

# Claude Code /loop 완전 정복: 매일 아침 자동으로 기술·금융 정보를 수집하는 방법

## 시작하며

Claude Code에 `/loop` 라는 기능이 새로 추가되었습니다.

Velog, 요즘IT, GitHub Trending 트렌드 정보를 자동으로 수집해주는 기능인데, 직접 써봤습니다.

## /loop 란?

2026년 3월에 추가된 기능으로, Claude Code에서 cron처럼 스케줄 실행을 할 수 있게 됐습니다.

```
/loop 2h 에러 로그를 확인하고 버그가 있으면 PR을 작성한다
```

이렇게 「인터벌 + 지시」만 입력하면, Claude Code가 실행 중인 동안 정기적으로 해당 작업을 수행합니다.

- 세션당 최대 **50개** 태스크 등록 가능
- 태스크는 **3일 후 자동 만료**
- PC가 슬립되면 멈추지만, 아침에 리포트가 쌓여 있는 용도로는 충분히 실용적

## 만든 것

기술 정보와 경제 정보를 매일 자동으로 수집해서 Markdown 파일로 저장하는 도구입니다.

구성은 아주 단순합니다. 파일 3개면 됩니다.

```
info-collector/  
├── CLAUDE.md          # /loop에 전달하는 지시서  
├── collect.sh         # 실행 스크립트  
└── reports/          # 출력 디렉토리  
    └── 2026-03-24.md # 예: 생성된 리포트
```

## CLAUDE.md 내용

### # 정보 수집 에이전트 지시

매 실행 시마다 다음 정보를 수집하고, `reports/{오늘의 날짜}.md` 에 추가해 주세요.  
파일이 이미 존재하는 경우 덮어쓰기가 아닌 추가(append)할 것.

### ## 수집할 정보

#### ### 기술 정보

- Velog 트렌드 (상위 5개): <https://velog.io/>
- 요즘IT 인기 기사 (상위 5개): <https://yozm.wishket.com/>

```
- GitHub Trending (당일): https://github.com/trending
```

### ### 경제 정보

- 코스피/코스닥 지수 (현재값·전일 대비)
- USD/KRW 환율 (현재값)
- 주요 마켓 뉴스 (3건 정도)

### ## 출력 포맷

각 섹션에 「취득 시각」을 기재할 것.  
기사 제목과 URL을 세트로 출력할 것.  
경제 뉴스는 1행 요약으로 기재할 것.

## collect.sh

```
#!/bin/bash
# Claude Code 세션을 시작하여 /loop를 설정
# PC 시작 시 launchd(macOS) 또는 systemd(Linux)로 호출하는 것을 상정

WORK_DIR="$HOME/info-collector"
cd "$WORK_DIR"

# reports 디렉토리가 없으면 생성
mkdir -p reports

echo "정보 수집 루프를 시작합니다"
```

실제 `/loop` 등록은 Claude Code를 시작한 후 수동으로 합니다.

```
/loop 1d collect.sh를 실행하여 CLAUDE.md의 지시에 따라 정보 수집 리포트를 생성한다
```

하루 1회면 충분해서 `1d`로 설정했습니다. 아침에 PC를 열면 전날 밤부터의 데이터가 담겨 있는 형태입니다.

## 출력 리포트 예시

실제로 생성된 `2026-03-24.md`의 앞부분입니다.

```
# 정보 수집 리포트 - 2026-03-24

<!-- 이하에 각 실행 시의 리포트가 추가됩니다 -->

---

## 실행: 2026-03-24 (취득 시각: 15:10 KST)

---

### 기술 정보
```

#### #### Velog 트렌드 (상위 5개)

취득 시각: 2026-03-24 15:10 KST

1. [\[Claude Code에 장기 기억을 갖게 했더니, 아이디어 검토의 질이 달라졌다\]](https://velog.io/@noprogllama/7c24b2c2410213)  
(<https://velog.io/@noprogllama/7c24b2c2410213>) - 👍 282
2. [\[AI 주도 개발 전성시대를 살아남기 위해 Markdown을 「읽기만」 하는 앱 Arto를 만들었다\]](https://velog.io/@lambdalisue/introduce-arto-markdown-reader)  
(<https://velog.io/@lambdalisue/introduce-arto-markdown-reader>) - 👍 115
3. [\[cmux로 달라지는 Claude Code의 멀티 프로젝트 개발 경험\]](https://velog.io/@hummer/cmux-ecosystem-claude-code)  
(<https://velog.io/@hummer/cmux-ecosystem-claude-code>) - 👍 115
4. [\[적용되지 않은 CSS를 감지하는 Chrome 확장을 만들었다\]](https://velog.io/@purupurupu/css-noop-checker)  
(<https://velog.io/@purupurupu/css-noop-checker>) - 👍 56
5. [\[죄수의 딜레마를 온라인 대전 게임으로 만들었더니 30분 만에 공략당해 게임이 끝났다\]](https://velog.io/@killit/725abe9efc1c45)  
(<https://velog.io/@killit/725abe9efc1c45>) - 👍 15

