Subclass Registration

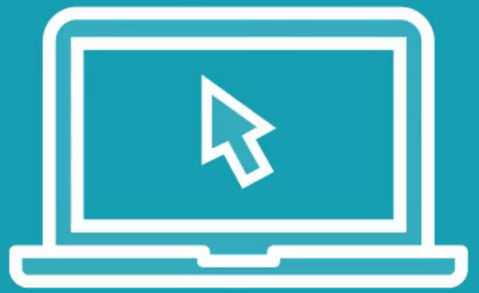
Robert Smallshire
COFOUNDER - SIXTY NORTH

@robsmallshire

Austin Bingham
COFOUNDER - SIXTY NORTH

@austin_bingham

Demo



Situate in historical context

Subclass registration

- Widely accomplished with metaclasses
- Led to new feature `__init_subclass__`

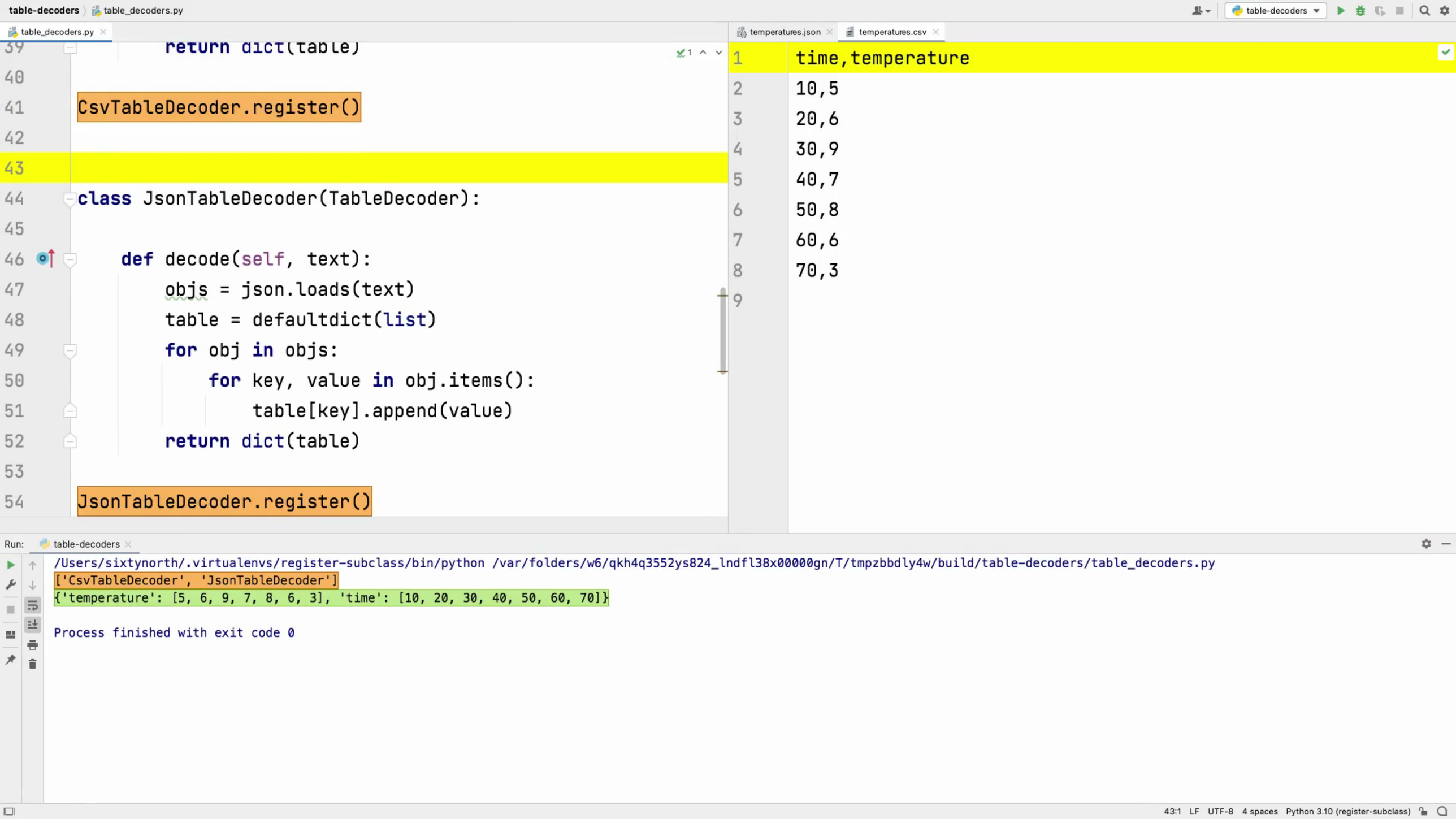


table-decoders
table_decoders.py
table-decoders
table_decoders.py
temperatures.json
temperatures.csv
return dict(table)
CsvTableDecoder.register()
class JsonTableDecoder(TableDecoder):
def decode(self, text):
objs = json.loads(text)
table = defaultdict(list)
for obj in objs:
for key, value in obj.items():
