warónków

In [22]:

```

"""możemy przeszukiwać dane wprowadzając różne warunki """
war1=data["Type 1"]=="Grass"#
war2=data["Attack">100
war3=data["Legendary"]==True
data[war1]# uzyskamy wszystkie wiersze, które w kolumnie "Type 1" mają "Grass"
data[war1&war2]#uzyskam wszystkie wiersze które spełniają warunek 1 i 2
data[war1|war3]# wszystkie wiersze, które spełniają warunek 1 lub 3
data[~war3].head()# znak "~" odwraca warunek uzyskujemy wszystkie wiersze które nie spełniają war3

```

Out[22]:

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legend
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	

In [23]:

```

#pokemon.loc [(pokemon['Name'].str.contains('Mega'))]
#pokemon.loc [~(pokemon['Name'].str.contains('Mega'))]
data.loc[(data["Name"].str.contains("Mega"))]#wyszukuje wszystkie pokemony które mają w nazwie słowo "Mega"
data.loc[~(data["Name"].str.contains("Mega"))].head() #zaprzeczamy powyższy warunek używając "~"

```

Out[23]:

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legend
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	F
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	F
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1	F
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1	F
5	5	Charmeleon	Fire	NaN	58	64	58	80	65	80	1	F

In [24]:

```
data.sort_values('Name') #alfabetycznie
data.sort_values('Name',ascending=False) #alfabetycznie odwrócona kolejność
data.sort_values(['Type 1', "HP"], ascending=[1,0]).head() #sortowanie po wielu kolumnach
# ( ascending=[1,0]) podaliśmy czy sortowanie w konkretnej kolumnie ma być w kolejności rosnącej "1" czy malejącej "0"
```

Out[24]:

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation
520	469	Yanmega	Bug	Flying	86	76	86	116	56	95	4
698	637	Volcarona	Bug	Fire	85	60	65	135	105	100	5
231	214	Heracross	Bug	Fighting	80	125	75	40	95	85	2
232	214	HeracrossMega Heracross	Bug	Fighting	80	185	115	40	105	75	2
678	617	Accelgor	Bug	NaN	80	70	40	100	60	145	5

In [25]:

```
"""dodawanie nowej kolumny"""
data["nazwa nowej kolumny"]=0
data.head()
```

Out[25]:

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Leg
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1

In [26]:

```
data.rename(columns={'nazwa nowej kolumny':'suma'}, inplace=True)# Zmieniamy nazwę jedn
ej kolumny
data["suma"]=data.iloc[:,4:11].sum(axis=1)#do każdej komórki w kolumnie "suma" wpisujem
y wartość sumy komórek
#z kolumn od "hp" do "generation". Sumowanie odbywa się wzdłuż wierszy usta
wiliśmy to parametrem axis=1
data["HP*Speed"]=data["HP"]*data["Speed"]
data.head()
```

Out[26]:

#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Leg
0	1	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1
1	2	Ivysaur	Grass	Poison	60	62	63	80	80	60	1
2	3	Venusaur	Grass	Poison	80	82	83	100	100	80	1
3	3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1
4	4	Charmander	Fire	NaN	39	52	43	60	50	65	1

In [27]:

```
"""usuwanie kolumn"""
data=data.drop(columns=['#'])
data.head()
```

Out[27]:

	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legend
0	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	Fa
1	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	Fa
2	Venusaur	Grass	Poison	80	82	83	100	100	80	1	Fa
3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	Fa
4	Charmander	Fire	NaN	39	52	43	60	50	65	1	Fa

In [28]:

```

"""zmiana kolejności kolumn"""
kolumny=list(data.columns)# w zmiennej „kolumny” tworzymy listę z opisami kolumn
data[[kolumny[-2]]+kolumny[1:4]+kolumny[3:10]+[kolumny[-1]]]# tworzymy nowy układ kolumn
#powyżej używamy dodatkowych "[" żeby zrobić listę
# jeśli chcemy zachować nowy układ kolumn to musimy przypisać go do zmiennej
data1=data[[kolumny[-2]]+kolumny[0:4]+kolumny[3:10]+[kolumny[-1]]].head()
data1.head()

```

Out[28]:

	suma	Name	Type 1	Type 2	HP	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generat
0	319	Bulbasaur	Grass	Poison	45	45	49	49	65	65	45	
1	406	Ivysaur	Grass	Poison	60	60	62	63	80	80	60	
2	526	Venusaur	Grass	Poison	80	80	82	83	100	100	80	
3	626	VenusaurMega Venusaur	Grass	Poison	80	80	100	123	122	120	80	
4	310	Charmander	Fire	NaN	39	39	52	43	60	50	65	

In [29]:

```

data.loc[data['Type 1']=="Fire", "Type 1"]="Flamer"#zmieniamy w kolumnie "Type 1" zawartość
każdej kolorki w której jest "Fier"
#na słowo "Flamer"
#komenda działa na zasadzie data.loc[warunek, nazwa kolumny lub lista nazw kolumn]
data.head()

