1 A tiny word counter

We will build a minimal word counter and explain its parts. The program runs in $O(n)$ over the bytes of the input. We assemble the final script in the chunk `<<wc.py>>=` by referring to named subchunks that we define later.

⟨*wc.py*⟩≡
  ⟨*imports*⟩
  ⟨*parse-args*⟩
  ⟨*count-words*⟩
  ⟨*main-guard*⟩

1.1 Imports

⟨*imports*⟩≡
```
import sys
from collections import Counter
from pathlib import Path
import argparse
```

1.2 Argument parsing

⟨*parse-args*⟩≡
```
def parse_args(argv=None):
    p = argparse.ArgumentParser(
        description="Count words in a file or stdin."
    )
    p.add_argument("file", nargs="?", help="Path to text file; default: stdin")
    p.add_argument("-n", "--top", type=int, default=10,
                   help="Show top-N words (default: 10)")
    return p.parse_args(argv)
```

1.3 Counting logic

⟨*count-words*⟩≡
```
  def words_from_text(text: str):
      # Very naive tokenization: split on whitespace and punctuation
      # Lowercasing makes counts case-insensitive.
      import re
      for w in re.findall(r"[A-Za-z0-9']+", text.lower()):
          yield w

  def count_words(stream) -> Counter:
      c = Counter()
      for line in stream:
          c.update(words_from_text(line))
      return c
```

1.4 Program entry point

⟨*main-guard*⟩≡
```
  def main(argv=None):
      args = parse_args(argv)
      if args.file:
          data = Path(args.file).read_text(encoding="utf-8", errors="ignore")
          stream = data.splitlines(keepends=True)
      else:
          stream = sys.stdin

      counts = count_words(stream)
      for word, n in counts.most_common(args.top):
          print(f"{n:7d}  {word}")

  if __name__ == "__main__":
      main()
```