

Arbeitsblatt: Haskell Scripts

Wenn Sie GHCi beenden, gehen alle Eingaben verloren. Um das zu vermeiden, erstellen Sie Haskell Scripts. Ein Haskell Script ist ein Textdokument, das eine Abfolge von Definitionen enthält. Sie werden jetzt ein Haskell Script erstellen und die darin definierten Funktionen aus GHCi aufrufen.

- a. Auf dem AD finden Sie das File 01_Intro/Lecture/first.hs mit folgendem Inhalt:

```
-- Das erste Haskell Script

size :: Integer
size = (4 - 3) * 2

-- Gibt den doppelten Wert zurück
times2 :: Integer -> Integer
times2 x = 2 * x           -- (t2)

-- Quadriert den Wert
square :: Integer -> Integer
square x = x * x           -- (sqr)

-- Pythagoras a^2 + b^2 = c^2
pyth :: Integer -> Integer -> Integer
pyth a b = square a + square b -- (py)
```

- b. Kopieren Sie dieses File lokal auf Ihren Computer. Merken Sie sich das Verzeichnis.
c. Laden Sie das Script mit GHCi:
Wechseln Sie zuerst in das Verzeichnis wo das Script liegt
> cd /path/to/script/folder
Dann starten Sie ghci mit dem Script Name als Argument:
> ghci first.hs

Die Standard Bibliothek und Ihre eigenen drei Funktionen wurden geladen und können verwendet werden.

Überlegen Sie sich die Resultate der folgenden Ausdrücke zuerst im Kopf und überprüfen Sie dann das Resultat mit GHCi:

Ausdruck	Resultat (Kopf)	Resultat (GHCi)
times2 (times2 4)		
square (times2 3)		
pyth 1 2		

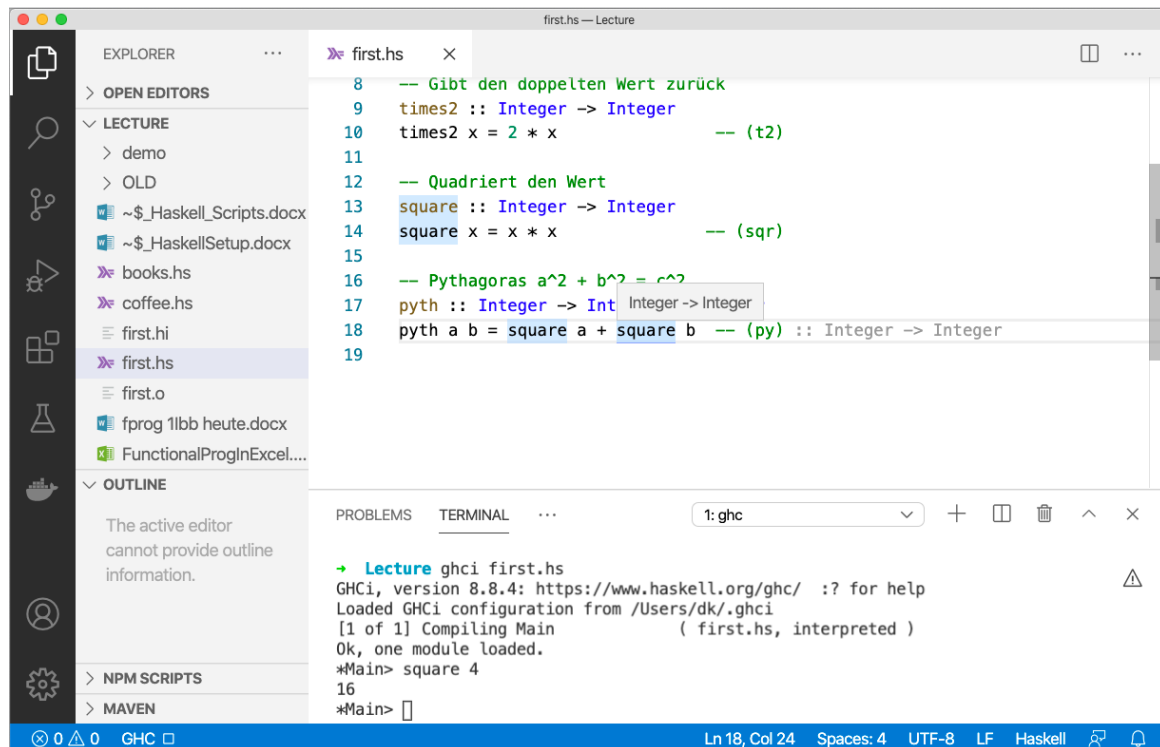
Die wichtigsten GHCi Kommandos

GHCi Kommandos beginnen alle mit einem Doppelpunkt ':':

Kommando	Beschreibung
:?	Zeigt den Hilfetext an.
:cd /path/to	Wechselt das Arbeitsverzeichnis auf /path/to
:l script.hs	Lädt das Script namens Script.hs.
:r	Lädt das aktuelle Script neu.
:t expr	Zeigt den Typ des Ausdrucks expr.
:q	Beendet die Sitzung.

Haskell Editor Setup

In dieser Anleitung ist das Setup einer leichtgewichtigen Haskell Entwicklungsumgebung beschrieben.



Voraussetzung: GHC ist bereits installiert!

1. Installieren Sie Visual Studio Code <https://code.visualstudio.com/>

2. Haskell Extensions

Installieren Sie die folgenden zwei Extensions unter

OSX: Code > Preferences > Extensions

WIN: File > Preferences > Extensions

Haskell Syntax Highlighting

<https://marketplace.visualstudio.com/items?itemName=justusadam.language-haskell>

Simple GHC (Haskell) Integration

<https://marketplace.visualstudio.com/items?itemName=dramforever.vscode-ghc-simple>

Öffnen Sie im VS Code integrierten Terminal eine GHCi Session, um mit dem Code zu experimentieren.

Viel Vergnügen!