table[key].append(value)
return dict(table)
JsonTableDecoder.register()
time,temperature
10,5
20,6
30,9
40,7
50,8
60,6
70,3
Run:
table-decoders
/Users/sixtynorth/.virtualenvs/register-subclass/bin/python /var/folders/w6/qkh4q3552ys824_lndfl38x00000gn/T/tmpzbbdly4w/build/table-decoders/table_decoders.py
['CsvTableDecoder', 'JsonTableDecoder']
{'temperature': [5, 6, 9, 7, 8, 6, 3], 'time': [10, 20, 30, 40, 50, 60, 70]}
Process finished with exit code 0
43:1
LF
UTF-8
4 spaces
Python 3.10 (register-subclass)

Strong coupling is often a sign of a weak design.

Subclass Registration by Metaclass

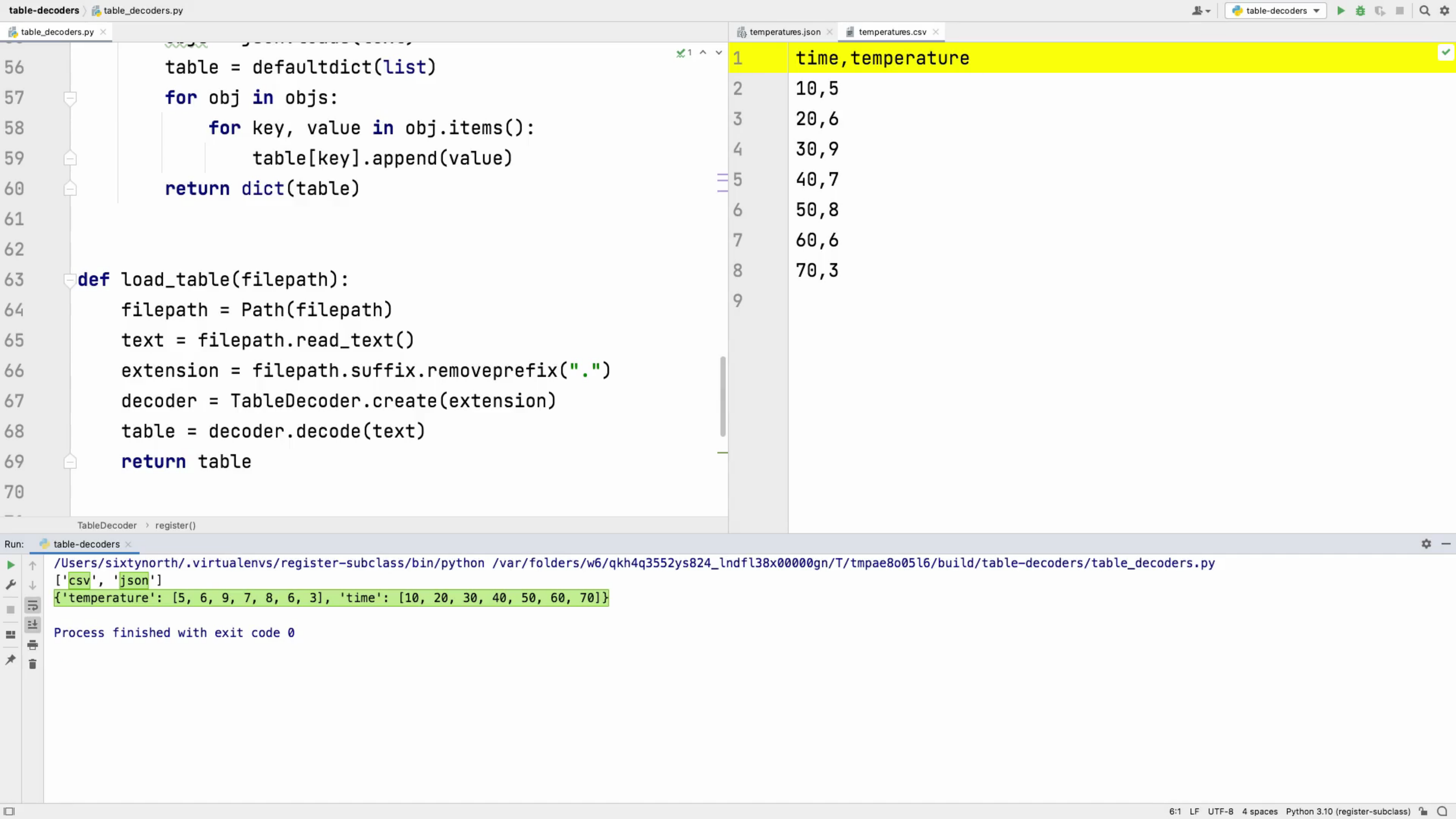
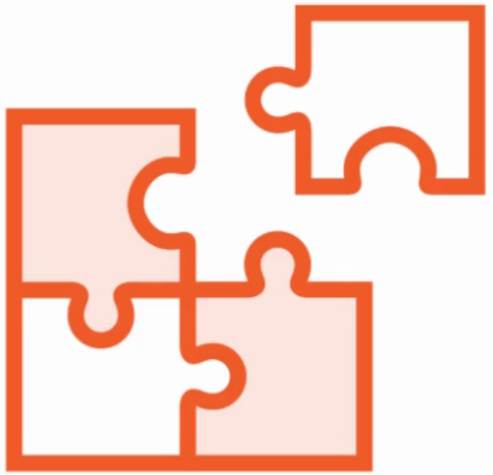