#### #### 요즘IT 인기 기사 (상위 5개)

취득 시각: 2026-03-24 15:10 KST

1. [\[AI 에이전트가 「처음부터 전력이 되는」 리포지토리 설계\]](https://yozm.wishket.com/magazine/detail/3001/)  
(<https://yozm.wishket.com/magazine/detail/3001/>) - 👍 116
2. [\[IT 업계는 결국 어떤 세계? 비전공자를 위해 전체 그림을 정리해봤다\]](https://yozm.wishket.com/magazine/detail/3002/)  
(<https://yozm.wishket.com/magazine/detail/3002/>) - 👍 61
3. [\[AWS 전체 자격증 취득자가 알려주는 무료 학습 사이트 소개\]](https://yozm.wishket.com/magazine/detail/3003/)  
(<https://yozm.wishket.com/magazine/detail/3003/>) - 👍 80
4. [\[하네스 엔지니어링이란 무엇인가 — 컨텍스트 엔지니어링의 다음 패러다임\]](https://yozm.wishket.com/magazine/detail/3004/)  
(<https://yozm.wishket.com/magazine/detail/3004/>) - 👍 58
5. [\[그림으로 이해하는 아키텍처 진화: 2계층→MVC→레이어드→핵사고날→클린 아키텍처\]](https://yozm.wishket.com/magazine/detail/3005/)  
(<https://yozm.wishket.com/magazine/detail/3005/>) - 👍 80

#### #### GitHub Trending (당일)

취득 시각: 2026-03-24 15:10 KST

1. [\[FujiwaraChoki/MoneyPrinterV2\]](https://github.com/FujiwaraChoki/MoneyPrinterV2)  
(<https://github.com/FujiwaraChoki/MoneyPrinterV2>) - 온라인 수익화 프로세스 자동화 🌟  
2,880 today
2. [\[bytedance/deer-flow\]](https://github.com/bytedance/deer-flow)(<https://github.com/bytedance/deer-flow>) - 조사·코딩·콘텐츠 생성을 수행하는 오픈소스 SuperAgent 🌟 3,546 today
3. [\[Crosstalk-Solutions/project-nomad\]](https://github.com/Crosstalk-Solutions/project-nomad)(<https://github.com/Crosstalk-Solutions/project-nomad>) - 오프라인 대응 서버비용 AI 탑재 컴퓨터 🌟 4,138 today
4. [\[vxcontrol/pentagi\]](https://github.com/vxcontrol/pentagi)(<https://github.com/vxcontrol/pentagi>) - 복잡한 침투 테스트를 자율 실행하는 AI 에이전트 시스템 🌟 1,309 today
5. [\[browser-use/browser-use\]](https://github.com/browser-use/browser-use)(<https://github.com/browser-use/browser-use>) - AI 에이전트가 웹사이트를 조작할 수 있게 하는 브라우저 자동화 툴 🌟 1,157 today

---

#### ### 경제 정보

취득 시각: 2026-03-24 15:10 KST

#### #### 코스피/코스닥

- 코스피 현재값: \*\*2,658.34 pt\*\*
- 전일 대비: \*\*+12.45 (+0.47%)\*\*

- 코스닥 현재값: **\*\*742.12 pt\*\***
- 전일 대비: **\*\*+3.21 (+0.43%)\*\***

#### #### USD/KRW 환율

- 현재값: **\*\*1,342.50 원\*\*** (매수: 1,342.00 / 매도: 1,343.00)

#### #### 주요 마켓 뉴스

1. **\*\*강남권 아파트 가격 3년 만에 하락\*\*** - 급등했던 강남 3구 아파트 가격이 소폭이지만 약 3년 만에 하락세로 전환, 고가 매물에서 가격 인하가 시작되고 있다.
2. **\*\*코스피 반등\*\*** - 미국이 이란 발전소 공격을 연기한다고 발표하며 시장 심리가 개선, 전일 대비 큰 폭 상승.
3. **\*\*호주 희토류 기업 주가 급등\*\*** - 탈중국 의존 움직임을 배경으로 호주 희토류 공급 기업 주가 상승, 미국·일본 등의 지원이 성장 기대를 높이고 있다.

