



DartLab Cheat Sheet

종목코드 하나로 공시, 재무제표, 분석, 리포트까지

★ Polars-first

✔ pandas-compatible

✚ DART + EDGAR

1 / 2

왜 DartLab인가?

공시 데이터를 빠르고, 쉽게, 깔끔하게 분석합니다.

- ✔ XBRL 표준
- ✔ 구조화
- ✔ 자동화
- ✔ 재현성
- ✔ Polars-first ... and more

⚠ RAW 데이터 (지저분한 원천)

- ✗ 영업이익(손실)
- ✗ 영업이익 / Loss from Ops
- ✗ 영업이익 (K-IFRS)
- ✗ 영업이익 [누적]
- ✗ OPERATING INCOME...

✔ 정규화 & 구조화 (표준화)

- 계정 표준화
- 형식 통일
- 누락 / 중복 정리
- 기간 정렬

Polars DataFrame (빠르고 메모리 효율적)

- 수치형 고속 처리
- 대용량에 강함
- 병렬 / 지연 실행

Compare / Analyze / Report (태고 • 분석 • 리포트)

- 지표 비교
- 패턴 발견
- 스토리로 정리

DartLab은 단순한 다운로드가 아닌, 금융 데이터 분석을 위한 워크플로우입니다.

자주 쓰는 시작 순서

- 1 회사/종목 지정
- 2 섹션/기간 확인
- 3 항목 & 기간 선택
- 4 데이터 보기
- 5 스토리 & 내보내기

패널 축소 : company * sections → show
• select → story → export

1 Start Here

회사/종목을 지정하기

```
import dartlab
c = dartlab.Company("005930")
```

입력 예시 (accepted inputs)

종목코드 (6자리)

티커 (Ticker)

회사명 (영/영)

005930 →

AAPL → Apple

삼성전자 → 삼성전자

삼성전자

MSFT → Microsoft

Apple → Apple

★ TIP • 코드, 티커, 회사명 모두 시작점이 될 수 있습니다.

2 먼저 확인

섹션 x 기간을 한눈에 보는 수평 맵

c.index	c.sections	sections = 주제 x 기간 커버리지 맵						
Company/Overview	2022Q1	2022Q2	2022Q3	2023Q4	2024Y1	...	2024FY	
BusinessOverview	✓	✓	✓	✓				
FinancialStatement	✓	✓	✓	✓				
BusinessOverviewLight	✓	✓	✓	✓	✓	—	✓	
FinancialStatementLight	✓	✓	✓	✓				
IFRS연결(연간)	✓	✓	✓	✓				✓
IFRS연결(분기)	✓	✓	✓	✓	—	—	—	—
주요속보	✓	✓	✓	✓				
기타Disclosure	✓	✓	✓	✓	—	—		

✓ = 있음 — = 없음 • sections = 주제 x 기간 커버리지 맵 (수평 보기)

3 재무제표 보기

주요 재무제표 & 지표 확인

```
c.show("IS")
c.show("BS")
c.show("CF")
c.show("CF", freq="Y")
c.ratio("ratios")
```

연결 | 단위 : 억원 | Polars DataFrame

account	2024	2023	2022
매출액	2,558,355	2,566,136	3,022,314
영업이익	288,532	366,558	433,766
당기순이익	222,186	247,036	399,074
...			

★ TIP • 보기만으로 기업의 핵심 스냅샷을 빠르게 파악하세요.

4 필요한 계정만 선택

관심 계정만 추려서 분석

```
c.select("IS", ["매출액", "영업이익"])
```

선택 전 (IS 전체)

account	2024	2023	...
매출액			•
영업이익			•
영업이익(분)			•
당기순이익			•
...			

선택 후 (필요한 계정만)

account	2024	2023	...
매출액			
영업이익			

★ TIP • 필요한 계정만 골라 데이터가 가볍고 분석이 빨라집니다.