table-decoders
table_decoders.py
table-decoders
table_decoders.py
temperatures.json
temperatures.csv
table = defaultdict(list)
for obj in objs:
for key, value in obj.items():
table[key].append(value)
return dict(table)
def load_table(filepath):
filepath = Path(filepath)
text = filepath.read_text()
extension = filepath.suffix.removeprefix(".")
decoder = TableDecoder.create(extension)
table = decoder.decode(text)
return table
time,temperature
10,5
20,6
30,9
40,7
50,8
60,6
70,3
TableDecoder › register()
Run:
table-decoders
/Users/sixtynorth/.virtualenvs/register-subclass/bin/python /var/folders/w6/qkh4q3552ys824_lndfl38x00000gn/T/tmpae8o05l6/build/table-decoders/table_decoders.py
['csv', 'json']
{'temperature': [5, 6, 9, 7, 8, 6, 3], 'time': [10, 20, 30, 40, 50, 60, 70]}
Process finished with exit code 0
6:1 LF UTF-8 4 spaces Python 3.10 (register-subclass)

Extensibility



Extensible modular system

Add behaviour only by adding code

No modification of existing code

Discover and import modules

Plug-in architecture

Subclass Registration with `__init_subclass__`

`__init_subclass__(cls)`

This method is called whenever the containing class is subclassed. `cls` is then the new subclass.

