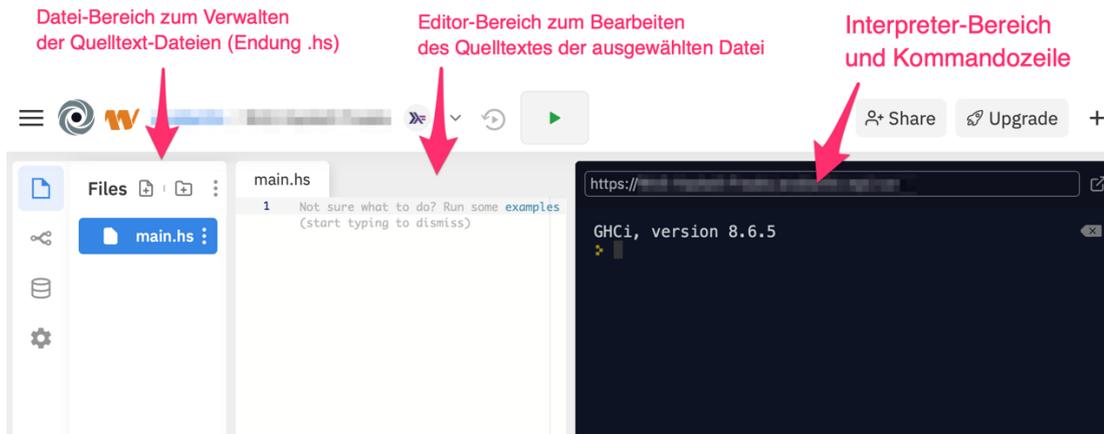


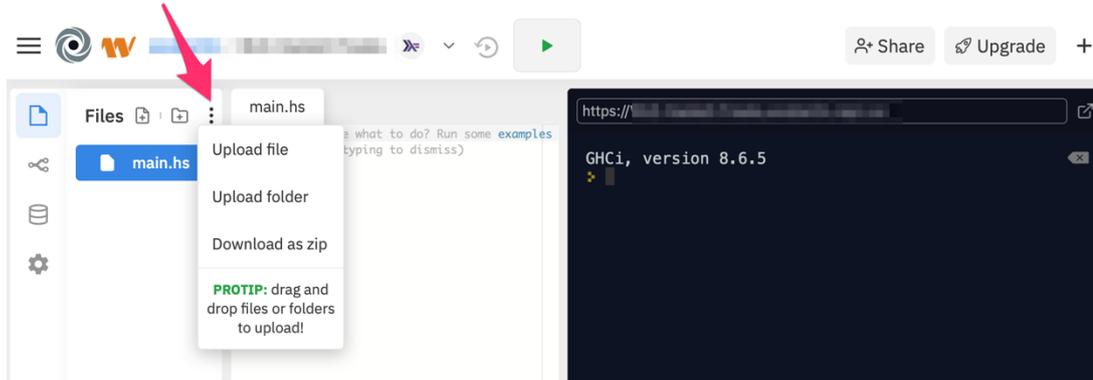
So sieht der Start-Oberfläche aus, nachdem man sich auf repl.it eingeloggt hat. Hinweis: das Bearbeiten von Dateien funktioniert auf jeden Fall, wenn man sich (kostenlos) angemeldet hat. Der beste Weg ist sicherlich einen eigenen Account mit eigener E-Mail zu **registrieren**.

Die Datei main.hs wird automatisch erstellt und ist leer. Diese lässt sich nicht löschen und bleibt so.

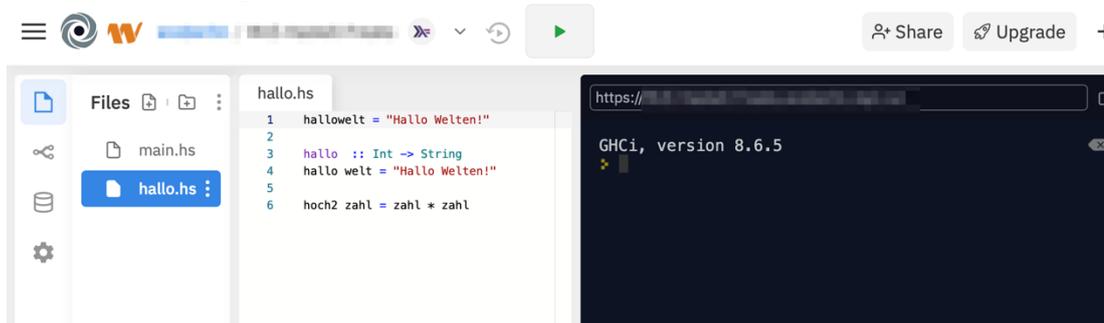


Jetzt kann eine eigene Datei hinzugefügt werden: entweder direkt über Upload oder Drag And Drop. Bei der Datei auf die Dateierdung .hs achten.

