



デカルト言語の操作方法

1. コマンド引数指定

コマンドプロンプトから起動するときのオプションについて 以下に示します。

```
usage: descartes [OPTION...] [FILE]

-?, --help      Show this help message
-v, --version   print the version of descartes being used

-u, --utf8      UTF8 code is used
-e, --euc       EUC code is used
-s, --sjis      SJIS code is used
```

これは、--helpオプションを付けて起動すると表示される ものです。

1.1 コマンドにプログラムファイルを指定

プログラムファイルを実行するときには、コマンドプロンプトで、`descartes`にプログラムファイルを指定します。(このとき、`descartes`はお使いのOSの実行パス上にあることを 確認しておいてください。)

以下の内容のファイルを`hello.car`という名前で保存してください。

```
? <print "hello, world!!">;
```

以下はLinux上の実行例です。

```
$ descartes hello.car
hello, world!!
result --
      <print hello, world!!>
-- true
```

`hello, world!!` が表示されているはずです。

2. 会話処理の操作

会話処理を実行するときには、コマンドプロンプトで、プログラムファイルを指定せずに起動します。

以下のように表示され、?プロンプトが表示され、コマンド入力を待ちます。

```
$ descartes
Descartes 0.10.0          (code:SJIS)
Copyright (C) 2009 Hideyuki Niwa

?
```

終了するには、`quit`と入力します。

```
? quit
```

2.1. プログラムの実行

?プロンプトの後に、実行するプログラムを入力します。

例えば前の項の実行例で示した、"hello, world!!"を表示するプログラムを実行するには以下のように入力します。

```
? <print "hello, world!!">;
hello, world!!
result --
      <print hello, world!!>
-- true
```

入力されたプログラムは、すぐに実行されます。

また、複数の節(clause)のプログラムも入力できます。

```
? <print abc><print def>;
abc
def
result --
      <print abc>
      <print def>
-- true
```

さらに、会話処理の場合限定なのですが、単一の節の場合には、一番外側の<>括弧と;セミコロンを省略できます。

```
? print abc
abc
result --
      <print abc>
-- true
```

モジュールの呼び出しでも同様です。以下の処理はどちらも同じ結果になります。

```
? ::sys <cos #x 0>;
result --
      ::sys <cos 1 0>
-- true
? ::sys cos #x 0
result --
      ::sys <cos 1 0>
-- true
```

2.2. プログラムの入力

プログラムを実行するのではなく、入力して登録するには、?プロンプトの後に/を入れた後にプログラムを入力します。

例えば、以下のプログラムを入力することとします。

```
<ex> <print example>;
```

このように/を打った後に入力します。

```
? /<ex> <print example>;
```

入力されたプログラムを確認するには、`list`述語を使います。

```
? <list>;
<ex>
    <print example>
    ;
result --
    <list>
-- true
?
```

単に`list`と入力するだけでもOKです。

```
? list
<ex>
    <print example>
    ;
result --
    <list>
-- true
```

入力されたプログラムを実行してみます。

```
? <ex>;
example
result --
    <ex>
-- true
? ex
example
result --
    <ex>
-- true
```

2.3. プログラムのロード

プログラムを、ファイルからロードして登録するためには、`load`述語を使います。

以下の内容のファイルを`hello2.car`という名前で保存しておいてください。

```
<exload>    <print "hello, world!!">;
```

`load`述語の引数に入力ファイルを指定して、実行します。

```
? load "hello2.car"
? list
<exload>
    <print hello, world!!>
    ;
result --
    <list>
-- true
? exload
hello, world!!
result --
    <exload>
-- true
```

プログラムのロードは、すでに入っているプログラムをクリアしません。同じプログラムをロードすると重複して、2重に同じプログラムが登録されてしまいます。

登録されたプログラムをクリアするためには、`new`述語を使ってクリアしてください。

```
? load "hello2.car"
? list
<exload>
    <print hello, world!!>
    ;
result --
    <list>
-- true
? load "hello2.car"
? list
<exload>
    <print hello, world!!>
    ;
<exload>
    <print hello, world!!>
    ;
result --
    <list>
-- true
? new
result --
    <new>
-- true
? list
result --
    <list>
-- true
```

2.4. プログラムのinclude

ライブラリをインクルードするためには、`include`述語を使います。

```
? include list
?
```

インクルードされるプログラムは、環境変数DLIBPATHのパス上からサーチされます。お使いのOSの環境変数の設定方法でパスを設定してください。

環境変数DLIBPATHの値は、`::sys<DLIBPATH #VAR>`述語で確認できます。

```
? ::sys DLIBPATH #VAR
result --
    ::sys <DLIBPATH (.)>
-- true
```

2.5. プログラムの編集

デカルト言語内からプログラムファイルを編集するために、`edit`述語を使います。

```
edit ファイル名
```

起動するエディタはデフォルトでは、Linux版ではvi, Windows版ではノートパッドになっています。

デフォルトと異なるエディタを使いたい場合は、環境変数DEDITORPATHに起動するエディタのパスを設定してください。

`edit`述語では、プログラムファイルの文字の編集しか行いません。編集したプログラムをロードする場合には、`load`述



語を使ってロードしてください。

2.6. プログラムの保存

デカルト言語内からプログラムファイルを保存するために、**save**述語を使います。

```
save ファイル名
```

登録されていたプログラムが、ファイルに保存されます。

【ページ情報】 更新日時: 2009-04-29 13:40:39, 更新者:  hniwa
【ライセンス】  クリエイティブ・コモンズ 表示
【権限】 表示:制限なし, 編集:ログインユーザ, 削除/設定:ログインユーザ
【IRI】 <http://sourceforge.jp/projects/descartes/wiki/ManOperation>