5 숫자를 스토리로

분석 → 인사이트를 리포트로

```
report = c.story("executive")
print(report.toMarkdown())
```

리포트 목차 (예시)

- 1 회사 개요
- 2 요약 지표
- 3 재무 성과
- 4 수익성
- 5 안정성
- 6 리스크

인사이트 미리보기

- 매출은 전년 대비 0.x% 증가했습니다.
- 영업이익은 전년 대비 0.x% 증가했습니다.
- 부채비율은 안정 수준을 유지하고 있습니다.
- 주요 리스크: 원/환 변동, 반도체 업황 변동 등

★ TIP • 스토리는 숫자를 '이야기'로 바꿔 의사결정에 도움을 줍니다.

6 결과 내보내기

분석 결과를 손쉽게 저장



DataFrame (pkl/csv)



CSV



Excel (xlsx)



Parquet



Markdown (.md)

```
df.write_csv("is_selected.csv")
df.write_excel("is_selected.xlsx")
df.write_parquet("is_selected.parquet")
report.to_markdown("executive_report.md")
```

★ TIP • 원하는 형태로 빠르게 저장하고 공유하세요.

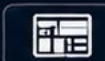
추천 첫 흐름 (한눈에 보기)



Company(code)
회사 / 종목 지정



sections / index
가능 범위 확인



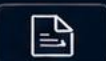
show("IS")
재무제표 보기



select(accounts)
필요 계정 선택



story("executive")
스토리 리포트 생성



export
내보내기

5줄 스타터 워크플로우

```
1 import dartlab
2 c = dartlab.Company("005930")
3 c.index, c.sections → 확인
4 df = c.select("IS", ["매출액", "영업이익"])
5 report = c.story("executive"); print(report.toMarkdown())
```

★ TIP

더 많은 옵션은 공식 문서에서 확인하세요.

Polars-first 설계로
대용량 데이터도 빠르고 안정적입니다.



공식 문서: eddmpython.github.io/dartlab



GitHub: github.com/eddmpython/dartlab



이슈 제보: github.com/eddmpython/dartlab/issues



DartLab Cheat Sheet

종목코드 하나로 공시, 재무제표, 분석, 리포트까지

Architecture

5 Core Engines

Polars-first

pandas-compatible

2 / 2

DartLab 아키텍처 (5단계 파이프라인)

1. 소스 레이어 DART 데이터 원천



DART
EDGAR
XBRL
Price / 가격/시세
Flow / 현금
Macro / 뉴스
News / 뉴스

모든 데이터의 시작점

2. 수집 레이어 종합 수집 · 정제

gather (수집) 공시/시세/뉴스
price (가격/시세) 일/주/월 시계열
flow (수급/거액) 두/월/기/연/일별
news (뉴스/이슈) 키워드/리스크
macro (거시/경제) 금리/환율/지표
consensus (관측치) 추정/목표주가
collect (모든 원천) 통합 적재

원천 데이터를 모아 일관되게 수집

3. 기업 레이어 정규화 · 표준화 (Normalized)

Company(code) 표준화
Filing 구조화 (DART/EDGAR 등)
재무 데이터 (Finance) 정규화 / 기준 통일
가로 비교 가능 (Horizontal Comparability)
모든 회사 · 모든 기간을 같은 기준으로!

★ 표준화된 계정 덕분에 기업/기간 간 비교가 가능합니다.

4. 엔진 레이어 분석 · 계산 엔진

show select
scan analysis
credit quant
macro ... (확장 엔진)
필요한 엔진을 조합해 인사이트 탐색

5. 인사이트 레이어 스토리 · 리포트 · 전달

story / 보고서
ask / 질의
markdown
html
json
분석 결과를 다양한 형태로 전달

자주 쓰는 시작 순서

- 1 회사/종목 지정
- 2 섹션/기간 선택
- 3 항목 & 기간 선택
- 4 데이터 보기
- 5 스토리 & 내보내기

핵심 흐름: Company → sections → show → select → story → export

5대 핵심 엔진 (자주 쓰는 실행 흐름)

1 Company

```
c = dartlab.Company("005930")
c.index
c.sections
```



기업 데이터



섹션 목록



기간 커버리지

2 show / select

```
c.show("IS")
c.select("IS", ["매출액", "영업이익"])
```



IS

기간/항목 선택



조회/선택

3 gather

```
g = dartlab.gather()
g.price("005930")
g.news("삼성전자")
g.collect("005930")
```



