

## Slicing in Both Directions

The screenshot shows a web browser window with the URL `campus.datacamp.com/courses/data-manipulation-with-pandas/slicing-and-indexing-dataframes?ex=8`. The page is titled "Slicing in both directions" and is part of a course on "Data Manipulation with pandas".

**Exercise: Slicing in both directions**

You've seen slicing DataFrames by rows and by columns, but since DataFrames are two-dimensional objects, it is often natural to slice both dimensions at once. That is, by passing two arguments to `.loc[]`, you can subset by rows and columns in one go.

pandas is loaded as `pd`. `temperatures_srt` is indexed by country and city, has a sorted index, and is available.

**Instructions** (100 XP)

- Use `.loc[]` slicing to subset rows from India, Hyderabad to Iraq, Baghdad.
- Use `.loc[]` slicing to subset columns from `date` to `avg_temp_c`.
- Slice in both directions at once from Hyderabad to Baghdad, and `date` to `avg_temp_c`.

[Take Hint \(-30 XP\)](#)

**Code Editor (script.py):**

```
1 # Subset rows from India, Hyderabad to Iraq, Baghdad
2 print(____)
3
4 # Subset columns from date to avg_temp_c
5 print(____)
6
7 # Subset in both directions at once
8 print(____)
```

**Python Shell:**

```
In [1]:
```

You've seen slicing DataFrames by rows and by columns, but since DataFrames are two-dimensional objects, it is often natural to slice both dimensions at once. That is, by passing two arguments to `.loc[]`, you can subset by rows and columns in one go.

pandas is loaded as `pd`. `temperatures_srt` is indexed by country and city, has a sorted index, and is available.

### Final Answer

```
# Subset rows from India, Hyderabad to Iraq, Baghdad
print(temperatures_srt.loc[("India", "Hyderabad"):(("Iraq", "Baghdad"))])
```

```
# Subset columns from date to avg_temp_c
print(temperatures_srt.loc[:, "date":"avg_temp_c"])
```

```
# Subset in both directions at once
print(temperatures_srt.loc[("India", "Hyderabad"):(("Iraq", "Baghdad"),
"date":"avg_temp_c"])
```