Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

[1] 1s

```
import time
import tracemalloc
import os
import psutil
import py_compile

def is_palindrome(num):
    original = str(num)
    reversed_num = original[::-1]
    return original == reversed_num

# Start memory tracking
tracemalloc.start()

# Execution time
start_time = time.perf_counter()
number = 12321
palindrome_result = is_palindrome(number)
end_time = time.perf_counter()
execution_time = end_time - start_time

# Memory usage
current, peak = tracemalloc.get_traced_memory()
tracemalloc.stop()

# Save code to file to measure size & compile
code_filename = "temp_code.py"
with open(code_filename, "w") as f:
    f.write("def is_palindrome(num):\n"
```

Empty cell

What can I help you build?

Variables Terminal

8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN 09:15 09-08-2025

Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

[1]

```
# Save code to file to measure size & compile
code_filename = "temp_code.py"
with open(code_filename, "w") as f:
    f.write("def is_palindrome(num):\n"
            "    return str(num) == str(num)[::-1]\n"
            f"print(is_palindrome({number}))\n")

file_size = os.path.getsize(code_filename)

compile_start = time.perf_counter()
py_compile.compile(code_filename, cfile="temp_compiled.pyc")
compile_end = time.perf_counter()
compilation_time = compile_end - compile_start

# CPU usage
process = psutil.Process(os.getpid())
cpu_usage_percent = process.cpu_percent(interval=1)

# Output
if palindrome_result:
    print(f"{number} is a palindrome.")
else:
    print(f"{number} is not a palindrome.")

print(f"Execution Time: {execution_time:.6f} seconds")
print(f"Current Memory Usage: {current / 1024:.2f}
print(f"Peak Memory Usage: {peak / 1024:.2f} KB")
print(f"Code Size: {file_size / 1024:.2f} KB")
print(f"Compilation Time: {compilation_time:.6f} s
print(f"CPU Usage: {cpu_usage_percent}%")
```

Empty cell

What can I help you build?

Variables Terminal

8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN

09:16 09-08-2025

Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

```
[1] compile_start = time.perf_counter()
    py_compile.compile(code_filename, cfile="temp_compiled.pyc")
    compile_end = time.perf_counter()
    compilation_time = compile_end - compile_start

    # CPU usage
    process = psutil.Process(os.getpid())
    cpu_usage_percent = process.cpu_percent(interval=1)

    # Output
    if palindrome_result:
        print(f"{number} is a palindrome.")
    else:
        print(f"{number} is not a palindrome.")

    print(f"Execution Time: {execution_time:.6f} seconds")
    print(f"Current Memory Usage: {current / 1024:.2f} KB")
    print(f"Peak Memory Usage: {peak / 1024:.2f} KB")
    print(f"Code Size: {file_size / 1024:.2f} KB")
    print(f"Compilation Time: {compilation_time:.6f} seconds")
    print(f"CPU Usage: {cpu_usage_percent}%")
```

```
12321 is a palindrome.
Execution Time: 0.000629 seconds
Current Memory Usage: 0.97 KB
Peak Memory Usage: 11.67 KB
Code Size: 0.09 KB
Compilation Time: 0.000583 seconds
CPU Usage: 0.0%
```

Empty cell

What can I help you build?

Variables Terminal

8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN 09:16 09-08-2025

Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

[2] 2s

```
import time  # For execution and compilation time
import tracemalloc  # For memory usage
import os
import psutil  # For CPU usage
import py_compile  # For compilation time
import math  # For factorial in Maclaurin series
import tempfile  # For creating temp files in Colab

# Function to calculate Maclaurin series for e^x
def maclaurin_series_e_x(x, terms):
    result = 0
    for n in range(terms):
        result += (x ** n) / math.factorial(n)  # Series term
    return result

# ---------------- Performance Measurement ---------------- #

# Measure execution time
start_time = time.perf_counter()
value = maclaurin_series_e_x(2, 10)  # Example: e^2 using 10 terms
end_time = time.perf_counter()
execution_time = end_time - start_time

# Measure memory usage
tracemalloc.start()
maclaurin_series_e_x(2, 10)  # Run function again for memory tracking
current, peak = tracemalloc.get_traced_memory()
tracemalloc.stop()

# Measure code size (simulate by writing this code
code_string = """
```

Empty cell

What can I help you build?

Variables Terminal

8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN 09:16 09-08-2025

Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

[2]
```
# Measure code size (simulate by writing this code to a temp file)
code_string = """
import math
def maclaurin_series_e_x(x, terms):
    result = 0
    for n in range(terms):
        result += (x ** n) / math.factorial(n)
    return result
"""
temp_code_path = tempfile.mktemp(suffix=".py")
with open(temp_code_path, "w") as f:
    f.write(code_string)
file_size = os.path.getsize(temp_code_path)

# Measure compilation time
compile_start = time.perf_counter()
py_compile.compile(temp_code_path, cfile=temp_code_path + "c")
compile_end = time.perf_counter()
compilation_time = compile_end - compile_start

# Measure CPU usage percentage
process = psutil.Process(os.getpid())
cpu_usage_percent = process.cpu_percent(interval=1)

# ---------------- Output Results ---------------- #
print(f"Maclaurin Series Result for e^2 (10 terms): {value}")
print(f"Execution Time: {execution_time:.6f} secon
print(f"Current Memory Usage: {current / 1024:.2f}
print(f"Peak Memory Usage: {peak / 1024:.2f} KB")
print(f"Code Size: {file_size / 1024:.2f} KB")
print(f"Compilation Time: {compilation_time:.6f} seconds")
```

Empty cell

What can I help you build?

