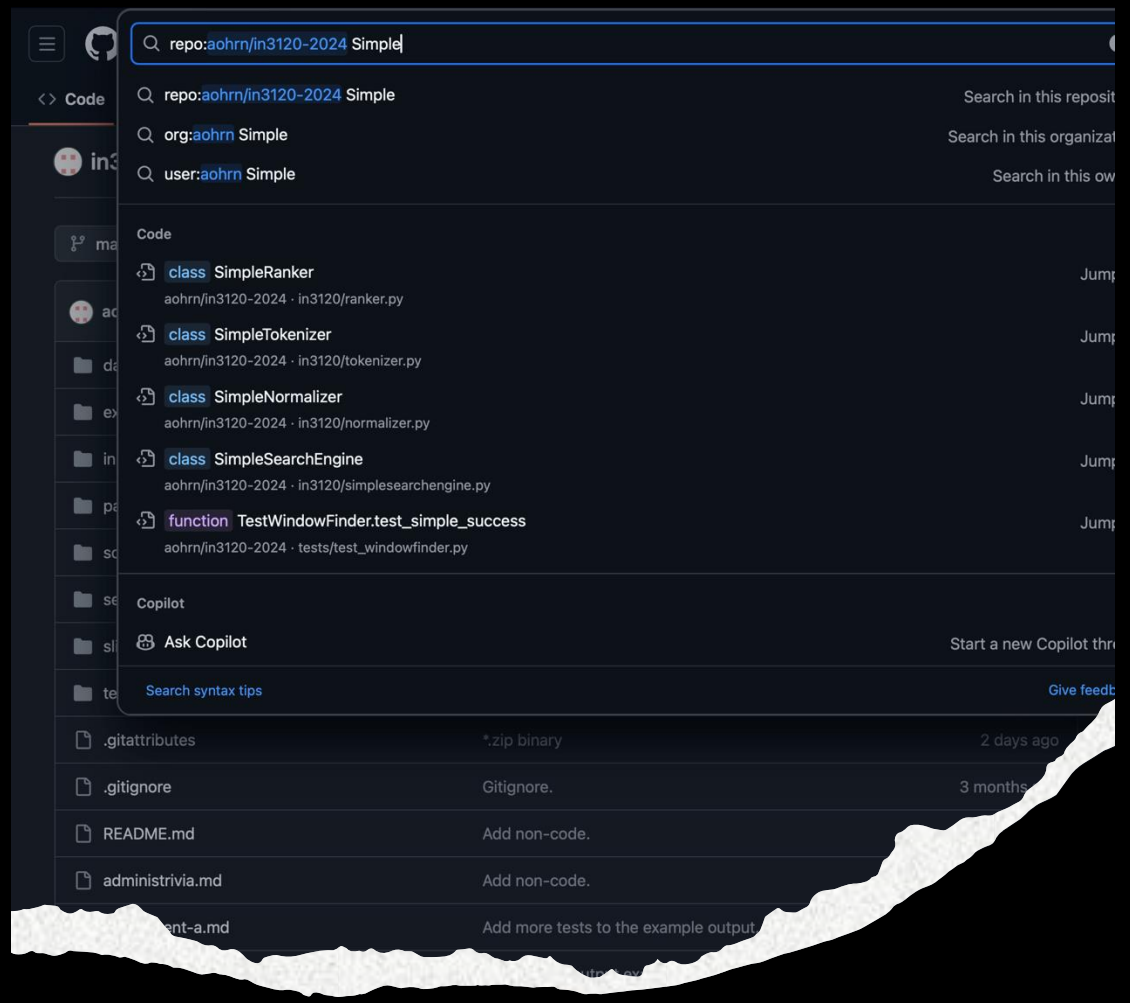
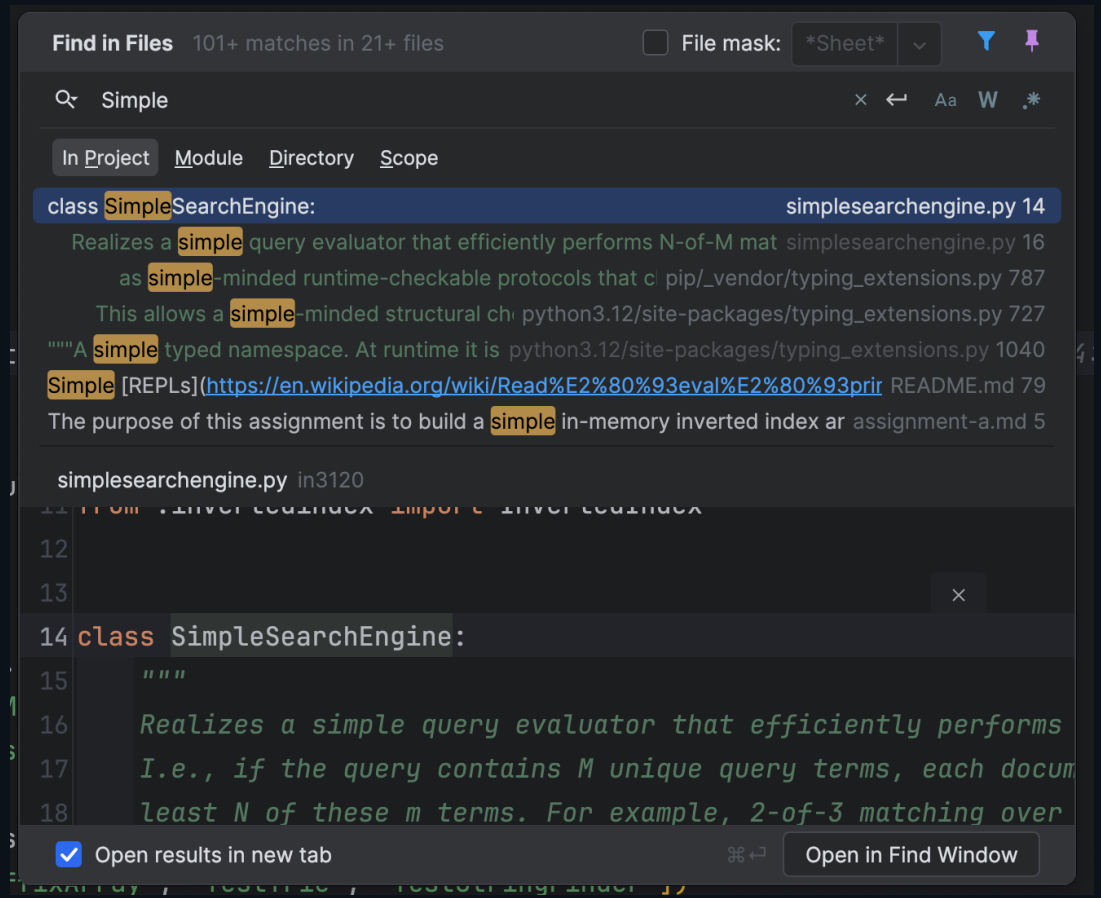
