

Risa_Enhancers の使い方

更新日：2025 年 06 月 24 日

1 基本的な使い方

この拡張機能は VSCode を用いて Risa/Asir のコードを書くこと、および計算を実行することを簡単に
にする目的で作られています。

1.1 事前準備

拡張機能を利用するためには、VSCode と Risa/Asir がインストールされていることが前提となります。
それぞれのインストール方法については割愛します。

まずはこちらの Github のページ https://github.com/kanjeve/Risa_Enhancers のページから最新版
の `risa-enhancers-0.3.0.vsix` をダウンロードします。次に VSCode を開き、左側にあるアクティビ
ティバーから「拡張機能」（正方形が4つのアイコン）を開きます。拡張機能のページの右上にある3つ
の点が横に並んだアイコン（ミートボールメニューと呼ばれるらしい）をクリックし、「VSIX からイン
ストール」を選択します。すると、エクスプローラーが開かれるので、先ほどダウンロードした `risa-
enhancers-0.3.0.vsix` を探し、開きます。これで、Risa_Enhancers が VSCode にインストールされまし
た。

この拡張機能はファイルの拡張子が `.rr` のものに対して、働きます。そのため、この拡張機能を利用
して Risa/Asir のコードを書く際には、ファイル名を `~.rr` という風にしてください。

以下で、「書くこと」、「計算すること」に関する基本的な使い方を紹介します。

1.2 「書くこと」

この拡張機能では、Asir 言語のシンタックスハイライト、コード補完（スニペット補完）とコード診
断を提供しています。

シンタックスハイライトとは、字句や構文の種類ごとに色付けをする機能です。例えば、 \LaTeX の拡
張機能を入れて利用すると、`\usepackage` などは青色に、数字は緑色になっているのが分かります。そ
れが、シンタックスハイライトです。この拡張機能でどのように色付けされるかは後述します。

コード補完・スニペット補完とは、簡単に言うと、コードの予測変換です。例えば、図 1 のように、“a”
と打つと、“a”から始まるコマンドやキーワードが表示されます（“a”が途中で来るものも表示される）。

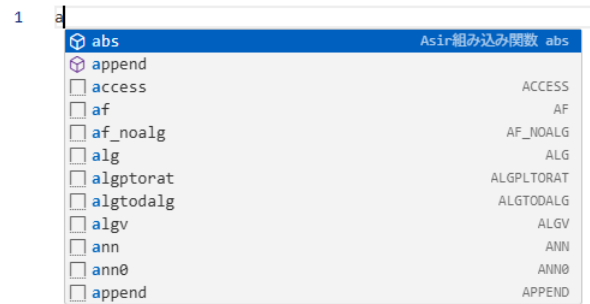


図 1: コード補完の様子

今回の例では、そのまま Enter キーを押すと、abs が入力されます。詳細な補完については後述します。

コード診断とは、括弧が閉じられていない時などに、間違っていると考えられる箇所に波線が引かれ、警告文を表示する機能となっています。詳細は後述しますが、**このコード診断は完全なものではありません**。参考程度にとどめてください。

1.3 「計算すること」

コードを書き上げた後に、ファイルを保存したり、Risa/Asir を起動したりせずに、計算を実行することができます。

ファイルを選択した状態で、Shift キーと Enter キーを同時押しすると、コードが Risa/Asir に送られ、計算結果が下の出力タブに、入出力が右側に Webview で表示されます（図 2 参照）。