```

Out[29]:

	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legend
0	Bulbasaur	Grass	Poison	45	49	49	65	65	45	1	F
1	Ivysaur	Grass	Poison	60	62	63	80	80	60	1	F
2	Venusaur	Grass	Poison	80	82	83	100	100	80	1	F
3	VenusaurMega Venusaur	Grass	Poison	80	100	123	122	120	80	1	F
4	Charmander	Flamer	NaN	39	52	43	60	50	65	1	F

Grupowanie danych

In [30]:

```

"""do grupowania przydatna jest metoda groupby()...."""
data.groupby("Type 1").count().head() #zlicza ilość komórek zawierające dane
#.sum() -> sumuje komórki
#.mean() -> średnia

```

Out[30]:

	Name	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary	suma
Type 1											
	Bug	69	52	69	69	69	69	69	69	69	69
	Dark	31	21	31	31	31	31	31	31	31	31
	Dragon	32	21	32	32	32	32	32	32	32	32
	Electric	44	17	44	44	44	44	44	44	44	44
	Fairy	17	2	17	17	17	17	17	17	17	17

In [31]:

```
data.groupby(["Type 1", "Type 2", "Generation"]).count().head(20) # można grupować według kilku kolumn wystarczy podać je w liście
```

Out[31]:

		Name	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Legendary	suma
Type 1	Type 2	Generation								
Bug	Electric	5	2	2	2	2	2	2	2	2
		2	2	2	2	2	2	2	2	2
	Fire	5	2	2	2	2	2	2	2	2
		1	3	3	3	3	3	3	3	3
	Flying	2	3	3	3	3	3	3	3	3
		3	3	3	3	3	3	3	3	3
		4	4	4	4	4	4	4	4	4
		6	1	1	1	1	1	1	1	1
	Ghost	3	1	1	1	1	1	1	1	1
		1	2	2	2	2	2	2	2	2
	Grass	4	1	1	1	1	1	1	1	1
		5	3	3	3	3	3	3	3	3
	Ground	3	1	1	1	1	1	1	1	1
		4	1	1	1	1	1	1	1	1
	Poison	1	6	6	6	6	6	6	6	6
		2	2	2	2	2	2	2	2	2
		3	1	1	1	1	1	1	1	1
		5	3	3	3	3	3	3	3	3
	Rock	2	1	1	1	1	1	1	1	1
		5	2	2	2	2	2	2	2	2



In [32]:

```
"""usuwanie brakujących danych"""  
data["Type 2"].isna # widać że w niektórych miejscach pojawiają się "NaN" oznacza to br  
ak danych
```

Out[32]:

<bound	method	Series.isna of 0	Poison
1	Poison		
2	Poison		
3	Poison		
4	NaN		
5	NaN		
6	Flying		
7	Dragon		
8	Flying		
9	NaN		
10	NaN		
11	NaN		
12	NaN		
13	NaN		
14	NaN		
15	Flying		
16	Poison		
17	Poison		
18	Poison		
19	Poison		
20	Flying		
21	Flying		
22	Flying		
23	Flying		
24	NaN		
25	NaN		
26	Flying		
27	Flying		
28	NaN		
29	NaN		
...			
770	NaN		
771	Flying		
772	Fairy		
773	Fairy		
774	NaN		
775	NaN		
776	NaN		
777	Fairy		
778	Grass		
779	Grass		
780	Grass		
781	Grass		
782	Grass		
783	Grass		
784	Grass		
785	Grass		
786	Grass		
787	Grass		
788	NaN		
789	NaN		
790	Dragon		
791	Dragon		
792	NaN		
793	Flying		
794	Ground		
795	Fairy		
796	Fairy		
797	Ghost		

```
798         Dark
799         Water
Name: Type 2, Length: 800, dtype: object>
```

In [33]:

```
"""czyszczenie danych i ich uzupełnianie"""
data["Type 2"].fillna("uzupełniona dana").head(10) # uzupełnianie danych jest dość dużym
obszarem biblioteki Panda
# tutaj jest zaprezentowany tylko drobny przykład
# w razie konieczności zachęcam do szukania informacji w
Internecie i dokumentacji biblioteki
```

Out[33]:

```
0         Poison
1         Poison
2         Poison
3         Poison
4  uzupełniona dana
5  uzupełniona dana
6         Flying
7         Dragon
8         Flying
9  uzupełniona dana
Name: Type 2, dtype: object
```

In [34]:

```
"""Powyższe uzupełnione dane nie zostały zapisane w zmiennej """
data.isna().sum() # ta komenda sprawdza gdzie występują brakujące dane i sumuje je w
dłuż każdej kolumny
# widzimy że brakujące dane występują tylko w kolumnie 'Type2'
```

Out[34]:

```
Name      0
Type 1    0
Type 2    386
HP         0
Attack    0
Defense   0
Sp. Atk   0
Sp. Def   0
Speed     0
Generation 0
Legendary 0
suma      0
HP*Speed   0
dtype: int64
```

In []: