

オブジェクトファイル

- オブジェクトをファイルに保存しておくためのオブジェクトです。
- 保存したオブジェクトは、再び読み込んで使うことができます。
- 数値、文字列、配列のオブジェクトに対応しています。
- オンライン版では利用できません。
- **作る**: オブジェクトファイルを作ります。パラメータとしてファイル名を指定します。
 - (例 `record file1.txt` というファイルを使う、「記録」というオブジェクトファイルを作ります。

記録 = オブジェクトファイル `record file1.txt` 作る。

- **書く**: 名前を付けてオブジェクトを書き込みます。この名前はファイルから読み出すときのキーワードになります。オブジェクトの名前と違って構いません。
 - (例) オブジェクトファイルに `record point` という名前ですべて「30」という値を保存します。

記録 = オブジェクトファイル `record file1.txt` 作る。

点数 = 30。

記録 `record point` (点数) 書く。

- **読む**: 名前を指定してオブジェクトを読み出します。保存してあったオブジェクトが返されます。
 - (例) オブジェクトファイルから `record point` という名前の値を取り出して表示します。

記録 = オブジェクトファイル `record file1.txt` 作る。

得点 = 記録 `record point` 読む。

ラベル! (得点) 作る。

- **消す**: 名前を指定してオブジェクトを消します。
 - (例) オブジェクトファイルから `record point` という名前の値を消します。

記録 = オブジェクトファイル `record file1.txt` 作る。

記録 `record point` 消す。

テキストファイル

- ファイルに文字列を読み書きするためのオブジェクトです。
- ファイルに書くときは、1行ずつ文字列を追加できます。配列を書き込む場合は、ファイルの内容が配列の内容で置き換わります。
- ファイルから読むときは、全体を配列に読み込みます。デフォルトの文字コードはOSごとに異なります(WindowsはShift-JIS、MacとLinuxはUTF-8)。
- ファイルの文字コードがデフォルトのものでない場合、「読む」では第2引数に文字コードを指定し、文字コードを変更できます。
- オンライン版では利用できません。
- **作る**: テキストファイルを作ります。パラメータとしてファイル名を指定します。
 - (例 `record text1.txt` というファイルを使う、「記録」というテキストファイルを作ります。

記録 = テキストファイル `record text1.txt` 作る。

- **書く**: 文字列を書き込みます。ファイルの末尾に文字列を1行追加します。
 - (例) テキストファイルに「 “ こんにちは。 ” 」を追加します。

記録 = テキストファイル "text1.txt" 作る。
記録! "こんにちは。" 書く。
ラベル! (記録! 読む) 作る。

- **全部書く**: 配列を書き込みます。ファイル全体を配列の中身で置き換えます。
 - (例) 文字列「“かめた”」、「“かめきち”」、「“カメ子”」が要素の配列を作り、テキストファイルをそれらで置き換えます。

記録 = テキストファイル "text1.txt" 作る。
全員 = 配列! "かめた" "かめきち" "カメ子" 作る。
記録! (全員) 全部書く。
ラベル! (記録! 読む) 作る。

- **読む**: 配列に読み出します。ファイル全体を読み、各行が要素になった配列が返されます。
 - (例) テキストファイルの全体を読み込んで表示します。

記録 = テキストファイル "text1.txt" 作る。
ラベル! (記録! 読む) 作る。

- **新規?**: ファイルが存在しないことを調べます。
 - (例) テキストファイルが存在しないかどうかを表示します。

結果 = テキストファイル "text1.txt" 新規?。
ラベル! (結果) 作る。

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_file33?rev=1578223248



Last update: **2020/01/05 20:20**