

Donate



Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!
# Input, assignment
>>> name = input('What is your name? ')
What is your name?
Python
>>> print(f'Hi, {name}.')
Hi, Python.
```



Quick & Easy to Learn

Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.

1

2

3

4

5

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

Join us in Pittsburgh, PA starting May 14, 2025. Grab your ticket today before we sell out!

REGISTER FOR PYCON US!

Usecase

Att skapa ett programmering-stöd som kan fungera lokalt utan koppling till “betaltjänster” och som hjälper användare att svara på frågor om Pythondokumentationen.

- Gör det lättare att få svar än att leta runt i dokumentationen
- Svarar utifrån den senaste releasen och kompletterar en LLM som har ett brytdatum eller forum där det blandas gammal och ny kod.

Alla bibliotek kan köras lokalt. Har fuskat lite med lokal LLM som körs via OpenRouter, då min dator inte klarar av att ladda ner och köra så stora modeller.

Om python-dokumentationen

- txt-filer
- 15 foldrar
- 507 filer
- längsta fil: 1 873 742 char
- kortaste fil: 133 char

```
=== Python Documentation Analysis ===  
Total number of documents: 507
```

Top 10 longest documents:

```
-----  
1,873,742 characters: whatsnew/changelog.txt  
199,849 characters: library/stdtypes.txt  
184,940 characters: library/os.txt  
157,337 characters: about_the_documentation/contents.txt  
156,253 characters: howto/logging-cookbook.txt  
144,328 characters: reference/datamodel.txt  
126,495 characters: library/typing.txt  
123,376 characters: whatsnew/2.6.txt  
115,790 characters: library/multiprocessing.txt  
113,462 characters: whatsnew/3.13.txt
```

Top 10 shortest documents:

```
-----  
389 characters: library/xdrLib.txt  
389 characters: library/uu.txt  
388 characters: library/sunau.txt  
382 characters: library/mailcap.txt  
380 characters: library/nntplib.txt  
376 characters: library/chunk.txt  
295 characters: faq/index.txt  
237 characters: distributing/index.txt  
156 characters: library/concurrent.txt  
133 characters: howto/clinic.txt  
(venv) █
```

python-3.13-docs-text

- > about_the_documentation
- > c-api
- > deprecations
- > distributing
- > extending
- > faq
- > howto
- > installing
- > library
- > reference
- > tutorial
- > using
- > whatsnew

```
count_char.py  
count_libraries.py  
doc_str_analys.py
```

Innehåll i dokumenten

listor med
funktioner,
och
ändringar

Varierande
formatering
på rubriker

förklarande
text,
tutorials

```
As of Python 3.3, this is equivalent to "os.pathconf(fd, name)".
Availability: Unix.

os.fstat(fd)

Get the status of the file descriptor *fd*. Return a
object.

As of Python 3.3, this is equivalent to "os.stat(fd)".
See also: The "stat()" function.

os.fstatvfs(fd, /)

Return information about the filesystem containing the
associated with file descriptor *fd*, like "statvfs()"
Python 3.3, this is equivalent to "os.statvfs(fd)".
Availability: Unix.

os.fsync(fd)

Force write of file with file descriptor *fd* to disk.
this calls the native "fsync()" function; on Windows,
"_commit()" function.

If you're starting with a buffered Python *file object
do "f.flush()", and then do "os.fsync(f.fileno())", to
```

```
changelog

*Release date: XXXX-XX-XX*

Tests

* gh-109981: The test helper that counts the list of open file
descriptors now uses the optimised "/dev/fd" approach on all Apple
platforms, not just macOS. This avoids crashes caused by guarded
file descriptors.