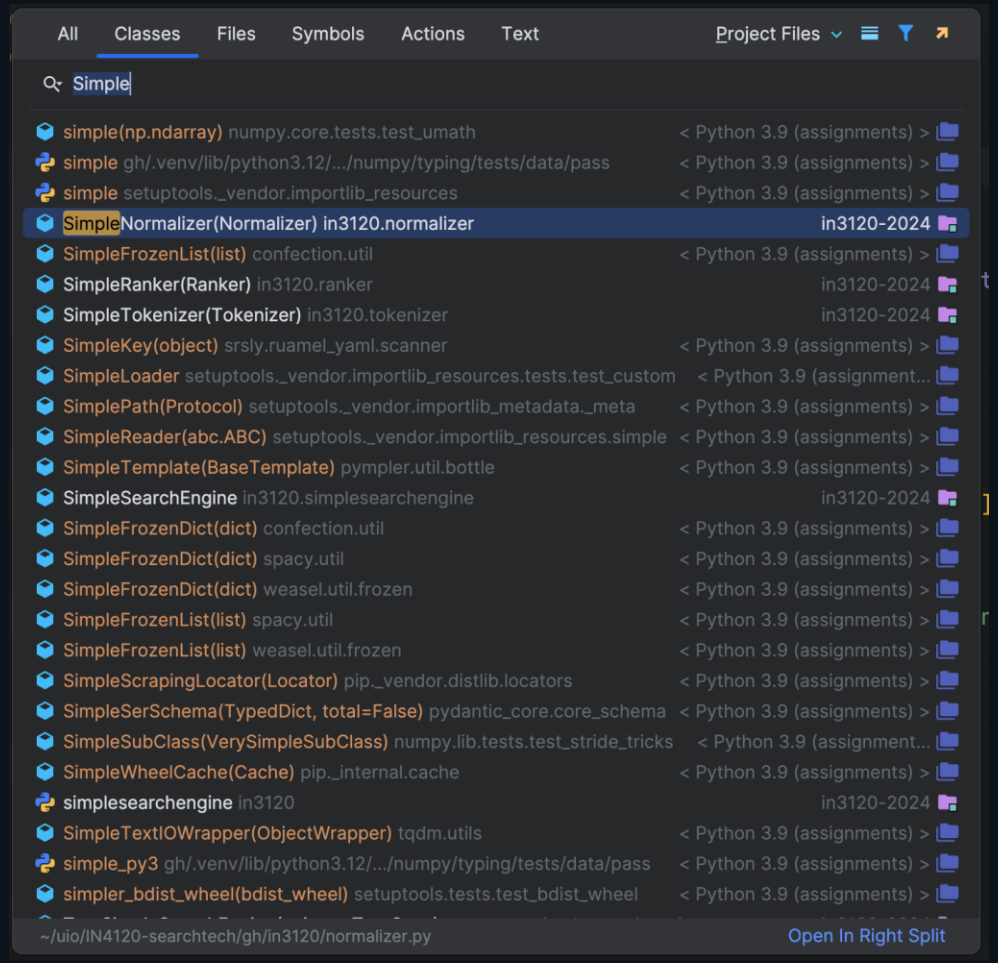
Tokenization og søk i code

