TP1 : Pandas_for_Beginners_Part_1_DataFrame_Basics

```
import pandas as pd

import matplotlib.pyplot as plt
```

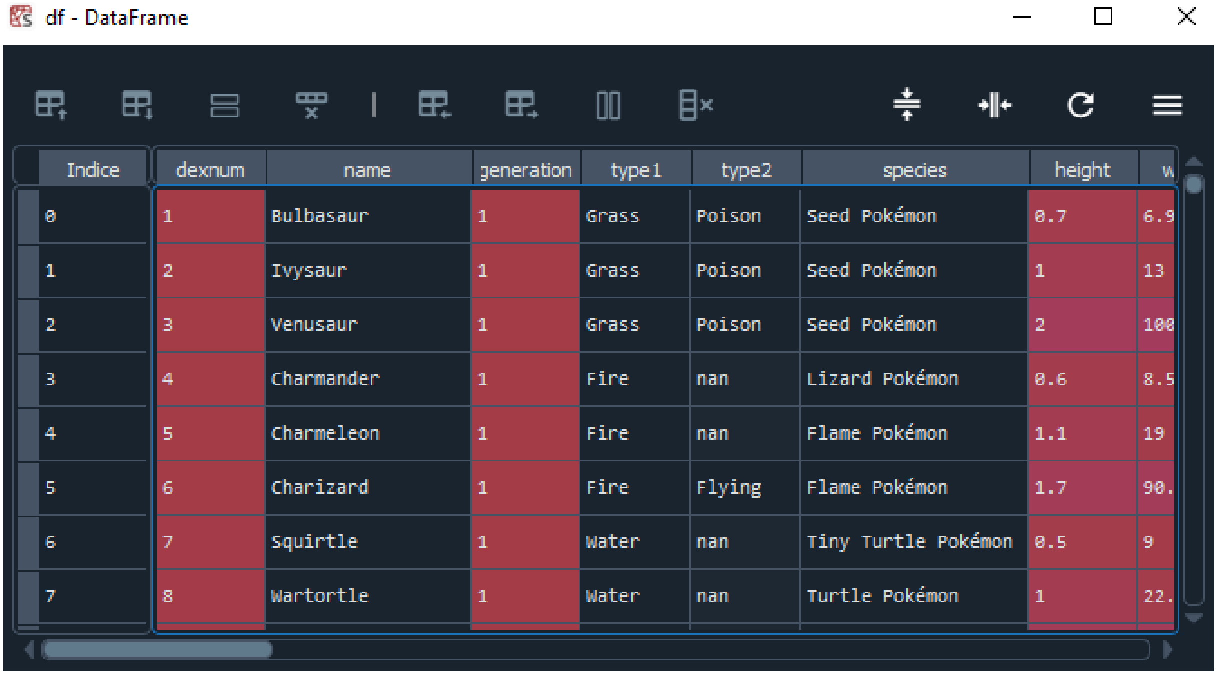
# Importing pandas

# Importing mathplotlib for visualization

Entrée:

```
df
```

Sortie:



df - DataFrame

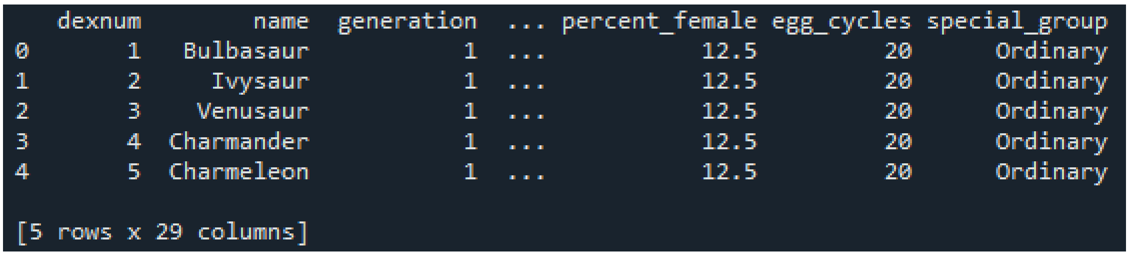
Indice	dexnum	name	generation	type1	type2	species	height	w
0	1	Bulbasaur	1	Grass	Poison	Seed Pokémon	0.7	6.9
1	2	Ivysaur	1	Grass	Poison	Seed Pokémon	1	13
2	3	Venusaur	1	Grass	Poison	Seed Pokémon	2	100
3	4	Charmander	1	Fire	nan	Lizard Pokémon	0.6	8.5
4	5	Charmeleon	1	Fire	nan	Flame Pokémon	1.1	19
5	6	Charizard	1	Fire	Flying	Flame Pokémon	1.7	90.
6	7	Squirtle	1	Water	nan	Tiny Turtle Pokémon	0.5	9
7	8	Wartortle	1	Water	nan	Turtle Pokémon	1	22.