Auswahl durch Klick auf die drei Punkte

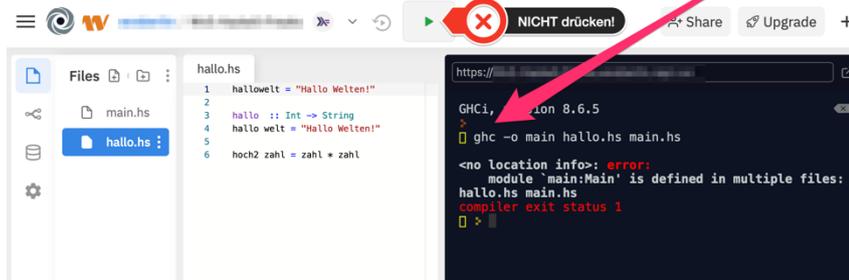


Nachdem die eigene Datei hinzugefügt wurde, steht sie jetzt im System und im Editor zur Verfügung. Die Funktionen sind allerdings noch nicht geladen!



Jetzt wird man verführt auf den Run-Button zu klicken. Prompt kommt eine **Fehlermeldung**. Hier geht es so nicht weiter, da wir nicht den Compiler aufrufen wollen, sondern im **Interpreter-Modus** arbeiten.

Fehlermeldung, wenn man auf den RUN-Button drückt. So geht es nicht weiter, da hier der Compiler aufgerufen wird. Wir wollen jedoch im Interpreter-Modus arbeiten.



NEU! Offenbar wird der Interpreter beim Starten von Haskell bei repl.it nicht mehr automatisch gestartet. In diesem Fall müssen wir deshalb in der **Console** den Befehl **ghci** eingeben, um den Haskell-Interpreter (bzw. den interaktiven Modus) zu starten. Es sollte der **Prelude>** Prompt erscheinen:

```
hallo.hs
1 hallowelt = "Hallo Welten!"
2
3 hallo :: Int -> String
4 hallo welt = "Hallo Welten!"
5
6 hoch2 zahl = zahl * zahl

Console
> ghci
GHCi, version 8.6.5: http://www.haskell.org/ghc/ :? for help
Prelude>
```

Um im **Interpreter-Modus** zu arbeiten, geben wir am **Prompt** die aus **WinHugs** bekannten **Interpreter-Befehle** ein. Eine Liste aller **Interpreter-Befehle** lässt sich mit **:?** Anzeigen.

```
:l Dateiname.hs
```

hier:

```
:l hallo.hs
```

```
hallo.hs
1 hallowelt = "Hallo Welten!"
2
3 hallo :: Int -> String
4 hallo welt = "Hallo Welten!"
5
6 hoch2 zahl = zahl * zahl

Terminal
https://
GHCi, version 8.6.5
> :l hallo.hs
[1 of 1] Compiling Main interpreted)
Ok, one module loaded.
```

Jetzt können die definierten Funktionen wie in WinHugs am Interpreter-Prompt aufgerufen werden. Änderungen werden wie bekannt mit

```
:r
```

eingelezen. Die Pfeiltasten hoch und runter zeigen eingegebene Eingaben am Prompt an.

```
hallo.hs
1 hallowelt = "Hallo Welten!"
2
3 hallo :: Int -> String
4 hallo welt = "Hallo Welten!"
5
6 hoch2 zahl = zahl * zahl

Terminal
https://
GHCi, version 8.6.5
> :l hallo.hs
[1 of 1] Compiling Main interpreted)
Ok, one module loaded.
> :r
> hoch2 34
=> 1156
> hallowelt
=> "Hallo Welten!"
> hallo 42
=> "Hallo Welten!"
> hallo "Hallo"
=> "Hallo"
<interactive>:21:7: error:
 * Couldn't match expected type `Int' with actual
type `[Char]'
 * In the first argument of `hallo', namely
`"Hallo"'
   In the expression: hallo "Hallo"
   In an equation for `it': it = hallo "Hallo"
```

Hinweise zum Speichern auf der folgenden Seite!

Das **Speichern** der Datei erfolgt leider nur gepackt als ZIP-Datei. Dies bitte beim Weiterarbeiten in WinHugs (präferiert) und Abgeben beachten. Alternativ lässt sich der Inhalt des Editors auch mit dem eigenen Rechner (eigener Editor) durch **Kopieren** (und Einfügen) austauschen und sichern.

