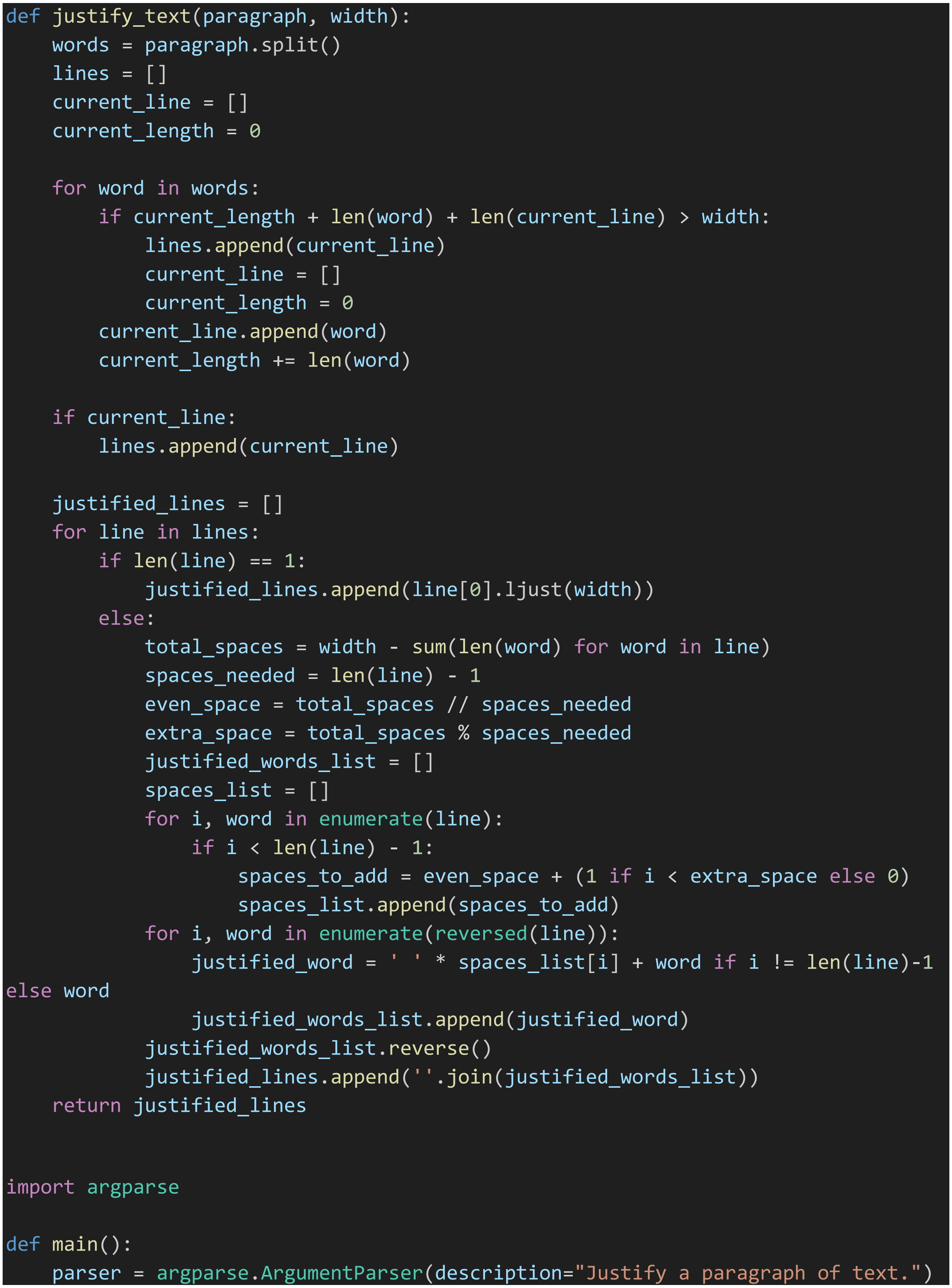

```
def justify_text(paragraph, width):
    words = paragraph.split()
    lines = []
    current_line = []
    current_length = 0

    for word in words:
        if current_length + len(word) + len(current_line) > width:
            lines.append(current_line)
            current_line = []
            current_length = 0
        current_line.append(word)
        current_length += len(word)

    if current_line:
        lines.append(current_line)

    justified_lines = []
    for line in lines:
        if len(line) == 1:
            justified_lines.append(line[0].ljust(width))
        else:
            total_spaces = width - sum(len(word) for word in line)
            spaces_needed = len(line) - 1
            even_space = total_spaces // spaces_needed
            extra_space = total_spaces % spaces_needed
            justified_words_list = []
            spaces_list = []
            for i, word in enumerate(line):
                if i < len(line) - 1:
                    spaces_to_add = even_space + (1 if i < extra_space else 0)
                    spaces_list.append(spaces_to_add)
            for i, word in enumerate(reversed(line)):
                justified_word = ' ' * spaces_list[i] + word if i != len(line)-1
else word
                justified_words_list.append(justified_word)
            justified_words_list.reverse()
            justified_lines.append(''.join(justified_words_list))
    return justified_lines


import argparse

def main():
    parser = argparse.ArgumentParser(description="Justify a paragraph of text.")
```

```python
    parser.add_argument('--paragraph-string', type=str, required=True, help="The paragraph string to justify.")
    parser.add_argument('--paragraph-width', type=int, required=True, help="The width to justify the paragraph to.")
    args = parser.parse_args()

    paragraph = args.paragraph_string
    width = args.paragraph_width

    justified_lines = justify_text(paragraph, width)

    for i, line in enumerate(justified_lines):
        print(f"Array [{i+1}] = \"{line}\"")

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