Python documentation

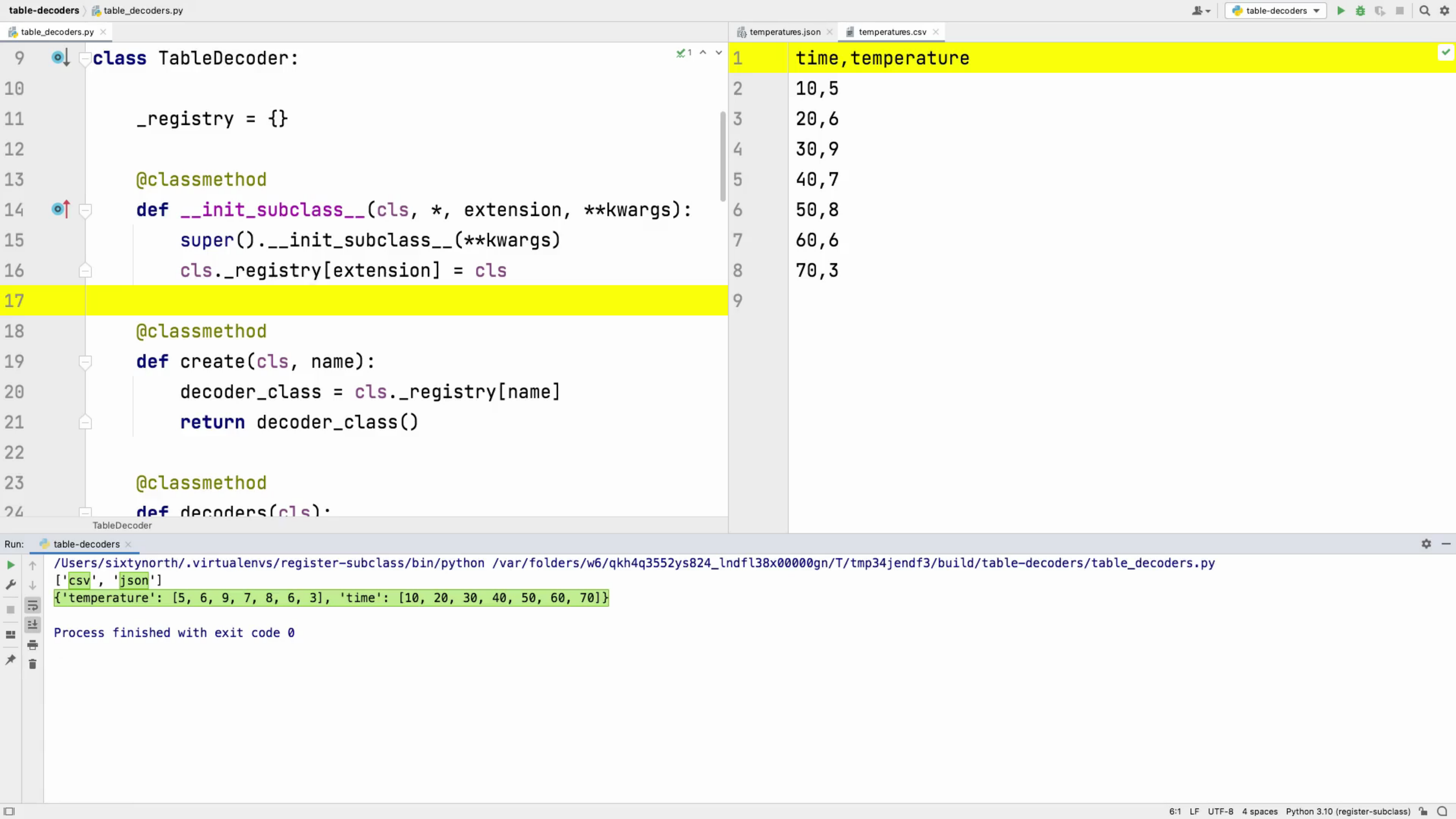


table-decoders
table_decoders.py
table-decoders
table_decoders.py
temperatures.json
temperatures.csv
class TableDecoder:
_registry = {}
@classmethod
def __init_subclass__(cls, *, extension, **kwargs):
super().__init_subclass__(**kwargs)
cls._registry[extension] = cls
@classmethod
def create(cls, name):
decoder_class = cls._registry[name]
return decoder_class()
@classmethod
def decoders(cls):
TableDecoder
time,temperature
10,5
20,6
30,9
40,7
50,8
60,6
70,3
Run:
table-decoders
/Users/sixtynorth/.virtualenvs/register-subclass/bin/python /var/folders/w6/qkh4q3552ys824_lndfl38x00000gn/T/tmp34jendf3/build/table-decoders/table_decoders.py
['csv', 'json']
{'temperature': [5, 6, 9, 7, 8, 6, 3], 'time': [10, 20, 30, 40, 50, 60, 70]}
Process finished with exit code 0
6:1
LF
UTF-8
4 spaces
Python 3.10 (register-subclass)