entrée

```
df.head()
```

Sortie:

Le 5 premières lignes

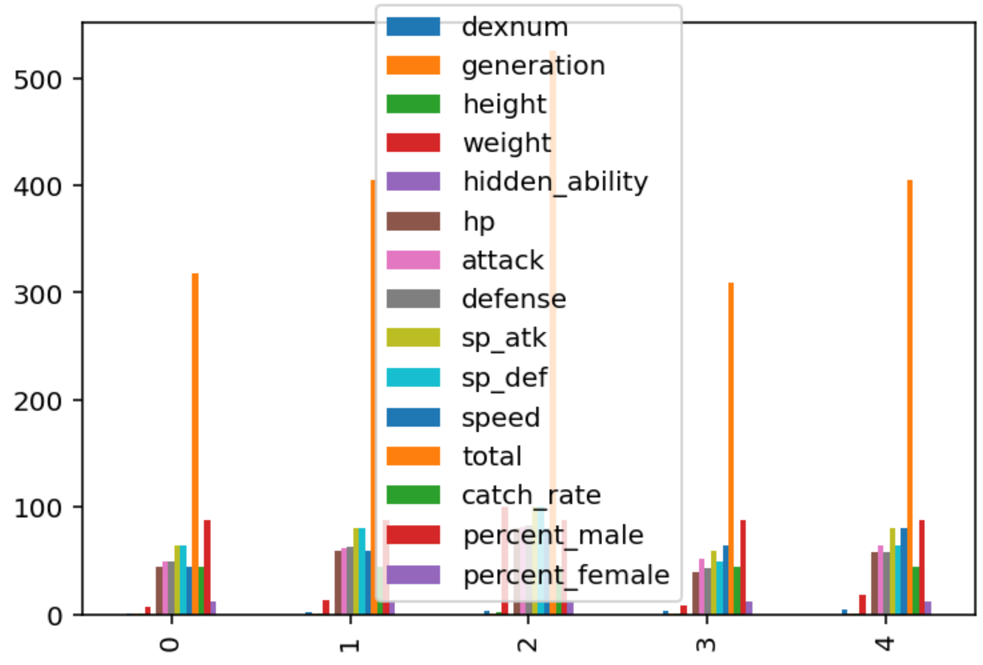


```
   dexnum        name  generation  ... percent_female egg_cycles special_group
0       1   Bulbasaur           1  ...           12.5         20      Ordinary
1       2     Ivysaur           1  ...           12.5         20      Ordinary
2       3    Venusaur           1  ...           12.5         20      Ordinary
3       4  Charmander           1  ...           12.5         20      Ordinary
4       5  Charmeleon           1  ...           12.5         20      Ordinary

[5 rows x 29 columns]
```

Entrée

```
df.head().plot.bar()
```

Sortie:

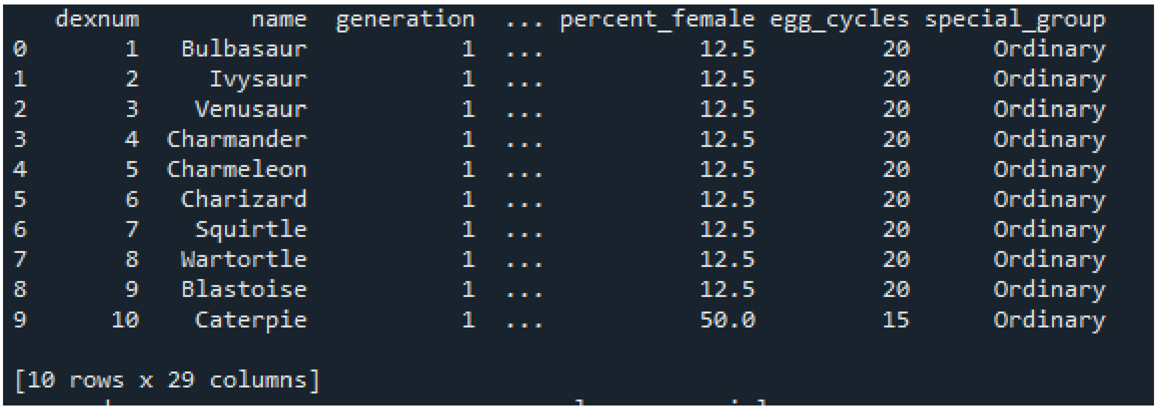


Entrée

```
print(df.head(10))
```

Sortie :

