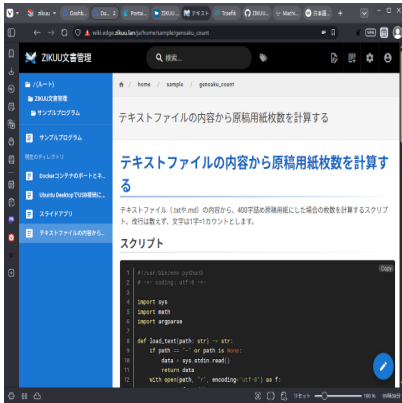




ã??ã?ã?¹ã??ã??ã?|ã?ªã?«ã?®å? å®¹ã?•ã??ã??ç?¿ç?´ç?æ??æ?°ã??è?´ç?®?ã?
?ã??ã?¹ã?¬ã?ªã??ã??

Description

ZIKUUã•®æ??æ?,ç®¡ç?·ã?·ã?¹ã??ã? ã•«ã?·ã?·ã?µã?³ã??ã?«ã??ã?ã?°ã?©ã? ã•®à,?ã•µã•§ã•
 ?ã??ã?¹½ã•«ç«?ã•¡ã•«ã?·ã?®ã•§ã?·ã -é??ã•è?¹½ã•ã??ã•®ã• -ã -ã«ã?ã•¡ã?ã?ã?ã?·ã?·æ?·ã•£ã•¡ã•
 ?ã•¾ã?·ã??

ä•?ä•®ä?¹ä?¬ä?ªä??ä??ä ä?ä??ä?ä¹ä??ä??ä??ä?jä?ªä?«ä??èªä??ä•§ä?•ä??ç¨ç?¨ç?¨ç?æ•?ç®?ä•?ä•?ä•
 ¨ä•ä•®æ??æ?°ä??è¨ç®?ä•?ä??ä??ä•®ä•§ä?ä??ä??ä•ä•£ä?ä??ä¹½ä•£ä•jä•ä•ä•ä•ä•ä??

ã?¹ã?⁻ã?ªã??ã??

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
```

```
import sys
import math
import argparse
```

```
def load_text(path: str) -> str:
    if path == "-" or path is None:
        data = sys.stdin.read()
        return data
    with open(path, "r", encoding="utf-8") as f:
        return f.read()
```

```
def count_chars(text: str, exclude_spaces: bool) -> int:
    # æ?¹è¡?ã·-ã?«ã?|ã?³ã?ã·?ã·ªã·?ï¼?å??ç¨¿ç?¨ç´?ã·®ã?ã?¹ç?®ã·«å?¥ã?ã·ªã·
    ·?ã·?ã·?·ï¼?
    # å??ã??ã?©ã?«ã??ã·§ã·-ç©°ç?½ï¼?å·?è§/?/å?"è§?ï¼?ã?»å·¥èªç?¹ã?»è¨?å·
    ·ã?»æ?°å?ã?»è±å?ã??1å?ã·¨ã·?ã·|æ?°ã·?ã??
    if exclude_spaces:
        # å·?è§?ã?¹å??ã?¾ã?¹ã?»å?"è§?ã?¹ã??ã?¾ã?¹ã?»ã?¿ã??ã??é?¤å¤?
        filtered = text.replace("\n", "").replace("\r", "")
        filtered = filtered.replace(" ", "").replace("\u3000", "").replace("\t
```

```
        return len(filtered)
    else:
        filtered = text.replace("\n", "").replace("\r", "")
        return len(filtered)

def main():
    parser = argparse.ArgumentParser(
        description="ã??ç??ç??ç??æ•
?ç®?ã??ã?¼ã?«i¼?400å?=1æ??æ?çå®?i¼?ã??æ?¹èj?ã•¯æ?°ã•?ã•?ã?•æ??å?ã•
¹å?=1ã?«ã?|ã?³ã??ã??"
    )
    parser.add_argument("path", help="ã??ã?ã?¹ã??ã??ã?jã?ã?«ã•
®ã??ã?¹ã??æ"?æ?°å?¥å??ã??ã?¼ã?ã?å 'å•?ã•¯ '-' ã??æ??å®?ã??")
    parser.add_argument("--chars-per-page", type=int, default=400, help="1æ??ã
•?ã•?ã??ã•®æ??å?æ?°i¼?æ?çå®?: 400i¼?")
    parser.add_argument("--exclude-spaces", action="store_true",
        help="ç©°ç?½i¼?å•?è§?/å?"è§?i¼?ã•¯ã?¼ã??ã??æ??å?æ?°ã
•?ã??é?ã?ã?ã?ã?ã?")
    args = parser.parse_args()

    text = load_text(args.path)
    lines = text.splitlines()
    total_lines = len(lines)

    total_chars = count_chars(text, exclude_spaces=args.exclude_spaces)
    pages = math.ceil(total_chars / max(1, args.chars_per_page)) # 0é?ç®?ã¼
•è.

    # çµ•æ??èj"çµ°
    print("=== å??ç??ç??ç??æ•?ç®? çµ•æ?? ===")
    print(f"ã??ã?jã?ã?« : {args.path}")
    print(f"èj?æ?°i¼?ç©°ç?½èj?å•?«ã??i¼? : {total_lines}")
    print(f"ç••æ??å?æ?° : {total_chars} æ??å?"
        + ("i¼?ç©°ç?½é?ã?ã?i¼?" if args.exclude_spaces else "i¼?ç©°ç?½å•
«ã??i¼?"))
    print(f"æ•?ç®?è"å®? : 1æ??ã•?ã•?ã?? {args.chars_per_page} æ??å?")
    print(f"å??ç??ç??ç??æ•?ç®? : ç'? {pages} æ??")

if __name__ == "__main__":
    main()

ã?¹ã? ¯ã?ªã??ã??ã?®ã½¼ã?æ?¹

# ã?³ã??ã?³ã??æ§?æ??
gensaku_count.py [-h] [--chars-per-page CHARS_PER_PAGE]
                [--exclude-spaces]
                path
# ç°jå•?ç??i¼?ã??ã??ã?®ã?«ã?ã?ã•¯400å?è©°ã?•å??ç??ç??ç??i¼?
python gensaku_count.py ã??ã?jã?ã?«å••

# ä_?æ??ã•®æ??å?æ?°ã??æ??å®?ã?ã?ã?
python gensaku_count.py --chars-per-page æ??å?æ?° ã??ã?jã?ã?«å••

# å••ã•®ã?»ã?•ã??ã?«ã??å?ç?§ã??
```


1. æ?¥ã? ã•®æ´»å??

Tags

- 1. Python
- 2. ä,?é??å??å,?
- 3. ä°?å??æµ•
- 4. å±±±æ€¨ç??
- 5. é?±æ?«ç§§?å!è?

Date Created
2025å¹10æ??27æ?¥

Author
kazuo-tsubaki

default watermark