```python
# Handling large numbers

import math
'''We can use the built-in math module, which is optimized and handles large integers natively.'''
# Calculate 100 factorial
fact_100 = math.factorial(100)

# Print the result
print("100! =", fact_100)
```

> 100! = 93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253692082722375825118!

```python
def factorial(n):
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result

# Calculate and print 100!
fact_100 = factorial(100)
print("100! =", fact_100)
```

> 100! = 93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253692082722375825118!

```python
'''Both the above approaches work fine in Python because Python's integers have arbitrary precision,
meaning they can grow as large as the memory allows.'''

#When we do the below
s = "string"
s = s + "name"
'''A new string is created.
The original memory address of s changes.
This is seen using id(s) (i.e., the memory address).
This leads to unnecessary memory allocation, especially in loops (inefficient).'''
```

> 'A new string is created.\nThe original memory address of s changes.\nThis is seen using id(s) (i.e., the memory address).\nThis leads to unnecessary memory allocation, especially in loops (inefficient).'

```python
s = "hello"
print(id(s))
s = s + " world"
print(id(s))  # different id → new object created
```

> 1827731820144
> 1827735522544

```python
result = ""
for word in ["a", "b", "c"]:
    result += word  # New string created every time
    print(id(result))
```

> 140729977634016
> 1827735344960
> 1827735350864

```python
#Using List Instead
lst = []
for i in range(5):
    lst.append(str(i))  # Efficient
s = ''.join(lst)  # Only one string created here

ch = chr(65) #To print a character using it's encode
print(ch)
print(ord('R')) #To print encoding of a character
```

⤓ A
    82


```
import unicodedata
#To print a character that is not in the keyboard
x = unicodedata.lookup("GREEK SMALL LETTER OMEGA")
print(x)

print("\U0001F60A") #This Unicode code points above U+FFFF require 8-digit hex
```

⤓ ω
    😊


```
import sys  # Imports the sys module to access the Python interpreter path (since the interpreter is in C
!{sys.executable} -m pip install nltk
```

⤓ Requirement already satisfied: nltk in c:\users\rufin\appdata\local\programs\python\python312\lib\site-packages (3.9.1)
    Requirement already satisfied: click in c:\users\rufin\appdata\local\programs\python\python312\lib\site-packages (from nltk) (8.1.7
    Requirement already satisfied: joblib in c:\users\rufin\appdata\local\programs\python\python312\lib\site-packages (from nltk) (1.4.2
    Requirement already satisfied: regex>=2021.8.3 in c:\users\rufin\appdata\local\programs\python\python312\lib\site-packages (from nlt
    Requirement already satisfied: tqdm in c:\users\rufin\appdata\local\programs\python\python312\lib\site-packages (from nltk) (4.67.1
    Requirement already satisfied: colorama in c:\users\rufin\appdata\local\programs\python\python312\lib\site-packages (from click->nlt

    [notice] A new release of pip is available: 25.0.1 -> 25.1.1
    [notice] To update, run: C:\Users\rufin\AppData\Local\Programs\Python\Python312\python.exe -m pip install --upgrade pip


```
import nltk

nltk.download('all')
```

⤓ [nltk_data] Downloading collection 'all'
    [nltk_data] |
    [nltk_data] | Downloading package abc to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package abc is already up-to-date!
    [nltk_data] | Downloading package alpino to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package alpino is already up-to-date!
    [nltk_data] | Downloading package averaged_perceptron_tagger to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package averaged_perceptron_tagger is already up-
    [nltk_data] | to-date!
    [nltk_data] | Downloading package averaged_perceptron_tagger_eng to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package averaged_perceptron_tagger_eng is already
    [nltk_data] | up-to-date!
    [nltk_data] | Downloading package averaged_perceptron_tagger_ru to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package averaged_perceptron_tagger_ru is already
    [nltk_data] | up-to-date!
    [nltk_data] | Downloading package averaged_perceptron_tagger_rus to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package averaged_perceptron_tagger_rus is already
    [nltk_data] | up-to-date!
    [nltk_data] | Downloading package basque_grammars to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package basque_grammars is already up-to-date!
    [nltk_data] | Downloading package bcp47 to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package bcp47 is already up-to-date!
    [nltk_data] | Downloading package biocreative_ppi to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package biocreative_ppi is already up-to-date!
    [nltk_data] | Downloading package bllip_wsj_no_aux to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package bllip_wsj_no_aux is already up-to-date!
    [nltk_data] | Downloading package book_grammars to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package book_grammars is already up-to-date!
    [nltk_data] | Downloading package brown to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package brown is already up-to-date!
    [nltk_data] | Downloading package brown_tei to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package brown_tei is already up-to-date!
    [nltk_data] | Downloading package cess_cat to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package cess_cat is already up-to-date!
    [nltk_data] | Downloading package cess_esp to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
    [nltk_data] | Package cess_esp is already up-to-date!
    [nltk_data] | Downloading package chat80 to
    [nltk_data] | C:\Users\rufin\AppData\Roaming\nltk_data...
```