Les dix premières lignes



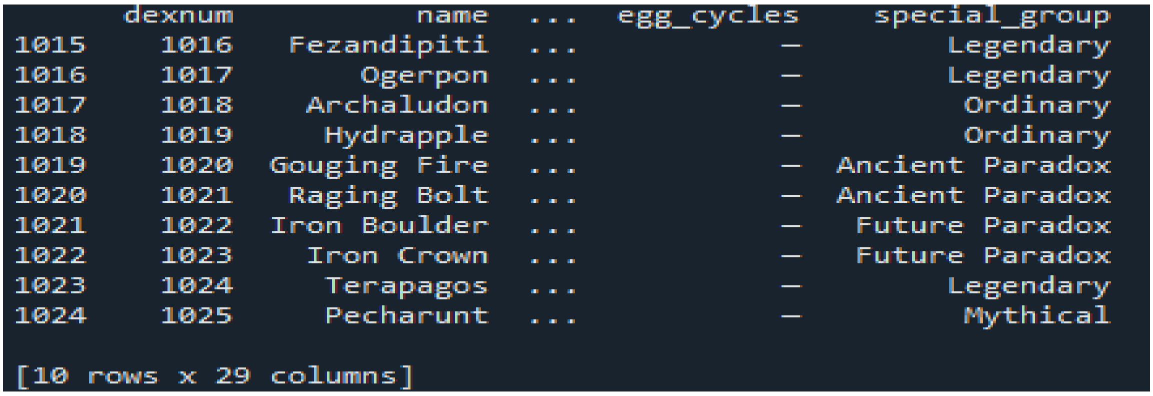
```
   dexnum        name  generation  ... percent_female egg_cycles special_group
0       1   Bulbasaur           1  ...           12.5         20      Ordinary
1       2     Ivysaur           1  ...           12.5         20      Ordinary
2       3    Venusaur           1  ...           12.5         20      Ordinary
3       4  Charmander           1  ...           12.5         20      Ordinary
4       5  Charmeleon           1  ...           12.5         20      Ordinary
5       6   Charizard           1  ...           12.5         20      Ordinary
6       7    Squirtle           1  ...           12.5         20      Ordinary
7       8   Wartortle           1  ...           12.5         20      Ordinary
8       9   Blastoise           1  ...           12.5         20      Ordinary
9      10    Caterpie           1  ...           50.0         15      Ordinary

[10 rows x 29 columns]
```