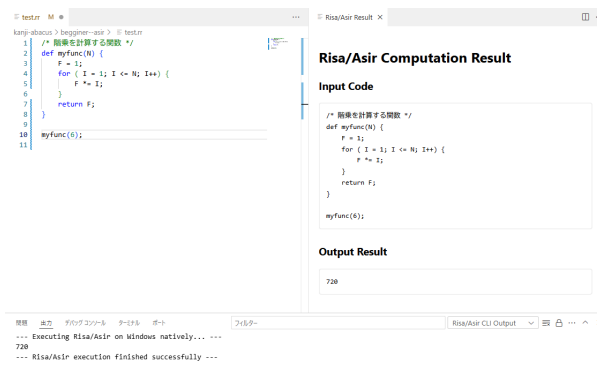


図 2: コード実行の様子

また、実行したい部分だけを範囲選択して Shift + Enter を押すことで、部分実行することもできます（図 3 参照）。

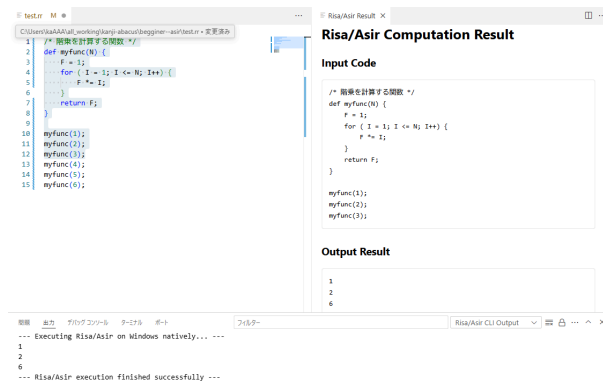


図 3: コード部分実行の様子

コード実行中に計算時間が非常に長くなってしまった場合に、右上にあるコーヒーカップのアイコン (図 4) をクリックすることで、計算をキャンセルすることができます。



図 4: キャンセルコマンドのアイコン

余談ですが、このアイコンは Team SNAC Tsukuba の初期ロゴデザイン中央上部にあるコーヒーを模倣しています。

コードの実行にはインタラクティブセッションを利用することもできます。製作者はデバッグで使われることを想定しているので、以下ではデバッグ実行と呼びます。

使い方は、まず書き上げたコードのうち、デバッグしたい関数などを範囲選択し、Ctrl キー (cmd キー) と Shift キーと D を同時に押すことで、デバッグ実行できます。実行すると、下部に“Risa/Asir Interactive”という名前のターミナルが立ち上がります。すると、数秒後に Risa/Asir が起動し、`load("~.rr");` コマンドが自動で送られます (※注意: load コマンドが送られるタイミングは、起動コマンドを送ってから 3 秒後です。もし、自分で使ってみて中々読み込まれないと感じた場合、この時

間の設定を変えることで、読み込まれるようになるかと思います。詳細は後述します)。この時に読み込まれる `~.rr` は、範囲選択したコードのみが記載されたファイルになっています。

インタラクティブセッションが起動している間は、範囲選択して `Shift + Enter` を押すことで、コードを実行することができます。また、ターミナルに直接書き込むことで実行することもできます。これによって、ターミナルに `Ctrl + C` と `d` を送ることでデバッグモードに入ることができます（※注意：Windows 版は非対応です。WSL を使って利用することはできます）。前述した通常のコード実行ではできなかったデバッグを使うことができる点で、このセッションをデバッグ用と位置付けています。

インタラクティブセッションは、右下にある“Stop Risa/Asir Debug Session”をクリックすることで、停止します。基本的に、セッション利用後はこのコマンドをクリックしてセッションを終了してください。

Windows 向けの機能ですが、コードを実行する際に Windows 上の `asir.exe` を起動するか、WSL 上の `asir` を起動するかを選ぶコマンドがあります。左下の円形の矢印をクリックすることで変更でき、“Risa/Asir: Windows”となっているときは、コードが Windows 上で実行され、“Risa/Asir: WSL”となっているときは、コードが WSL 上で実行されます。（※注意： このコマンドを利用するためには 1.WSL がインストールされていること、2.WSL 上に Risa/Asir がインストールされていること、3.VSCode の拡張機能 Remote Development がインストールされていることが条件です。）

2 仕様と注意

この章では各コマンドの詳細な仕様とその注意を説明します。「基本的な使い方」の章では分からないことがこの章に書かれている可能性はあります。

2.1 「書くこと」の仕様

シンタックスハイライト、コード補完、コード診断の詳細な仕様を説明します。

2.1.1 シンタックスハイライト

2.1.2 コード補完

2.1.3 コード診断

2.1.4 「書くこと」に関する注意

2.2 「計算すること」の仕様

コード実行の諸コマンドの詳細な仕様について説明します。

2.2.1 executeCode

2.2.2 cancelExecution

2.2.3 startAsirInteractive

2.2.4 stopAsirInteractive

2.2.5 switchExecutionMode

2.2.6 設定と注意