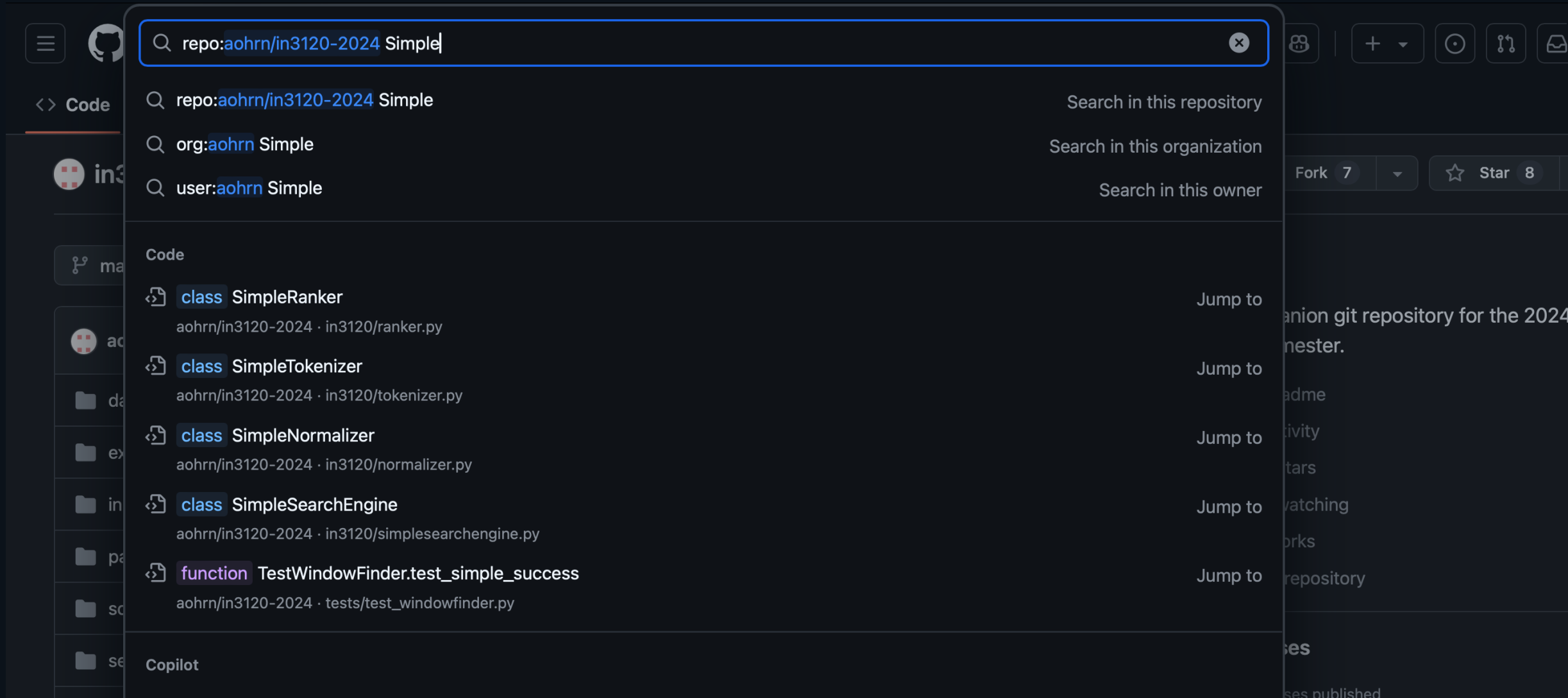


Fritekstsøk vs Symbolsøk





Samme på GitHub



Supernaivt eksempel i in3120-2024 1/4

```python
from in3120 import Tokenizer
from typing import Iterator, Tuple
import tokenize
from io import StringIO

class PythonTokenizer(Tokenizer):  3 usages
    """
    A simple tokenizer for Python files that can identify function and variable names
    """

    def __init__(self):
        pass

    def spans(self, buffer: str) -> Iterator[Tuple[int, int]]:...

    def strings(self, buffer: str) -> Iterator[str]:
        tokens = tokenize.generate_tokens(StringIO(buffer).readline)
        for token in tokens:
            if token.type == tokenize.NAME:
                yield token.string
```

Supernaivt eksempel i in3120-2024 2/4

```
files = [os.path.join(directory, file) for file in os.listdir(directory)
         if os.path.isfile(os.path.join(directory, file))]
for i, file in enumerate(files):
    with open(file, "r") as f:
        content = f.read()
    corpus.add_document(in3120.InMemoryDocument(i, fields: {
        "filename": file.split("/")[-1],
        "content": content
    }))

tokenizer = in3120.PythonTokenizer()
index = in3120.InMemoryInvertedIndex(corpus, fields: ["content"], self.__normalizer, tokenizer)