Entrée

```
df.tail(10)
```

Sortie

les dix dernières lignes

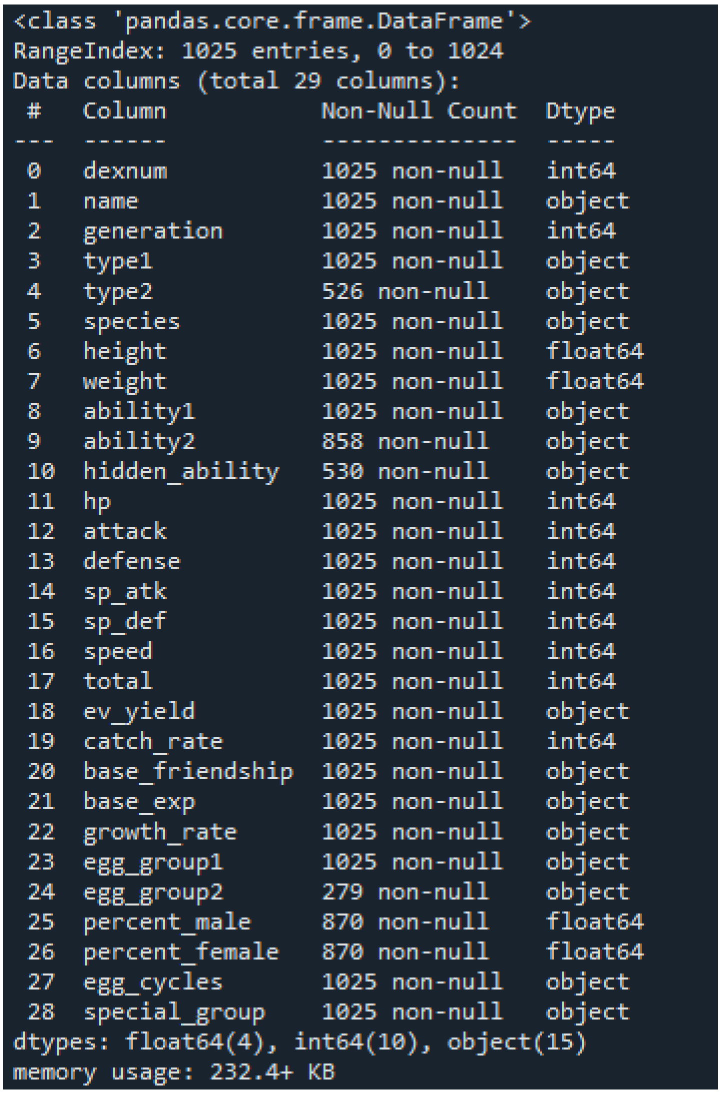


```
      dexnum           name  ... egg_cycles    special_group
1015    1016    Fezandipiti  ...          —        Legendary
1016    1017        Ogerpon  ...          —        Legendary
1017    1018     Archaludon  ...          —         Ordinary
1018    1019      Hydrapple  ...          —         Ordinary
1019    1020   Gouging Fire  ...          —  Ancient Paradox
1020    1021    Raging Bolt  ...          —  Ancient Paradox
1021    1022   Iron Boulder  ...          —   Future Paradox
1022    1023     Iron Crown  ...          —   Future Paradox
1023    1024      Terapagos  ...          —        Legendary
1024    1025      Pecharunt  ...          —         Mythical

[10 rows x 29 columns]
```

Entrée

```
df.info()
```

Sortie

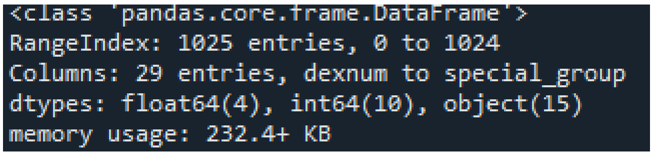


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 29 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   dexnum            1025 non-null   int64  
 1   name              1025 non-null   object 
 2   generation        1025 non-null   int64  
 3   type1             1025 non-null   object 
 4   type2             526 non-null    object 
 5   species           1025 non-null   object 
 6   height            1025 non-null   float64
 7   weight            1025 non-null   float64
 8   ability1          1025 non-null   object 
 9   ability2          858 non-null    object 
 10  hidden_ability    530 non-null    object 
 11  hp                1025 non-null   int64  
 12  attack            1025 non-null   int64  
 13  defense           1025 non-null   int64  
 14  sp_atk            1025 non-null   int64  
 15  sp_def            1025 non-null   int64  
 16  speed             1025 non-null   int64  
 17  total             1025 non-null   int64  
 18  ev_yield          1025 non-null   object 
 19  catch_rate        1025 non-null   int64  
 20  base_friendship   1025 non-null   object 
 21  base_exp          1025 non-null   object 
 22  growth_rate       1025 non-null   object 
 23  egg_group1        1025 non-null   object 
 24  egg_group2        279 non-null    object 
 25  percent_male      870 non-null    float64
 26  percent_female    870 non-null    float64
 27  egg_cycles        1025 non-null   object 
 28  special_group     1025 non-null   object 
dtypes: float64(4), int64(10), object(15)
memory usage: 232.4+ KB
```

Entrée

```
df.info(verbose=False)
```

sortie

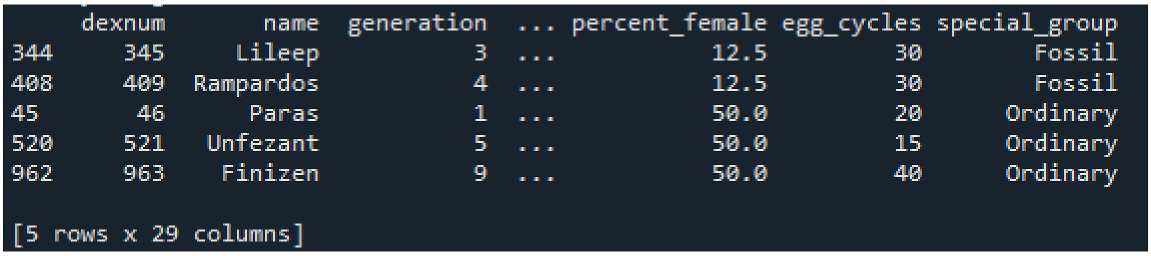


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Columns: 29 entries, dexnum to special_group
dtypes: float64(4), int64(10), object(15)
memory usage: 232.4+ KB
```

