マニュアルに戻る。

ルート

- ドリトルのすべてのオブジェクトの親となるオブジェクトです。
- ルートオブジェクトのプロパティは、プログラム全体から参照される変数です。
- ルートオブジェクトのプロパティに値を書くと、他のオブジェクトからその値を参照できます。
- ルートオブジェクトのプロパティは、オブジェクトを明示した「ルート：」、またはオブジェクトを省略した「：」で指定します。
    - （例）ルートオブジェクトに「歩幅」というプロパティを設定します。

```
ルート：歩幅＝30。
```

- （例）ルートオブジェクトに「歩幅」というプロパティを設定します。

```
：歩幅＝30。
```

未定義

- 予期しない実行が行われた場合にシステムから返されるオブジェクトです。
- 存在しない変数を参照したときは、未定義オブジェクトが返ります。
- 命令が正しくない結果になった場合にも、未定義オブジェクトが返されることがあります。
- 値が未定義オブジェクトかどうかは、**未定義** または**undef** と比較することで判別できます。

システム

- ドリトルや動かしているコンピュータの情報を利用するためのオブジェクトです。
- システムの情報を得るためのプロパティが用意されています(Java版のみ)
    - （例）ドリトルのバージョンを表示します。

```
ラベル！（システム□versionstr□作る。
```

プロパティ	値	プロパティ	値
versionstr	バージョン文字列	**hostname**	ホスト名
version	バージョン番号	**displayWidth**	ディスプレイの幅（※2）
javavendor	Javaのベンダー	**displayHeight**	ディスプレイの高さ（※2）
javaversion	Javaのバージョン	**memory**	メモリ使用量
osname	OS名	**user**	ユーザー名（※3）
osversion	OSバージョン	**lang**	言語（※3）
ipaddress	IPアドレス（※1）		

- （補足）
    - □※1□ipaddressは、コンピュータが複数のIPアドレスを持つ場合は任意の1個を返します□IPアドレスは、サーバーウィンドウにも表示されます。
    - □※2□displayWidthとdisplayHeightはディスプレイのサイズを返します。ウィンドウのサイズは**obj□screen** の画面オブジェクトで取得してください。
    - □※3□userとlangはローカル版で利用可能です。

- **日時？**：日時を「Thu Feb 05 18:35:05 JST 2009□の形式で返します。

    - （例）日時を表示します。

ラベル！（システム！日時？）作る。

- **曜日？**: 曜日を「木」の形式で返します。
    - （例）曜日を表示します。

ラベル！（システム！曜日？）作る。

- **年？**: 年を西暦で返します。
    - （例）年を表示します。

ラベル！（システム！年？）作る。

- **月？**: 月を数字で返します。
    - （例）月を表示します。

ラベル！（システム！月？）作る。

- **日？**: 日を数字で返します。
    - （例）日を表示します。

ラベル！（システム！日？）作る。

- **時刻？**: 時刻を「18:35:05」の形式で返します。
    - （例）時刻を表示します。

ラベル！（システム！時刻？）作る。

- **時？**: 時間を数字で返します。
    - （例）時間を表示します。

ラベル！（システム！時？）作る。

- **分？**: 分を数字で返します。
    - （例）分を表示します。

ラベル！（システム！分？）作る。

- **秒？**: 秒を数字で返します。
    - （例）秒を表示します。

ラベル！（システム！秒？）作る。

- **システム秒？**: システム秒を返します。2つの値の差分を取ることでミリ秒単位の経過時間を計算できます。
    - （例）掛け算を10万回計算する時間を表示します。

```
start□システム！システム秒？。
□□□□□□□□□□□□□□□□繰り返す。
end□システム！システム秒？。
ラベル□□end - start□作る。
```

- 以下はJava版のみ
    - **表示ダイアログ**: ダイアログを表示します。

    - （例）「こんにちは」と表示します。

```
システム！"こんにちは"　表示ダイアログ。
```

- **確認ダイアログ**: 確認するダイアログを表示します。
- （例）質問を表示し、「はい/いいえ」の回答に合わせて文を表示します。

```
回答＝システム！"今日は晴れですか？"　確認ダイアログ。
「回答」！なら「文＝"天気です"」そうでなければ「文="くもりです"」実行。
システム！（文）　表示ダイアログ。
```

- **入力ダイアログ**: 入力ダイアログを表示します。
- （例）入力ダイアログを表示し、入力された文を表示します。

```
入力=システム！"今日の天気は？"　入力ダイアログ。
システム！（入力）　表示ダイアログ。
```

- **選択ダイアログ**: 選択ダイアログを表示します。
- （例）今日の天気を選択し、回答を表示します。

```
候補＝配列！"晴れ"　"曇り"　"雨"　"雪"　作る。
回答=システム！"今日の天気は？"　（候補）　選択ダイアログ。
システム！（回答）　表示ダイアログ。
```

- **使う**:初期化ファイルを実行します。プログラムごとの初期化ファイルを指定するときに使います。指定した名前に “.ini” の拡張子を付けたファイルが読み込まれ、実行されます。ただし、**startup.ini**という名前のファイルが存在する場合は、この命令に関わらず、自動的に読み込まれて実行されます。
- （例）abc.iniという初期化ファイルを実行します。

```
システム｜"abc"｜使う。
```

- **実行**: 外部コマンドを実行します。ローカル版で利用可能です。戻り値は外部コマンドの標準出力の文字列です。複数行の場合は改行記号で連結されます。戻り値のプロパティ｜stderr｜は外部コマンドのエラー出力です。戻り値のプロパティ｜retcode｜は外部コマンドのリターンコードです。
- （例）OSの「date｜コマンドの結果を表示します。

```
結果＝システム｜"date"｜実行。
ラベル！（結果）作る。
ラベル！（結果:stderr｜作る。
ラベル！（結果:retcode｜作る。
```

- **proxy** : HTTP通信を行う際のProxyを設定します。ポートを省略した場合は8080が使われます。
- （例）Proxyを “ proxy.eplang.jp”に設定します

```
システム｜"proxy.eplang.jp"｜proxy｜
```

- （例）Proxyを “ proxy.eplang.jp”の8080に設定します。

```
システム｜"proxy.eplang.jp"｜8080｜proxy｜
```

- **サーバーポート** : サーバーオブジェクトと通信するポート番号を指定します。標準では2020が使われます。通常の利用では、ポート番号を変更する必要はありません。
- （例）通信ポートを2000に変更します。

システム！2000　サーバーポート。

- **終了する**：ドリトルを終了します。パラメータに真（はい）を指定すると、確認せずにドリトルを終了します。
- （例）ドリトルを終了します。

システム！終了する。

- （例）ドリトルを終了します。確認のダイアログは表示されません。

システム！（はい） 終了する。

- **再実行**：プログラムを再実行します。実行ボタンを押した際と同等の動作を行います(V3.11以降)

【注意】本機能は非推奨機能です。通常は画面下部の実行ボタンを使用してください。

- （例）プログラムを再実行します。

システム！再実行。

From:
https://dolittle.eplang.jp/ - **プログラミング言語「ドリトル」**

Permanent link:
https://dolittle.eplang.jp/ref_system?rev=1515028337

Last update: **2018/01/04 10:12**