engine = in3120.SimpleSearchEngine(corpus, index)
ranker = in3120.SimpleRanker()
```

Supernaivt eksempel i in3120-2024 3/4

```
matches = list(
    engine.evaluate(
        query: "__tokenizer __normalizer __inverted_index category",
        options: {"match_threshold": 0.5, "hit_count": 100},
        ranker
    ))

for match in matches:
    print(match)
```

Supernaivt eksempel i in3120-2024 4/4

```
{'score': 26.0, 'document': {'document_id': 15, 'fields': {'filename': 'naivebayesclassifier.py',
{'score': 7.0, 'document': {'document_id': 19, 'fields': {'filename': 'simplesearchengine.py', 'co
{'score': 7.0, 'document': {'document_id': 16, 'fields': {'filename': 'suffixarray.py', 'content':
{'score': 6.0, 'document': {'document_id': 11, 'fields': {'filename': 'windowfinder.py', 'content'
{'score': 6.0, 'document': {'document_id': 1, 'fields': {'filename': 'editsearchengine.py', 'conte
{'score': 5.0, 'document': {'document_id': 7, 'fields': {'filename': 'similaritysearchengine.py',
{'score': 5.0, 'document': {'document_id': 3, 'fields': {'filename': 'shinglegenerator.py', 'conte
{'score': 2.0, 'document': {'document_id': 17, 'fields': {'filename': 'stringfinder.py', 'content'
```