Entrée

```
print(df.sample(5))
```

Sortie

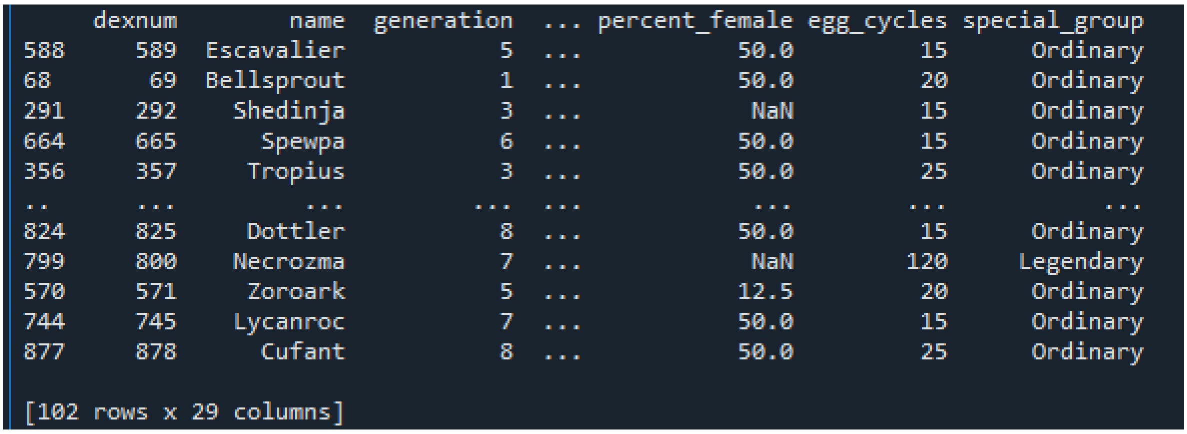


```
     dexnum       name  generation  ... percent_female egg_cycles special_group
344     345     Lileep           3  ...           12.5         30        Fossil
408     409  Rampardos           4  ...           12.5         30        Fossil
45       46      Paras           1  ...           50.0         20      Ordinary
520     521   Unfezant           5  ...           50.0         15      Ordinary
962     963    Finizen           9  ...           50.0         40      Ordinary

[5 rows x 29 columns]
```

Entrée

```
df.sample(frac = 0.1)
```

Sotie

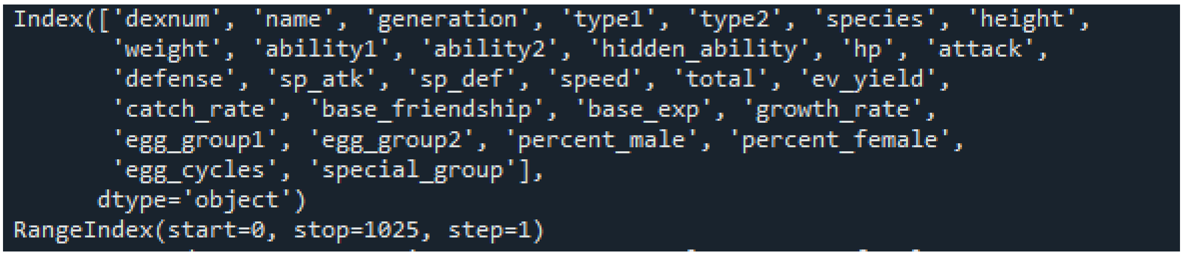


```
     dexnum        name  generation  ... percent_female egg_cycles special_group
588     589  Escavalier           5  ...           50.0         15      Ordinary
68       69  Bellsprout           1  ...           50.0         20      Ordinary
291     292    Shedinja           3  ...            NaN         15      Ordinary
664     665      Spewpa           6  ...           50.0         15      Ordinary
356     357     Tropius           3  ...           50.0         25      Ordinary
..      ...         ...         ...  ...            ...        ...           ...
824     825     Dottler           8  ...           50.0         15      Ordinary
799     800    Necrozma           7  ...            NaN        120     Legendary
570     571     Zoroark           5  ...           12.5         20      Ordinary
744     745    Lycanroc           7  ...           50.0         15      Ordinary
877     878      Cufant           8  ...           50.0         25      Ordinary

[102 rows x 29 columns]
```