Variables Terminal

8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN

09:16 09-08-2025

Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

```
with open(temp_code_path, "w") as f:
    f.write(code_string)
file_size = os.path.getsize(temp_code_path)

# Measure compilation time
compile_start = time.perf_counter()
py_compile.compile(temp_code_path, cfile=temp_code_path + "c")
compile_end = time.perf_counter()
compilation_time = compile_end - compile_start

# Measure CPU usage percentage
process = psutil.Process(os.getpid())
cpu_usage_percent = process.cpu_percent(interval=1)

# ---------------- Output Results ---------------- #
print(f"Maclaurin Series Result for e^2 (10 terms): {value}")
print(f"Execution Time: {execution_time:.6f} seconds")
print(f"Current Memory Usage: {current / 1024:.2f} KB")
print(f"Peak Memory Usage: {peak / 1024:.2f} KB")
print(f"Code Size: {file_size / 1024:.2f} KB")
print(f"Compilation Time: {compilation_time:.6f} seconds")
print(f"CPU Usage: {cpu_usage_percent}%")
```

```
Maclaurin Series Result for e^2 (10 terms): 7.3887125220458545
Execution Time: 0.000108 seconds
Current Memory Usage: 1.04 KB
Peak Memory Usage: 11.77 KB
Code Size: 0.15 KB
Compilation Time: 0.001570 seconds
CPU Usage: 0.0%
```

Empty cell

What can I help you build?

Variables Terminal 8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN 09:17 09-08-2025

Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

[3] 2s

```
import time  # For execution and compilation time
import tracemalloc  # For memory usage
import os
import psutil  # For CPU usage
import py_compile  # For compilation time
import tempfile  # For creating temp files in Colab
import sys  # For recursion limit adjustment

# ---------------- Deep Recursion Function ---------------- #
# Recursive Fibonacci
def fibonacci_recursive(n):
    if n <= 1:
        return n
    return fibonacci_recursive(n-1) + fibonacci_recursive(n-2)

# Increase recursion limit (be careful with very large values)
sys.setrecursionlimit(3000)

# ---------------- Performance Measurement ---------------- #

# Measure execution time
start_time = time.perf_counter()
value = fibonacci_recursive(30)  # Example: Fibonacci(30)
end_time = time.perf_counter()
execution_time = end_time - start_time

# Measure memory usage
tracemalloc.start()
fibonacci_recursive(30)  # Run function again for
current, peak = tracemalloc.get_traced_memory()
tracemalloc.stop()
```

Empty cell

What can I help you build?

Variables Terminal

8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN 09:17 09-08-2025

Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

[3]
```
# Measure memory usage
tracemalloc.start()
fibonacci_recursive(30)  # Run function again for memory tracking
current, peak = tracemalloc.get_traced_memory()
tracemalloc.stop()

# Measure code size (simulate by writing code to a temp file)
code_string = """
def fibonacci_recursive(n):
    if n <= 1:
        return n
    return fibonacci_recursive(n-1) + fibonacci_recursive(n-2)
"""
temp_code_path = tempfile.mktemp(suffix=".py")
with open(temp_code_path, "w") as f:
    f.write(code_string)
file_size = os.path.getsize(temp_code_path)

# Measure compilation time
compile_start = time.perf_counter()
py_compile.compile(temp_code_path, cfile=temp_code_path + "c")
compile_end = time.perf_counter()
compilation_time = compile_end - compile_start

# Measure CPU usage percentage
process = psutil.Process(os.getpid())
cpu_usage_percent = process.cpu_percent(interval=1

# ---------------- Output Results ----------------
print(f"Fibonacci(30) Result: {value}")
print(f"Execution Time: {execution_time:.6f} seconds")
```

Empty cell

What can I help you build?

Variables Terminal

8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN 09:17 09-08-2025

Welcome to Col | Untitled11.ipynb | Introducing Cha | Code logic corre | Code correction | WhatsApp | Sec | (110) WhatsApp | Python codes wi

colab.research.google.com/drive/1qrBKHH2qN5GnARSYHXkE6uqh-hefHL1k#scrollTo=_MfjNMnt2Sdn

Untitled11.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

[3]

```
# Measure CPU usage percentage
process = psutil.Process(os.getpid())
cpu_usage_percent = process.cpu_percent(interval=1)

# ---------------- Output Results ---------------- #
print(f"Fibonacci(30) Result: {value}")
print(f"Execution Time: {execution_time:.6f} seconds")
print(f"Current Memory Usage: {current / 1024:.2f} KB")
print(f"Peak Memory Usage: {peak / 1024:.2f} KB")
print(f"Code Size: {file_size / 1024:.2f} KB")
print(f"Compilation Time: {compilation_time:.6f} seconds")
print(f"CPU Usage: {cpu_usage_percent}%")
```

```
Fibonacci(30) Result: 832040
Execution Time: 0.606381 seconds
Current Memory Usage: 0.97 KB
Peak Memory Usage: 11.70 KB
Code Size: 0.12 KB
Compilation Time: 0.000688 seconds
CPU Usage: 1.0%
```

Start coding or generate with AI.

Empty cell

What can I help you build?

Variables Terminal

8:51 AM Python 3

Light rain Tomorrow

Search

ENG IN 09:17 09-08-2025