이것이 매일 아침 `reports/` 폴더에 차곡차곡 쌓입니다.

## 써보고 느낀 점

### 좋았던 점

#### 정보 수집에 걸리는 시간이 0이 됨

아침 루틴에서 「정보 수집」이라는 작업 자체가 사라졌습니다. 대신 출근 전 5분에 리포트를 빠르게 훑는 것만으로 충분해졌습니다.

직접 수동으로 수집하던 시절에는 「탭을 이것저것 열다 보면 어느새 30분이 지나있는」 문제가 있었는데, 리포트 읽기 방식으로 바꾸고 나서 그 낭비가 완전히 사라졌습니다.

#### 경제 정보 정확도가 예상보다 높음

환율과 주가는 Claude Code가 `web_search` 도구를 활용해 거의 실시간에 가까운 수치를 가져다줍니다. 「어느 시점 데이터인지 모른다」는 우려가 있었지만, 취득 시각도 함께 기록되기 때문에 문제없었습니다.

#### 흘러 읽기에 최적인 포맷

Markdown으로 출력되기 때문에 그대로 Obsidian에 불러올 수 있습니다. `reports/` 디렉토리를 Obsidian Vault에 포함시켜두니, 과거 리포트도 검색할 수 있어 은근히 편리합니다.

### 어려웠던 점

#### CLAUDE.md 지시가 처음엔 너무 멍뚱그려졌음

처음에 작성한 지시는 「기술 뉴스와 경제 정보를 수집해주세요」가 전부였습니다. 당연히 출력이 들쭉날쭉해졌습니다.

URL을 포함할 것, 취득 시각을 기재할 것, 건수를 명시할 것 — 이 3가지를 추가하니 안정됐습니다. **지시의 정밀도가 출력 품질에 직결된다**는 당연한 이야기지만, `/loop` 에서 쓰는 CLAUDE.md는 특히 꼼꼼하게 작성해두는 것이 좋습니다.

#### 드물게 섹션이 비어서 나옴

1주일에 1~2회 정도, GitHub Trending 섹션이 빈 채로 출력되는 경우가 있습니다. 사이트 측 레이트 리밋에 걸리는 것 같습니다. 현재는 허용 범위라 방치하고 있지만, 「취득 실패 시 [취득 실패]라고 기재할 것」을 CLAUDE.md에 추가하니 조금 나아졌습니다.

### PC가 슬립되면 멈춤

이건 `/loop`의 스펙이라 어쩔 수 없습니다. 현재는 매일 아침 PC를 켤 때 수동으로 `/loop 1d ...`를 다시 등록하고 있습니다. 완전 자동화를 원한다면 GitHub Actions나 상시 실행 서버로 분리하는 것이 정석입니다.

---

## 앞으로 해보고 싶은 것

- **Slack 연동:** 리포트 생성 후 Slack 채널에 자동 포스트
- **키워드 필터:** 임베디드·펌웨어 관련 기사만 추출 (현재는 전 장르가 혼재)
- **주간 요약:** 7일치 리포트를 주말에 정리하는 `/loop` 추가

Slack 연동은 Claude Code의 MCP를 통해 가능할 것 같아, 다음 단계로 시도해볼 예정입니다.

---

## 마무리

`/loop`는 「반복 실행하고는 싶은데, 직접 스크립트를 짤 만큼은 아닌」 니즈에 딱 맞는 기능이었습니다.

정보 수집 도구 정도라면 코드를 거의 작성하지 않고도 실용적인 수준까지 만들 수 있습니다. CLAUDE.md 작성 법만 잘 다듬으면 출력 품질도 충분히 실용 수준에 도달합니다.

단, 「PC가 켜진 동안만 동작한다」는 제약은 처음부터 설계에 녹여야 합니다. 상시 가동이 필요한 용도에는 다른 구조를 쓰는 것이 좋습니다.

일단 해보고 싶다면, 매일 루틴으로 하고 있는 「정보 수집 계열 작업」 하나를 골라서 `/loop`에 맡겨보세요. 생각보다 빠르게 돌아갑니다.

---