수집



경제



누적



통합 적재

4 scan

```
dartlab.scan("ratio", "영업이익률")
dartlab.scan("zscore", "부채비율")
```



지표 스크리닝



조건 탐색



순위/분포

5 story / ask

```
c.story("executive")
dartlab.ask("삼성전자 수익성) 최근 추이")
```



자연어 질의



자동 리포트

즉시 요약

심층 분석 엔진 (분석을 더 깊고 다각도로)

A AI analysis (심층 재무분석)

```
c.analysis("financial")
c.analysis("quality")
```

- 수익성/안정성/성장성/현금흐름
- 동종업계/기간 비교 분석
- 이상 징후 탐지 · 알림
- 핵심 지표 요약 · 인사이트 · 리스크 요약



B credit (신용/재무건전성)

```
c.credit()
c.credit(detail=True)
```

- 재무제표 기반 신용평가
- 신용등급 추정 / 위험 신호 탐지
- 재무상환 능력 / 레버리지 / 유동성
- 핵심 지표 / 스코어 / 시그널



C quant (퀀트/팩트)

```
dartlab.quant("factor")
dartlab.quant("momentum")
```

- 모멘텀/밸류/유틸리티/사이즈/수익성 팩트
- 팩트 스코어 산출 · 팩트세트
- 상위/하위 순위 선택
- 팩트 스코어 & 리스크어

MTM	VAL	QTY	SIZE	PROF
82	61	75	48	70

D macro (거시/시장)

```
dartlab.macro("금리")
dartlab.macro("환율")
```

- 금리/환율/물가/통화공급
- 주요 거시 지표/경기 동향지수
- 사이클/국가 비교/상관 관계
- 거시 리스크 & 이벤트 모니터링



6-Act 사이클 요약

1. 회사 찾기
2. 섹션 보기
3. 항목 선택
4. 분석
5. 해석
6. 전달

Polars-first,
pandas-compatible

기본 설정: Polars DataFrame (권장)

```
df = c.show("IS")
df.head(5)
df.select("account", "2024.12")
df.head(5)
```

account	2024.12	2023.12
매출액	2,589,355	2,586,135
영업이익	289,532	306,558
당기순이익	222,186	247,036

빠른 · 메모리 효율 · 지연 평가

Polars로 빠르게 필터링/가공

```
import polars as pl
df = c.show("IS")
df.select(pl.col("당기순이익") > 0)
df.select([pl.col("account", "2024.12", "YoY")
.sort("account")])
```

account	2024.12	YoY
매출액	2,589,355	0.1%
영업이익	289,532	-5.5%
당기순이익	222,186	-10.1%

필터 선택 · 필터 · 정렬 · 집계 · 윈도우

필요할 때만 pandas 변환 (경계/내보내기)

```
pdf = df.to_pandas()
type(pdf)
pdf.head(5)
```

	1	2	2
account	2024.12	YoY	
매출액	2,589,355	0.1%	
영업이익	289,532	-5.5%	
당기순이익	222,186	-10.1%	

호환성 · 시각화 · 내보내기

★ 기본은 Polars, 필요할 때 pandas로 보완!

- 기본 연산/분석은 Polars로 최대한 빠르게 처리
- 시각화/레거시 라이브러리 호환이 필요할 때 pandas로 변환
- .to_pandas() 는 경계적으로 호출

핵심 원칙

- 표준화된 데이터로 가로 비교 가능
- 필요한 엔진을 조합해 인사이트 탐색
- 빠르고 일관된 워크플로우



도움이 필요하신가요?

예제, 사용법, 이슈 해결까지 도와드려요!



도움말 (Help)

help(dartlab)

이슈 제보: github.com/eddmypython/dartlab/issues



더 알아보기

eddmypython.github.io/dartlab



★ TIP

Polars-first가 기본이지만, pandas 생태계와의 호환을 보장합니다.

필요한 순간에만 pandas로 변환하면 최적의 성능과 호환성을 모두 얻을 수 있어요!

