

# Haskell with VS Code in MacOS

# Agenda

**1. Haskell Download**

**2. The Haskell Tool Stack**

**3. Editor to IDE**

# 1. Haskell Download

[하스켈 홈페이지](#)에 보면 하스켈을 다운로드하는 방법이 총 3가지가 있다.

1. [Minimal installers](#)
2. [Stack](#)
3. [Haskell Platform](#)

이 중, `Stack` 으로 설치하려고 한다.

[하스켈 학습 가이드](#), [Haskell 스터디 Part.01](#)를 참고하여 Stack으로 결정

## 2. The Haskell Tool Stack

Stack이란 하스켈의 크로스 플랫폼 빌드 도구이다. Stack을 통해서 하스켈을 비교적 쉽게 시작할 수 있다.

## 2-1. Stack Download

[홈페이지](#)에서 소개하는 몇 가지 방법 중, `homebrew` 를 통해 설치한다.

터미널에 다음과 같이 입력한다.

```
$ brew install haskell-stack
```

## 2-2. Create Project

이제 프로젝트를 생성해보자.

```
$ stack new my-project && cd my-project
```

`my-project` 라는 디렉토리 안에 프로젝트의 뼈대가 구성된다.

## 2-3. Compiler Download

```
$ stack setup
```

`stack setup` 은 컴파일러를 **격리된 위치**로 다운로드하여, 시스템의 다른 Haskell 의 설치에 대해 영향을 주지 않는다.

`stack path` 명령어를 통해 설치 경로를 볼 수 있다.

### 3. VS Code를 Editor에서 IDE로 바꾸기



VS Code를 켜서 확인해보자.

```
$ code .
```

```
1  module Main where
2
3  import Lib
4
5  main :: IO ()
6  main = someFunc
7  |
```

불-편

## 3-1. Syntax Highlighting

Haskell Syntax Highlighting Extension을 다운로드한다.

```
1  module Main where
2
3  import Lib
4
5  main :: IO ()
6  main = someFunc
7
```

편-안

## 3-2. Linting

`haskell-linter` Extension을 다운로드한다.

- 다운로드를 하면 이와 같은 메시지를 보게 될 것이다.

**Error** Cannot lint the haskell file. The lint program was not found. Use the 'haskell.lint.executablePath' :

- 추가로 다운로드가 필요하다.

```
$ stack install hlint
```

- `export PATH=$PATH:~/.local/bin` 을 각자 환경에 맞게 `.bash_profile` 이나 `.bashrc` , `.zshrc` 에 추가해준다.
- VS Code를 종료하고 다시 키면 lint가 작동하는 것을 확인할 수 있다.

### 3-3. Language Server

**Haskell Language Server**은 다양한 기능을 갖고 있다.

- 모던한 개발 환경
- 타입 정보
- 함수 정의
- jump to definition
- 대소문자 구분

- 하지만 마켓에서 제공하는 것은 **클라이언트** 뿐이다.
- 즉, 서버와 이어주는 역할만 해주는 것이기 때문에 **서버를 따로 설치**해야 한다.
- Language Server인 [haskell-ide-engine](#)을 설치하도록 한다.



## haskell-ide-engine

하스켈의 Language Server 중 가장 큰 커뮤니티와 높은 완성도를 갖고 있다.

먼저 컴파일러의 버전을 확인한다.

```
$ stack ghc -- --version
```

```
The Glorious Glasgow Haskell Compilation System,  
version 8.6.5 (2019.09.04 기준)
```

확인했으니 설치한다.

```
$ cd ..
```

```
$ git clone https://github.com/haskell/haskell-ide-engine  
--recurse-submodules && cd haskell-ide-engine
```

- 8.6.5 버전의 컴파일러에 맞는 haskell ide engine 버전을 설치해야 한다.
- 그런데 [haskell-ide-engine 레파지토리](#)를 확인해보면 8.6.4 버전의 컴파일러에 대한 `hie` 가 최신인 것을 알 수 있다.
- 고로 8.6.4 버전을 설치한다.

정확하게 확인은 못했지만 `hie-8.6.5` 를 설치해도 `hie-8.6.4` 가 설치되는 것으로 보인다.

정말 오래 걸린다.

```
$ stack ./install.hs hie-8.6.4  
$ stack ./install.hs build-data
```

- 설치가 완료되면 Haskell Language Server Extension을 설치해준다.

```
1 module Main where
```

```
2
```

```
3 import Lib
```

```
4 
```

```
5 main :: IO ()
```

```
6 main = s
```

```
7
```

```
8
```

```
9 square x
```

```
10 main1 =
```

```
11
```

```
12 quicksort
```

```
13 quicksort
```

```
14 where
```

```
15
```

```
16
```

```
17
```

```
18
```

A value of type `IO a` is a computation which, when performed, does some I/O before returning a value of type `a`.

There is really only one way to "perform" an I/O action: bind it to `Main.main` in your program. When your program is run, the I/O will be performed. It isn't possible to perform I/O from an arbitrary function, unless that function is itself in the `IO` monad and called at some point, directly or indirectly, from `Main.main`.

`IO` is a monad, so `IO` actions can be combined using either the `do`-notation or the `>>` and `>>=` operations from the `Monad` class.

[Documentation](#)

Language Server 가 제공하는 정보 중, IO 모나드에 대한 정보가 나오는 것을 확인할 수 있다.

# Reference

[Setting up Haskell in VS Code on macOS](#)

[haskell-ide-engine](#)

[Visual studio code 에서 haskell 개발환경 세팅하기 \(Haskell-Ide-Engine\)  
\(os x\)](#)

[Haskell 스터디 Part.01](#)

[하스켈 학습 가이드](#)

[하스켈 홈페이지 다운로드](#)

[Stack Documentation](#)

[LSP\(Language Server Protocol\) 간단 설명](#)