

多倍長整数

- 長い桁数の整数を扱うオブジェクトです。
- 文字列(または数値)に「大きい整数にする」を送り生成します。
- 数値(内部は倍精度実数)から生成する場合は、初期値が数値の有効数字に制限されます。そのため、10桁以上の場合は数値でなく文字列から生成してください。
 - (例) 文字列から多倍長整数を生成します。

`x["1000000"]` 大きい整数にする。

- 数値演算子「+,-,*,/,%」は、内部的にそれぞれ`add`, `sub`, `mul`, `div`, `mod`という命令に変換されて扱われます。論理演算子「==,!=,>,>=,<,<=」も、内部的にそれぞれ`eq`, `ne`, `gt`, `ge`, `lt`, `le`という命令に変換されて扱われます。
- +, -, *, /, %: 四則演算 * (例) 「3 * 40」を計算し「120」を表示します `x[3]` 大きい整数にする。ラベル `x[40]` 作る。 `*` 足す `add`, 引く `sub`, 掛ける `mul`, 割る `div`: 四則演算。命令として使います。 * (例) 「3 * 40」を計算し「120」を表示します `x[3]` 大きい整数にする。ラベル `x[40]` 掛ける) 作る。 `* %`: 余り * (例) 8を3で割った余りを計算し「2」を表示します `x[8]` 大きい整数にする。ラベル `x[3]` 作る。 `* 余り mod`: 余り。命令として使います。 * (例) 8を3で割った余りを計算し「2」を表示します `x[8]` 大きい整数にする。ラベル `x[3]` 余り) 作る。 `* ==, !=, >, >=, <, <=`: 比較演算。両辺が多倍長整数または多倍長整数に変換できる文字列の場合は、多倍長整数として比較されます。 * (例) 「4 > 3」を計算し「true」を表示します `x[4]` 大きい整数にする `y[3]` 大きい整数にする。ラベル `x` `>` `y` 作る。 `* pow`: べき乗。「2の3乗」は「`pow[2,3]`」ではなく「`2[3] pow`」と書くことに注意してください。 * (例) 「2^70」を計算します `x[2]` 大きい整数にする。ラベル `x[70] pow` 作る。 `* abs`: 絶対値 * (例) 「|-3|」を計算し「3」を表示します `x[3]` 大きい整数にする。ラベル `abs[x]` 作る。 `* 文字`: 指定された文字コードUTF-16の文字を返します。 * (例) 文字列「A」を表示します `x[0x41]` 大きい整数にする。ラベル `x` 作る。 `* 例) 文字列「あ」`を表示します `x[0x3042]` 大きい整数にする。ラベル `x` 作る。 `</code>`

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_bignum?rev=1518142301

Last update: 2018/02/09 11:11