Library

* gh-132673: Fix a crash when using "_align_ = 0" and "_fields_ = []"
in a "ctypes.Structure".
* gh-132527: Include the valid typecode 'w' in the error message when
an invalid typecode is passed to "array.array".
* gh-132429: Fix support of Bluetooth sockets on NetBSD and DragonFly
BSD.
* gh-132106: "QueueListener.start" now raises a "RuntimeError" if the
listener is already started.
* gh-132417: Fix a "NULL" pointer dereference when a C function called
using "ctypes" with "restype" "py object" returns "NULL".
```

```
What's New In Python 3.13
*****
```

Editors:
Adam Turner and Thomas Wouters

This article explains the new features in Python 3.13, compared to
3.12. Python 3.13 was released on October 7, 2024. For full details,
see the [changelog](#).

See also: [**PEP 719**](#) -- Python 3.13 Release Schedule

Summary -- Release Highlights

Python 3.13 is the latest stable release, with a mix of changes to
the standard library. The biggest changes are the interactive interpreter, experiment
mode (**PEP 703**), and a

```
errors.txt

B. Errors and Exceptions
*****

Until now error messages haven't been more than mentioned, but if you
have tried out the examples you have probably seen some. There are
(at least) two distinguishable kinds of errors: *syntax errors* and
*exceptions*.

8.1. Syntax Errors
=====

Syntax errors, also known as parsing errors, are perhaps the most
common kind of complaint you get while you are still learning Python:

>>> while True print('Hello world')
File "<stdin>", line 1
  while True print('Hello world')
      ^
SyntaxError: invalid syntax

The parser repeats the offending line and displays little arrows
pointing at the place where the error was detected. Note that this is
not always the place that needs to be fixed. In the example, the
error is detected at the function "print()", since a colon (":") is
missing just before it.

The file name ("<stdin>" in our example) and line number are printed
so you know where to look in case the input came from a file.

