

YAML, TOML, JSON cheatsheet

YAML (YAML Ain't Markup Language)

사람이 읽기 쉬운 데이터 직렬화 형식. 주로 설정 파일에 사용됩니다. 들여쓰기로 데이터의 계층 구조를 나타냅니다.

기본 문법

- 키-값 쌍: `key: value`
- 들여쓰기: 공백 2칸 또는 4칸을 사용하여 계층 구조를 표현합니다. 탭은 사용하지 않는 것이 좋습니다.
- 주석: `#` 뒤의 내용은 주석으로 처리됩니다.

데이터 타입

- 스칼라: 문자열, 숫자, 불리언 (`true/false`, `yes/no`).
 - 문자열은 따옴표 없이 사용할 수 있지만, 특수 문자가 포함된 경우 따옴표(' 또는 ")를 사용합니다.
- 시퀀스 (리스트/배열):
 - `item1`
 - `item2`
 - `key:`
 - `nested_item1`
 - `nested_item2`
- 매핑 (딕셔너리/객체):
 - `key:`
 - `nested_key1: value1`
 - `nested_key2: value2`

예시

```
# 사용자 정보
name: "John Doe"
age: 30
is_student: false
courses:
  - "History"
  - "Math"
address:
  street: "123 Main St"
  city: "Anytown"
```

TOML (Tom's Obvious, Minimal Language)

명확한 시맨틱을 가진 설정 파일을 위해 설계되었습니다. `key = value` 형식을 기본으로 하며, 테이블(섹션)을 사용하여 구조를 정의합니다.

기본 문법

- 키-값 쌍: `key = "value"`
- 테이블 (섹션): `[table_name]`
- 중첩 테이블: `[table.sub_table]`
- 주석: `#` 뒤의 내용은 주석으로 처리됩니다.

데이터 타입

- 문자열: ""로 묶습니다.
- 숫자: `123`, `3.14`
- 불리언: `true`, `false`
- 날짜/시간: `1979-05-27T07:32:00Z`
- 배열: `[1, 2, 3]`
- 인라인 테이블: `{ key1 = "value1", key2 = "value2" }`

예시

```
# 최상위 키-값
title = "TOML Example"

[owner]
name = "Tom Preston-Werner"
dob = 1979-05-27T07:32:00-08:00

[database]
server = "192.168.1.1"
ports = [ 8001, 8001, 8002 ]
data = [ ["delta", "phi"], [3.14] ]
enabled = true

[servers.alpha]
ip = "10.0.0.1"
dc = "eqdc10"

[servers.beta]
ip = "10.0.0.2"
dc = "eqdc10"
```

JSON (JavaScript Object Notation)

JavaScript 객체 문법을 따르는 경량의 데이터 교환 형식. 웹 API 등에서 널리 사용됩니다.

기본 문법

- 키-값 쌍: `"key": "value"` (키는 항상 큰따옴표로 묶어야 함)
- 객체: `{}`로 묶습니다.
- 배열: `[]`로 묶습니다.

- 주석: 공식적으로 지원하지 않습니다.

데이터 타입

- 문자열: ""로 묶습니다.
- 숫자: `123`, `3.14`
- 불리언: `true`, `false`
- 배열: `[1, "apple", true]`
- 객체: `{"key": "value"}`
- `null`: 빈 값을 나타냅니다.

예시

```
{
  "name": "John Doe",
  "age": 30,
  "isStudent": false,
  "courses": [
    "History",
    "Math"
  ],
  "address": {
    "street": "123 Main St",
    "city": "Anytown"
  }
}
```

주요 차이점 요약

특징	YAML	TOML	JSON
주 용도	설정 파일	설정 파일	데이터 교환(API)
사람 가독성	매우 높음	높음	보통
주석	지원 (#)	지원 (#)	미지원
계층 구조	들여쓰기	테이블 []	중괄호 { }
데이터 타입	다양함 (자동 인식)	엄격함	제한적
키	따옴표 불필요	따옴표 불필요	따옴표 필수