Entrée

```
print(df.columns)
```

sortie



```
Index(['dexnum', 'name', 'generation', 'type1', 'type2', 'species', 'height',
       'weight', 'ability1', 'ability2', 'hidden_ability', 'hp', 'attack',
       'defense', 'sp_atk', 'sp_def', 'speed', 'total', 'ev_yield',
       'catch_rate', 'base_friendship', 'base_exp', 'growth_rate',
       'egg_group1', 'egg_group2', 'percent_male', 'percent_female',
       'egg_cycles', 'special_group'],
      dtype='object')
RangeIndex(start=0, stop=1025, step=1)
```

Entrée

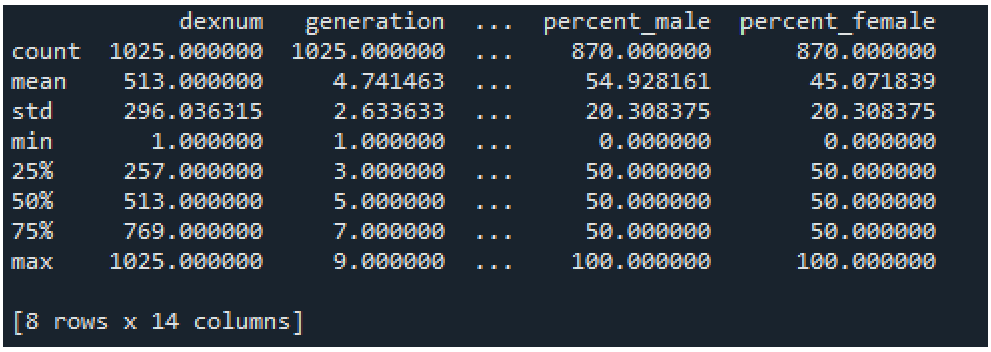
```
print(df.infex)
```

Sortie

```
Index(['dexnum', 'name', 'generation', 'type1', 'type2', 'species', 'height',
       'weight', 'ability1', 'ability2', 'hidden_ability', 'hp', 'attack',
       'defense', 'sp_atk', 'sp_def', 'speed', 'total', 'ev_yield',
       'catch_rate', 'base_friendship', 'base_exp', 'growth_rate',
       'egg_group1', 'egg_group2', 'percent_male', 'percent_female',
       'egg_cycles', 'special_group'],
      dtype='object')
RangeIndex(start=0, stop=1025, step=1)
```

```
print(df.describe())
```

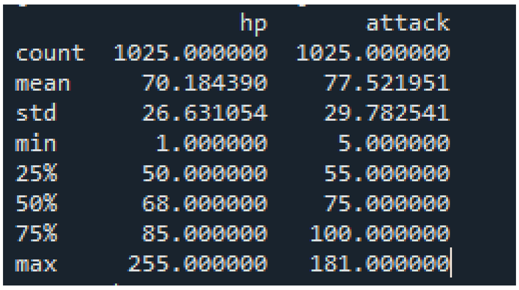
Sortie



```
            dexnum   generation  ...  percent_male  percent_female
count  1025.000000  1025.000000  ...    870.000000      870.000000
mean    513.000000     4.741463  ...     54.928161       45.071839
std     296.036315     2.633633  ...     20.308375       20.308375
min       1.000000     1.000000  ...      0.000000        0.000000
25%     257.000000     3.000000  ...     50.000000       50.000000
50%     513.000000     5.000000  ...     50.000000       50.000000
75%     769.000000     7.000000  ...     50.000000       50.000000
max    1025.000000     9.000000  ...    100.000000      100.000000

[8 rows x 14 columns]
```

```
print(df[['hp', 'attack']].describe())
```