8.2. Exceptions
=====
```

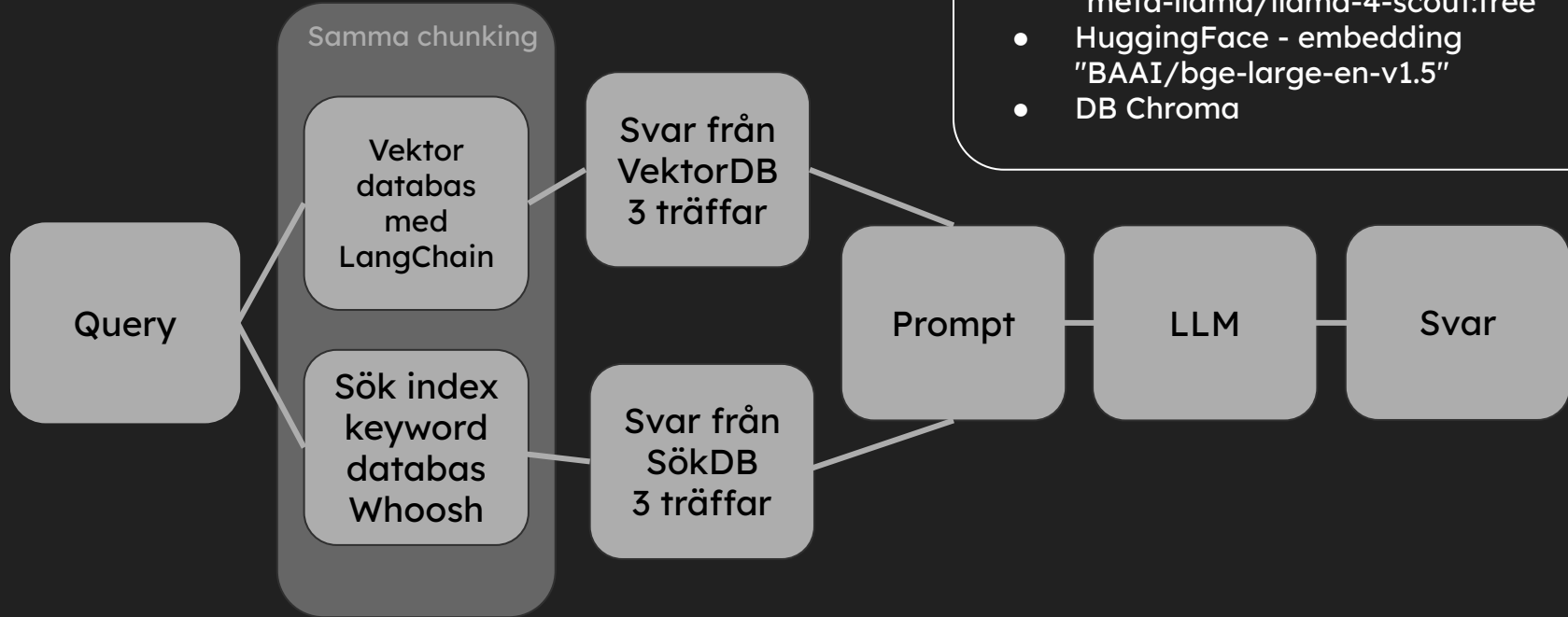
Bra
grej

Tog fram små script som analyserade dokumenten,
och visade statistik för chunkingen

Utmaningar

- Olika formateringar på rubriker
- Olika typer av innehåll - förklarande texter - perfekt för RAG
- Listor med funktioner och begrepp - funkar inte lika bra
- Förklaringar av olika begrepp, uttryck, funktioner - hittas inte alltid av en RAG som analyserar på kontext,
- Funktioner som är “depreciated”
- Olika versioner

Lösning - Hybrid RAG



- 14 506 chunks
- OpenRouter
"meta-llama/llama-4-scout:free"
- HuggingFace - embedding
"BAAI/bge-large-en-v1.5"
- DB Chroma

Chunking struktur

Chunking logik

- Chunking på
- Main rubrik
- Sub rubrik
- Sub sub rubrik
- Funktion
- Tabell
- Kodblock

Metadata struktur

- Folder
- Sidnamn
- Main rubrik
- Sub rubrik
- Sub sub rubrik
- Funktion
- Tabell
- Kodblock

Prompt

SEMANTIC SEARCH RESULTS (related concepts):

{vector_context}

KEYWORD SEARCH RESULTS (exact matches):

{keyword_context}

Question: {question}

IMPORTANT GUIDELINES:

****Grammar Notation Recognition:****

If the user asks about terms with underscores (like `set_display`,
`list_display`, `function_def`),

these might be grammar notation from Python's language reference, not
functions.

Look for BNF notation patterns like "term ::= definition" in the search
results.

Version

`**Python Version Context (Current: 3.13):**`

When interpreting version information:

- Versions 3.12 and below: Already released, past versions
- Version 3.13: Current version (what users are assumed to be using)
- Version 3.14 and above: Future versions, not yet released

For deprecation timelines, always contextualize:

- "Deprecated since X.Y" - explain how long it's been deprecated
- "Planned removal in X.Y" - explain how urgent the migration is

Spelling

****Spelling Error Detection:****

If your confidence in this answer is below 70% (especially due to not finding relevant information),

check if the question contains potential spelling errors for technical terms like:

- Module names (e.g., "asynchio" → "asyncio")
- Function names (e.g., "lenght" → "length")
- Method names (e.g., "apend" → "append")
- Keywords (e.g., "yeild" → "yield")

If you suspect spelling errors, suggest the correct spelling and search again mentally.

Response instruction

****Response Instructions:****

- Use information from both search results - prioritize exact matches for syntax
- If keyword results show exact syntax, include it
- If semantic results provide better explanations, use those
- Mention specific modules/functions when relevant
- For code examples, preserve exact syntax from the documentation
- Keep answers concise but complete
- Always provide practical examples when possible

Answer: ""

Möjlig vidareutveckling

- **Vikta upp depreciation**
- **Skruva mer på accuracy**
- **Vikta upp vissa delar, som PEP**
- **Addera fler bibliotek, md. filer**
- **Integrera i kodning med en LLM:** MCP eller wrapping. Så att LLM alltid promptas till att använda det senaste.
- Testa med olika LLMer för att hitta den bästa.

- **Addera GraphRAG** - för att tjänsten ska kunna diskutera kontext och göra jämförelser mellan olika funktioner och delar i dokumentationen. Det skulle göra det lättare för lösningen att svara på frågor om hur saker hänger ihop. Eftersom Pythondokumentationen i sig har vissa kopplingar men är ganska kortfattad kring dessa. Tänker jag att ett sånt upplägg skulle vinna på att man kompletterar den rena python-dokumentationen med exempelkod, eller ytterligare dokumentation kring Python. En utmaning är då att fånga upp utdaterad kod, ett av syftena med RAGen är att kunna ge svar på vad som är det senaste. Något som LLMer inte alltid klarar av.