

# .vscode

Tag

## 2 제일 효과 큰 설정: 파일 감시 제외

📌 이거 안 하면 다른 튜닝 다 의미 없음

.vscode/settings.json

```
{  
  "files.watcherExclude":{  
    "**/node_modules/**":true,  
    "**/.next/**":true,  
    "**/dist/**":true,  
    "**/build/**":true,  
    "**/venv/**":true,  
    "**/.venv/**":true,  
    "**/__pycache__/**":true,  
    "**/.mypy_cache/**":true  
  }  
}
```

👉 tsserver / pylance가

\*\*"이 폴더들 절대 보지 마"\*\*라고 명시하는 것

## 3 TypeScript 서버 범위 축소 (Next.js 필수)

frontend 전용 tsconfig 분리

frontend/tsconfig.json 안에 👉

```
{  
  "compilerOptions":{  
    "skipLibCheck":true,  
    "incremental":true  
  },  
  "exclude": [  
    "node_modules",  
    ".next",  
    "dist"  
  ]  
}
```

그리고 repo 루트에 tsconfig 두지 마

(있다면 삭제하거나 최소화)

👉 루트 tsconfig 있으면

Python 코드까지 TS가 훔음

## 4 Python 쪽: 분석 범위 명확히 제한

.vscode/settings.json에 추가

```
{  
  "python.analysis.autoSearchPaths": false,  
  "python.analysis.useLibraryCodeForTypes": false,  
  "python.analysis.exclude": [  
    "**/node_modules",  
    "**/.next",  
    "**/dist",  
    "**/build",  
    "**/frontend/**"  
  ]  
}
```

👉 Pylance가 frontend 쪽을 완전히 무시

## 5 monorepo에서 가장 좋은 구조 (권장)

📌 Workspace 분리 (진짜 체감 큼)

repo.code-workspace 파일 만들어서:

```
{  
  "folders": [  
    {  
      "path": "backend"  
    },  
    {  
      "path": "frontend"  
    }  
  ],  
  "settings": {  
    "files.watcherExclude": {  
      "**/node_modules/**": true,  
      "**/.next/**": true,  
      "**/venv/**": true  
    }  
  }  
}
```

그리고 이 workspace 파일로 VS Code 열어

👉 그러면:

- Python LS ↔ TS LS 완전히 분리
- hover latency 눈에 띄게 감소

## 6 Hover 느릴 때 추가로 끄면 좋은 것들

### TypeScript 쪽

```
{  
  "typescript.disableAutomaticTypeAcquisition":true,  
  "typescript.tsserver.maxTsServerMemory":4096  
}
```

### VS Code 자체 튜닝

```
{  
  "editor.hover.delay":300,  
  "editor.hover.enabled":true,  
  "editor.parameterHints.enabled":false  
}
```

(parameter hint는 생각보다 비쌈)