

Reading Pipe-Delimited Files (PS4.1)

Pipe | delimited files are plain text where fields are separated by the pipe character.

```
state|rate_per_kwh|supply_charge
VIC|0.282|1.20
NSW|0.305|1.15
```

Parsing pattern

```
records = []
with open("rates.txt", "r") as f:
    lines = f.readlines()
    header = lines[0].strip().split("|")

    for line in lines[1:]:
        parts = line.strip().split("|")
        record = {
            "state": parts[0],
            "rate": float(parts[1]),
            "charge": float(parts[2])
        }
        records.append(record)
```

Key steps

1. `strip()` – remove the trailing `\n`
2. `split("|")` – break into a list at each pipe
3. Convert types – `int()`, `float()` for numbers
4. Skip the header line – `lines[1:]`

Any delimiter works: `split("\t")` for tabs, `split(";")` for semicolons.

Writing XML Files (PS4.2)

Build XML strings with f-strings and write to file.

XML structure

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <item id="1">
    <name>SunPower</name>
    <watts>440</watts>
  </item>
</root>
```

Every file needs: **declaration** + **root element** + **child elements**.

Building XML with f-strings

```
xml = '<?xml version="1.0" '
xml += 'encoding="UTF-8"?>\n'
xml += '<panels>\n'

for panel in panels:
    xml += f'  <panel id="{panel["id"]}">\n'
    xml += f'    <brand>{panel["brand"]}'
    xml += f'  </brand>\n'
    xml += f'    <watts>{panel["watts"]}'
    xml += f'  </watts>\n'
    xml += f'  </panel>\n'

xml += '</panels>\n'

with open("export.xml", "w") as f:
    f.write(xml)
```

Data Linking (PS4.3)

Connect records across two lists using a **shared ID**.

Lookup function (linear search)

```
def find_panel_by_id(panels, panel_id):
    for panel in panels:
        if panel["panel_id"] == panel_id:
            return panel
    return None
```

Using the lookup

```
for install in installations:
    panel = find_panel_by_id(
        panels, install["panel_id"])
    if panel:
        print(f"{panel['brand']}: "
              f"{install['quantity']} units")
```

Common Errors

Forgetting `strip()` – leaves `\n` in the last field.

Always `line.strip().split("|")`.

Missing closing tags – every `<tag>` needs `</tag>`.

Check XML output carefully.

None from lookup – handle the case where ID not found.

Always check `if panel:` before using the result.

Quotes in f-strings – XML attributes use `"`.

Use matching quotes: `f'<panel id="{panel["id"]}">'`.