

■ Python f-String Cheat Sheet

Python f-strings (formatted string literals) are a powerful and concise way to embed expressions inside string literals using curly braces `{}`. Introduced in Python 3.6, f-strings improve readability and performance.

Topic	Example
Basic f-string	<pre>name = "Alice" msg = f"Hello, {name}!"</pre>
Multiline f-string	<pre>f""" Line 1: {line1} Line 2: {line2} """</pre>
Add newline	<pre>f"Line 1\nLine 2"</pre>
Float precision	<pre>f"{3.14159:.2f}" # 3.14</pre>
Integer formatting	<pre>f"{42:04}" # 0042 (leading zeros)</pre>
Hexadecimal	<pre>f"{255:#x}" # 0xff</pre>
Binary format	<pre>f"{8:b}" # 1000</pre>
Commas in numbers	<pre>f"{1000000:,.}" # 1,000,000</pre>
Casting	<pre>f"{int(float_val)}" # float to int</pre>
String with quotes	<pre>f"\"She said, 'Hello!'\""</pre>
String formatting	<pre>f"{value:<10}" # left-align with padding</pre>
Read file	<pre>with open("file.txt") as f: data = f.read() print(f"Data: {data}")</pre>
Write to console	<pre>print(f"Debug: {var=}")</pre>
Size of object	<pre>import sys sys.getsizeof(obj)</pre>
Edit text file	<pre>open("file.txt", "w").write("Updated")</pre>
Range float	<pre>import numpy as np np.arange(0, 1, 0.1)</pre>
Extract number from string	<pre>import re re.findall(r"\d+", "abc123")</pre>
Function in f-string	<pre>f"Result: {my_func(5)}"</pre>
Call function	<pre>def greet(): return "Hi" print(f"{greet()}")</pre>
Return statement	<pre>def add(x, y): return x + y</pre>