```
[nltk_data]    |    Package chat80 is already up-to-date!
[nltk_data]    | Downloading package city_database to
[nltk_data]    |      C:\Users\rufin\AppData\Roaming\nltk_data...
[nltk_data]    |    Package city_database is already up-to-date!
[nltk_data]    | Downloading package cmudict to
```

```python
from nltk.tokenize import word_tokenize, sent_tokenize
from urllib import request

url = "https://www.gutenberg.org/cache/epub/76503/pg76503-images.html"

response = request.urlopen(url)
raw = response.read().decode('utf8')
```

```python
import sys
!{sys.executable} -m pip install html2text
```

Requirement already satisfied: html2text in c:\users\rufin\appdata\local\programs\python\python312\lib\site-packages (2025.4.15)

```
[notice] A new release of pip is available: 25.0.1 -> 25.1.1
[notice] To update, run: C:\Users\rufin\AppData\Local\Programs\Python\Python312\python.exe -m pip install --upgrade pip
```

```python
import html2text

#Converting html to text
text = html2text.html2text(raw)

print("Text: " , text)
```

Text: ## The Project Gutenberg eBook of The man who mastered time

This ebook is for the use of anyone anywhere in the United States and most
other parts of the world at no cost and with almost no restrictions
whatsoever. You may copy it, give it away or re-use it under the terms of the
Project Gutenberg License included with this ebook or online at
www.gutenberg.org. If you are not located in the
United States, you will have to check the laws of the country where you are
located before using this eBook.

Title : The man who mastered time

Author : Ray Cummings

Illustrator : Ed Valigursky

Release date : July 14, 2025 [eBook #76503]

Language : English

Original publication : New York, NY: Ace Books, 1929

Credits : Greg Weeks, Paul Ereaut, Mary Meehan & the Online Distributed
Proofreading Canada Team at http://www.pgdpcanada.net

*** START OF THE PROJECT GUTENBERG EBOOK THE MAN WHO MASTERED TIME ***

THE MAN WHO MASTERED TIME

RAY CUMMINGS

ACE BOOKS

A Division of A. A. Wyn, Inc.
23 West 47th Street, New York 36, N. Y.

THE MAN WHO MASTERED TIME

Copyright, 1929, by Ray Cummings

An Ace Book, by arrangement with the author.

To Gabrielle
Who has given me affectionate

```
    assistance for a long, long time.
```

```python
from nltk.tokenize import sent_tokenize, word_tokenize

# Sentence tokenization
sentences = sent_tokenize(text)

# Word tokenization for each sentence
all_tokens = []
for sentence in sentences:
    tokens = word_tokenize(sentence)
    all_tokens.extend(tokens)  # flatten into one list, or store separately if needed

# Print results
print("Number of sentences:", len(sentences))
print("Number of tokens:", len(all_tokens))
print("Tokens:", all_tokens)
```

```
Number of sentences: 4767
Number of tokens: 70689
Tokens: ['#', '#', 'The', 'Project', 'Gutenberg', 'eBook', 'of', 'The', 'man', 'who', 'mastered', 'time', 'This', 'ebook', 'is', 'fo
```

```python
from nltk import pos_tag
# Part-of-speech tagging
tagged = pos_tag(all_tokens)
print("POS Tagged:", tagged)
```

```
POS Tagged: [('#', '#'), ('#', '#'), ('The', 'DT'), ('Project', 'NNP'), ('Gutenberg', 'NNP'), ('eBook', 'NN'), ('of', 'IN'), ('The',
```

Start coding or generate with AI.