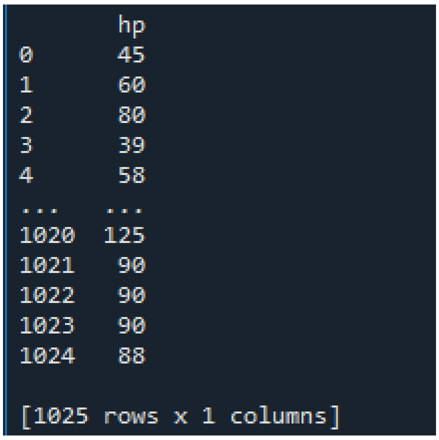
Sortie



```
                hp       attack
count  1025.000000  1025.000000
mean     70.184390    77.521951
std      26.631054    29.782541
min       1.000000     5.000000
25%      50.000000    55.000000
50%      68.000000    75.000000
75%      85.000000   100.000000
max     255.000000   181.000000
```

```
print(df[['hp']]) # Shows the DataFrame of "hp" column
```

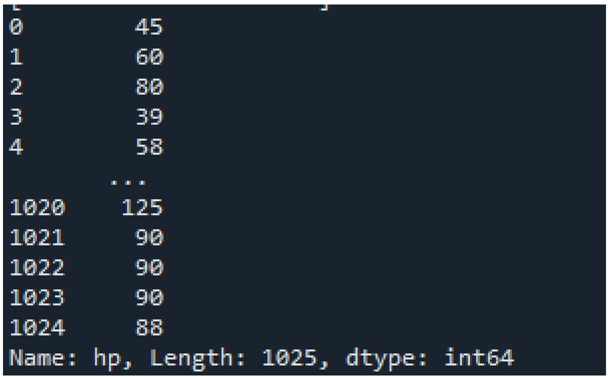
Sotie



```
        hp
0       45
1       60
2       80
3       39
4       58
...    ...
1020   125
1021    90
1022    90
1023    90
1024    88

[1025 rows x 1 columns]
```

```
print(df['hp']) # Shows the DataFrame of "HP" column
```

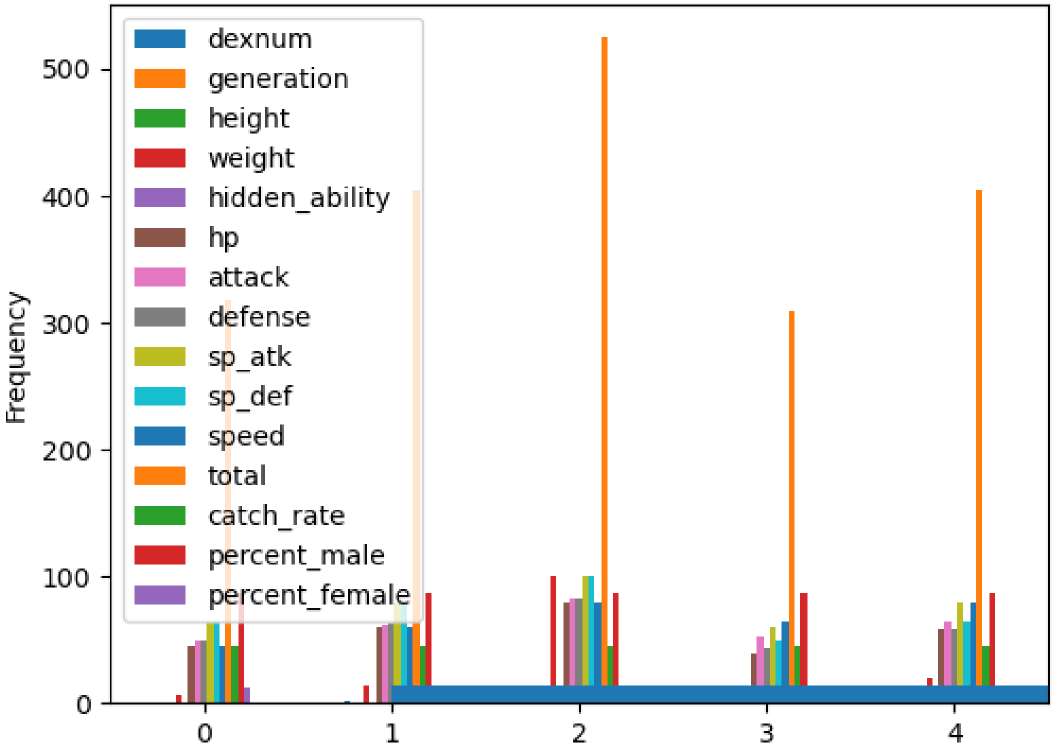
Sortie



```
0         45
1         60
2         80
3         39
4         58
        ...
1020     125
1021      90
1022      90
1023      90
1024      88
Name: hp, Length: 1025, dtype: int64
```

