

Acceso a los elementos de un DataFrame

```
In [22]: # Creación de un DataFrame e inicializarlo con un diccionario de Python.
Personas = {
    "Peso": pd.Series([72, 60, 74, 73], ["Emilio", "Anel", "Chucho", "Jocelin"]),
    "Altura": pd.Series({"Emilio": 169, "Anel": 145, "Chucho": 170, "Jocelin": 170}),
    "Mascotas": pd.Series([2, 9], ["Anel", "Jocelin"])
}

df = pd.DataFrame(Personas)
df
```

```
Out[22]:
```

	Peso	Altura	Mascotas
Anel	60	145	2.0
Chucho	74	170	NaN
Emilio	72	169	NaN
Jocelin	73	170	9.0

Acceso a los elementos de las columnas del DataFrame.

```
In [23]: df["Peso"]
```

```
Out[23]: Anel      60
Chucho    74
Emilio    72
Jocelin   73
Name: Peso, dtype: int64
```

```
In [24]: df[["Peso", "Altura"]]
```

```
Out[24]:
```

	Peso	Altura
Anel	60	145
Chucho	74	170
Emilio	72	169
Jocelin	73	170

```
In [25]: # Pueden convivir los elementos anteriores con expresiones booleanas.
df["Peso"] > 73
```

```
Out[25]: Anel      False
Chucho    True
Emilio    False
Jocelin   False
Name: Peso, dtype: bool
```

```
In [27]: # Pueden convivir los métodos anteriores con expresiones booleanas y mostrar el DataFrame
df[df["Peso"] > 72]
```

```
Out[27]:
```

	Peso	Altura	Mascotas
Chucho	74	170	NaN
Jocelin	73	170	9.0

Accediendo a los elementos de las filas del DataFrame

```
In [28]: # Mostrar el DataFrame
df
```

```
Out[28]:
```

	Peso	Altura	Mascotas
Anel	60	145	2.0
Chucho	74	170	NaN
Emilio	72	169	NaN
Jocelin	73	170	9.0

```
In [29]: df.loc["Emilio"]
```

```
Out[29]: Peso      72.0
Altura    169.0
Mascotas    NaN
Name: Emilio, dtype: float64
```

```
In [30]: df.iloc[1:3]
```

```
Out[30]:
```

```
In [30]: df.iloc[1:3]
```

```
Out[30]:
```

	Peso	Altura	Mascotas
Chucho	74	170	NaN
Emilio	72	169	NaN

Consultas avanzada de los elementos de un DataFrame

```
In [31]: # Mostrar el DataFrame
df
```

```
Out[31]:
```

	Peso	Altura	Mascotas
Anel	60	145	2.0
Chucho	74	170	NaN
Emilio	72	169	NaN
Jocelin	73	170	9.0

```
In [34]: df.query("Altura >= 170 and Peso >= 73")
```

```
Out[34]:
```

	Peso	Altura	Mascotas
Chucho	74	170	NaN
Jocelin	73	170	9.0

Copiar un DataFrame

```
In [35]: # Crear un DataFrame e inicializarlo con un diccionario de objetos Series
Personas = {
    "Peso": pd.Series([72, 60, 74, 73], ["Emilio", "Anel", "Chucho", "Jocelin"]),
    "Altura": pd.Series({"Emilio": 169, "Anel": 145, "Chucho": 170, "Jocelin": 170}),
    "Mascotas": pd.Series([2, 9], ["Anel", "Jocelin"])
}

df = pd.DataFrame(Personas)
```

Modificación de un DataFrame

```
In [37]: # Añadir una nueva columna al DataFrame
df["Año_Nac"] = [2004, 2004, 2004, 2004]
df
```

```
Out[37]:
```

	Peso	Altura	Mascotas	Año_Nac
Anel	60	145	2.0	2004
Chucho	74	170	NaN	2004
Emilio	72	169	NaN	2004
Jocelin	73	170	9.0	2004

```
In [39]: # Añadir una nueva columna calculada al DataFrame.
df["Edad"] = 2024 - df["Año_Nac"]
df
```

```
Out[39]:
```

	Peso	Altura	Mascotas	Año_Nac	Edad
Anel	60	145	2.0	2004	20
Chucho	74	170	NaN	2004	20
Emilio	72	169	NaN	2004	20
Jocelin	73	170	9.0	2004	20

```
In [41]: # Añadir una nueva columna creando un DataFrame nuevo.
df_mod = df.assign(Hijos = [2, 1, 2, 1])
df_mod
```

```
Out[41]:
```

	Peso	Altura	Mascotas	Año_Nac	Edad	Hijos
Anel	60	145	2.0	2004	20	2
Chucho	74	170	NaN	2004	20	1
Emilio	72	169	NaN	2004	20	2
Jocelin	73	170	9.0	2004	20	1

```
In [42]: df
```

In [42]: df

Out[42]:

	Peso	Altura	Mascotas	Anio_Nac	Edad
Anel	60	145	2.0	2004	20
Chucho	74	170	NaN	2004	20
Emilio	72	169	NaN	2004	20
Jocelin	73	170	9.0	2004	20

In [43]: *# Eliminar una columna existente del DataFrame*
~~del~~ df["Peso"]

In [44]: df

Out[44]:

	Altura	Mascotas	Anio_Nac	Edad
Anel	145	2.0	2004	20
Chucho	170	NaN	2004	20
Emilio	169	NaN	2004	20
Jocelin	170	9.0	2004	20

In [46]: *# Eliminar una columna existente, devlviendo una copia del DataFrame resultante.*
df_mod = df_mod.drop(["Hijos"], axis=1)
df_mod

Out[46]:

	Peso	Altura	Mascotas	Anio_Nac	Edad
Anel	60	145	2.0	2004	20
Chucho	74	170	NaN	2004	20
Emilio	72	169	NaN	2004	20
Jocelin	73	170	9.0	2004	20

In [47]: df

Out[47]:

...