

01-1 (人間とは異なり) 「単純な “計算手順” の反復」が得意です。

例えば「64×144cm のパネルに隙間無く敷き詰められる最大の正方形タイルの一辺の長さ」を知るには、64 と 144 の最大公約数 (GCD) を求める必要があります。GCD を求める方法には算数で習う「すだれ算」、より効率的な「互除法」などもありますが、最も単純なのは小さい方の値をm、他をnとしてまずxにmを代入し「① mとnとをxで割る→②共に割り切れたら GCD=xで終了→③その他ならxにx-1を代入して①に戻る」を繰り返すことです。そして、人間には面倒なこの手順をコンピュータは高速で代行してくれます。

title (題目) は無くても OK。2行目は「実行の自動許可」のためのコメントでIE以外では不要。

01-2 「1つ作ってどこでも実行」(Write once, run everywhere.) なのが JavaScript。

上の「手順」を (PC・タブレット・スマホを問わず ネット経由でもローカルでも) ブラウザ上で動く JavaScript で記述した一例を資料1に示しました。

用語は基本的な英語で、変数名や関数名には日本語も使えるため「理解しやすいプログラム」が書けます。

01-3 必要なのは script (脚本)。

JavaScript の最小構成は手順を指定する脚本 (script ~ /script) で、今回は併せて入出力用の欄などの用紙 (form ~ /form) も使用します。

form (用紙) では、8桁の入力欄2つ (初期値として64と144を指定)、script 実行ボタン、そして8桁の出力欄を指定しています。なお、<>の外の「小さい数 = 」などは、HTML の記述としてそのまま提示されます。

script (脚本) には function (関数) が1つのみ設定され、実行ボタンのクリックでまず form の1・2番目 ([0] と [1]) の要素の値が変数「小」と「大」に代入 (=) され、続いてxを「小」の値 (x=小) から1までの範囲 (0<x) について順次1ずつ減らしながら (x--) 以下の処理が行われるよう記述がなされています※:

- ①小がxで割り切れない (小%x != 0) なら次のxに進む (continue) →
 - ②大がxで割り切れないなら次のxに進む →
 - ③その他 (小も大もxで割り切れた) ならその値を4番目の要素の値に代入して反復を終了する (break)。
- 資料1の内容の実行結果は資料2のとおりです。
※ “大小” 誤入力対策 (例: if(大<小){仮=大;大=小;小=仮}) は割愛。

01-4 「互除法」で処理アルゴリズムの改善が可能。

資料3はユークリッドの互除法 (大の数が小の数で割り切れるまで「大の数への小の数の代入」と「小の数への余りの代入」を反復) を用いた改善例です (資料1との本質的差異は for()部分と while()部分のみ)。64 と 144 の場合、単純な方法では 48 回必要だった反復処理が 1 回で終了します。ちなみに、この場合に「誤入力対策」が不要な理由については各自で考えてみましょう (hint : 64%144=64)。

```

<title>単純な方法で求める最大公約数</title>
<!-- saved from url=(0008)about:internet -->

<form>
  小さい数 = <input type="text" size="8" value="64">
  大きい数 = <input type="text" size="8" value="144">
  <input type="button" value="実行" onClick="処理()">
  最大公約数 = <input type="text" size="8">
</form>

<script>
function 処理() {
  小=document.forms[0].elements[0].value;
  大=document.forms[0].elements[1].value
  for (x=小;0<x;x--) {
    if (小%x!=0) continue
    if (大%x!=0) continue
    document.forms[0].elements[3].value=x; break
  }
}
</script>

```

初期値

formの要素は0からの順番で指定。
; は命令の区切り。行末の場合省略可。

今回の script の中身は function 1 つのみ。

%は剰余。!=は≠。continueは次要素に進む。breakは反復終了。

if 該当処理は 1 命令のみ ({} で括弧は複数命令を実行※。 ※資料3の while() {} 参照。

資料1 GCDを求める手順などを指示する JavaScript の例

資料2 資料1の内容の実行結果

Script 保存時の「文字コード」は UTF-8 がオススメ。

```

<title>互除法で求める最大公約数</title>
<!-- saved from url=(0008)about:internet -->

<form>
  小さい数 = <input type="text" size="8" value="64">
  大きい数 = <input type="text" size="8" value="144">
  <input type="button" value="実行" onClick="処理()">
  最大公約数 = <input type="text" size="8">
</form>

<script>
function 処理() {
  小=document.forms[0].elements[0].value
  大=document.forms[0].elements[1].value
  while (大%小!=0) {余=大%小;大=小;小=余}
  document.forms[0].elements[3].value=小
}
</script>

```

大÷小が「余り0」になるまで {} 内を反復実行※。 ※最初から割り切れる→ {} 内不実行の場合も GCD=小なので OK。

資料3 互除法を用いた JavaScript の改善例

- ①基本構成 : script (脚本) 及び form (用紙) と HTML。
- ②代入の書式 : 変数名=値; (条件は等 =、不等 !=)。
- ③繰り返し : for(初期値; 条件; 増減)や while(条件)。