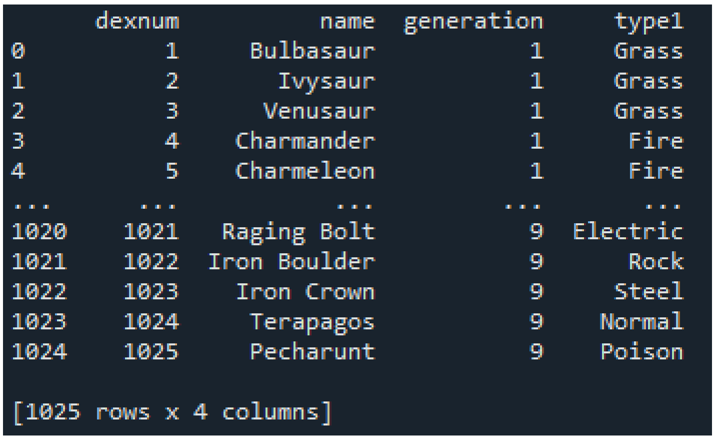
```
df['hp'].plot.hist()

plt.show() # Shows the histogram of DataFrame in "HP" column
```

Sortie



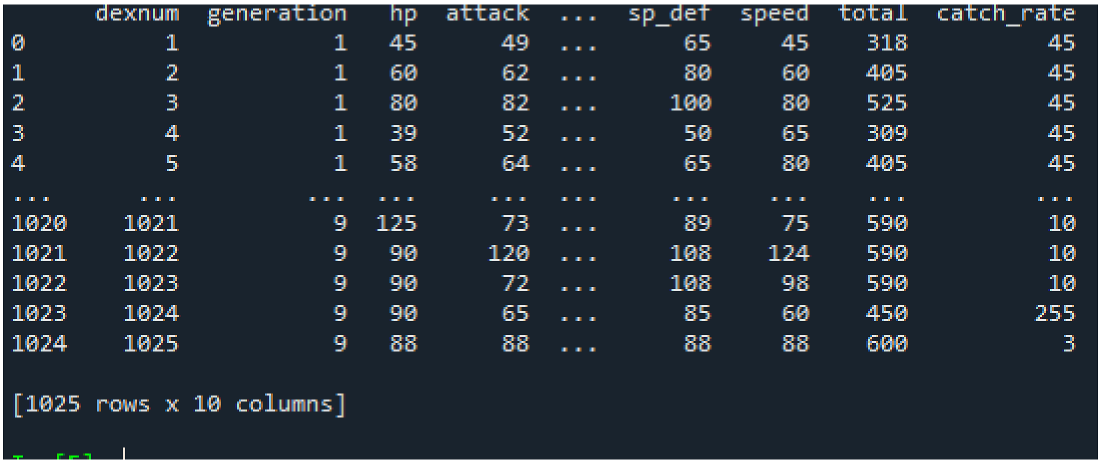
```
print(df[df.columns[:4]])
```

Sortie



```
      dexnum          name  generation     type1
0          1     Bulbasaur           1     Grass
1          2       Ivysaur           1     Grass
2          3      Venusaur           1     Grass
3          4    Charmander           1      Fire
4          5    Charmeleon           1      Fire
...      ...           ...         ...       ...
1020    1021   Raging Bolt           9  Electric
1021    1022  Iron Boulder           9      Rock
1022    1023    Iron Crown           9     Steel
1023    1024     Terapagos           9    Normal
1024    1025     Pecharunt           9    Poison

[1025 rows x 4 columns]
```

```
print(df.select_dtypes('int')) # here we see rows only with filtering columns as intege
```



```
      dexnum  generation   hp  attack  ...  sp_def  speed  total  catch_rate
0          1           1   45      49  ...      65     45    318          45
1          2           1   60      62  ...      80     60    405          45
2          3           1   80      82  ...     100     80    525          45
3          4           1   39      52  ...      50     65    309          45
4          5           1   58      64  ...      65     80    405          45
...      ...         ...  ...     ...  ...     ...    ...    ...         ...
1020    1021           9  125      73  ...      89     75    590          10
1021    1022           9   90     120  ...     108    124    590          10
1022    1023           9   90      72  ...     108     98    590          10
1023    1024           9   90      65  ...      85     60    450         255
1024    1025           9   88      88  ...      88     88    600           3

[1025 rows x 10 columns]

In [5]:
```