

매직 명령어(주피터 노트북을 더 강력하게 만드는 비밀 무기)

매직 명령어는 주피터 노트북에서 데이터 분석과 프로그래밍 작업을 더 효율적으로 만들어주는 특별한 지시어이다. 이 명령어들은 퍼센트 기호(%)로 시작하며, 일반 Python 코드로는 달성하기 어려운 많은 작업을 수행하는 데 도움을 준다.

1. 매직 명령어의 유형

- 라인 매직 명령어: 단일 % 로 시작하며, 해당 명령어가 있는 줄에만 영향을 미친다
- 셀 매직 명령어: %% 로 시작하며, 전체 코드 셀에 영향을 미친다

2. 일반적인 매직 명령의 예

2.1 차트 표시하기

```
In [1]: 1 %matplotlib inline
```

이 명령을 사용하면 새 창을 띄우는 대신 노트북에 차트를 바로 표시할 수 있다.

2.2. 타이밍 코드 실행

```
In [2]: 1 %timeit [1, 2, 3, 4, 5].sort()
```

127 ns \pm 0.28 ns per loop (mean \pm std. dev. of 7 runs, 10,000,000 loops each)

이는 한 줄의 코드 실행 시간을 측정하는 것으로 알고리즘 효율성을 비교하는 데 매우 유용하다.

```
In [3]: 1 %%timeit
2 total = 0
3 for i in range(1000):
4     total += i
```

44.1 μ s \pm 1.64 μ s per loop (mean \pm std. dev. of 7 runs, 10,000 loops each)

이는 전체 코드 블록의 실행 시간을 측정한다.

2.3. 시스템 관련 명령

```
In [ ]: 1 %pwd # Display the current working directory
2 %ls # List the contents of a folder
3 %cd folder_path # Change directory
```

2.4. 외부 파일 실행

```
In [ ]: 1 %run my_script.py
```

명령줄에서 실행하는 것과 마찬가지로 외부 Python 파일을 실행한다.

2.5. 다른 언어 간 전환

```
In [ ]: 1 %%html
2 <div style="color: red; font-size: 20px;">This is HTML content</div>
```

```
In [ ]: 1 %%javascript
2 alert("This is JavaScript code!");
```

2.6. 사용 가능한 모든 매직 명령 보기

```
In [ ]: 1 %lsmagic
```

사용 가능한 모든 마법 명령의 목록이 표시된다.

2. 도움말을 받는 방법

특정 마법 명령어를 사용하는 방법을 배우고 싶다면:

```
In [ ]: 1 %magic # Get general help for all magic commands
2
3 %magic command? # Get help for a specific magic command, such as %timeit?
```

3. 왜 매직 커맨드를 사용하나요?

1. 시간 절약: 한 줄의 코드로 복잡한 작업을 완료할 수 있다.
2. 효율성 향상: 빠른 성능 측정, 차트 표시
3. 기능 향상: 시스템 명령 실행, 다양한 언어와의 통합
4. 워크플로우 개선: 데이터 분석 시 코드 세부 사항보다 문제에 집중.

매직 커맨드는 Jupyter Notebook을 단순한 코드 편집기가 아닌 강력한 대화형 컴퓨팅 환경으로 만들어 업무 효율성을 크게 높여준다!

```
In [ ]: 1
```

```
In [ ]: 1
```