

# HTML アプリケーションを用いた 簡易 SAS ツールの開発事例

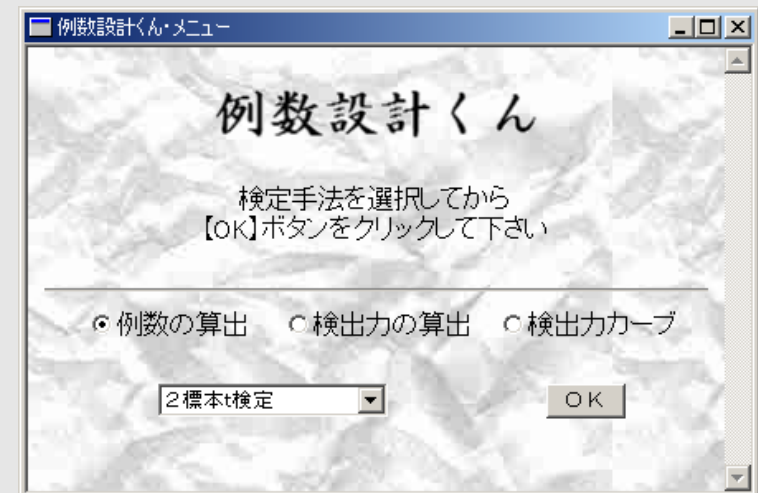
- 臨床試験における症例数設計用ツールの構築 -

舟尾 暢男、高浪 洋平  
武田薬品工業株式会社

〔 SAS を実行する為の簡易ツール作成方法を紹介 〕

- ツールは HTML/JavaScript のみで構成
- GUI (マウスでメニューを選択) で操作可能
- 作成とメンテナンスが簡単
- コストは SAS のライセンス料のみ

- ツール作成の動機
- HTML アプリケーションって何？
- 「例数設計くん」の紹介



- 臨床試験が終わった後は・・・
  - ・ SAS によりデータ解析を行ってから、
  - ・ データ解析結果に関する報告書を作成する
- 最近の臨床試験における解析報告書の分量は膨大・・・
- 臨床試験ごとにプログラムを作成するのはたいへん!

SAS プログラムの汎用マクロを作成する

SAS の汎用マクロを自動で実行するアプリケーション

(ソフト)を開発する

GUI で操作できるアプリケーションを開発する

SAS の知識が無い人でも解析報告書を作成できる

解析報告書を作成する工数確保が楽になる

- アプリケーション開発言語 :

- C++、Java、Microsoft® Visual Basic® for App. (VBA)、Microsoft® ASP.NET® など

- アプリケーション開発環境 :

- SAS® AppDev Studio™、Java Eclipse、Microsoft® Visual Studio® など

- これらを用いて開発を行う デメリット

新しい言語を覚える労力がかかり過ぎる

開発費用がかかり過ぎる

場合によってはサーバを立てる必要がある

発表者（私）は知識もお金も無い。。。

# 本日の提案っ！

〔HTMLアプリケーションを用いたツールの作成〕

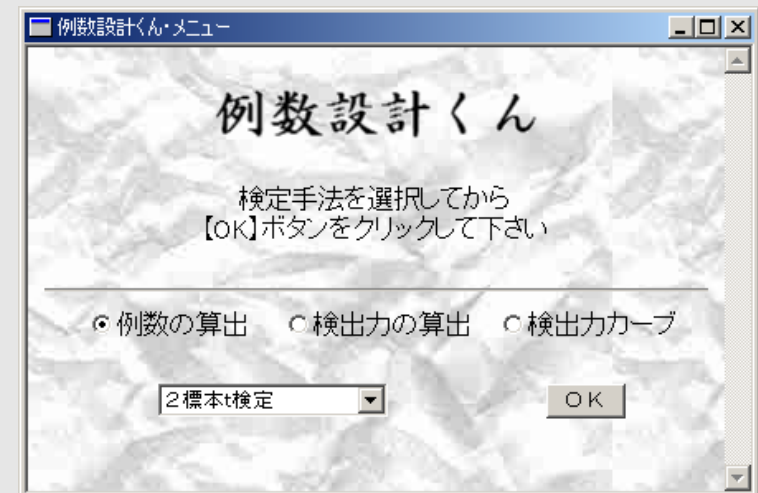
アプリケーション作成が簡単

GUI を実装（メニュー操作でデータ解析が可）

メンテナンスが簡単

実装例として症例数設計ツールを紹介  
（弊社内に解析システムが既にあったため）

- ツール作成の動機
- HTML アプリケーションって何？
- 「例数設計くん」の紹介



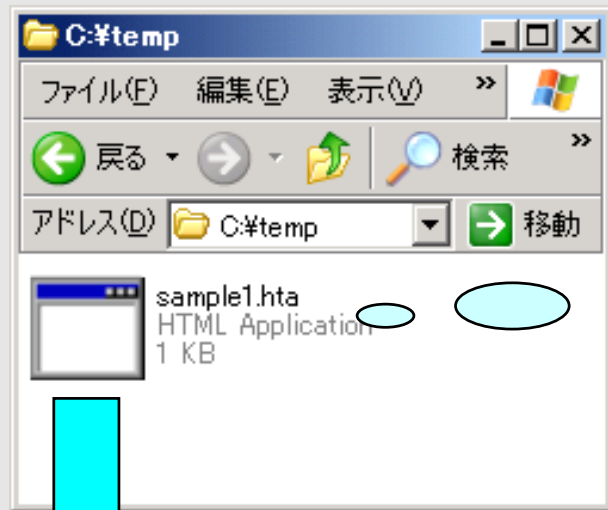
- 必要とする知識は「ホームページ作成のための HTML 言語」と「JavaScript の限られた知識」のみ

予稿集に載せている知識だけでアプリケーションが作成できる

- 大規模なウェブアプリケーションと比較して作成とメンテナンスが簡単
- PC SAS のライセンスを除く費用（サーバ代など）がかからない
- 本稿で紹介した方法ならば、SAS を実行することも Windows のアプリケーションを実行することも出来る

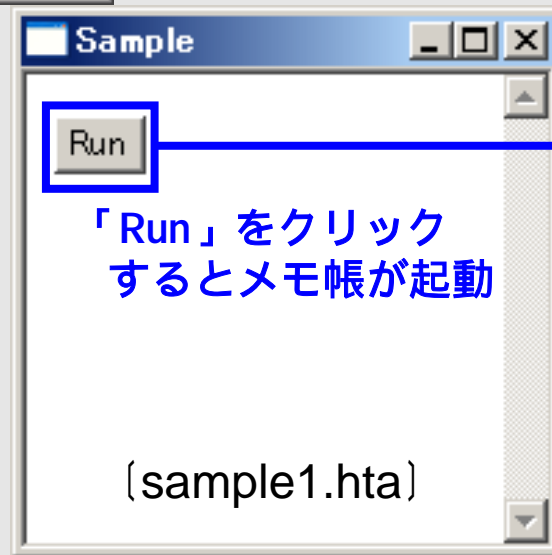
SAS とその他のアプリケーションを組み合わせたレポート作成が可能

# (例) シンプルな HTML アプリケーション HTAApplication

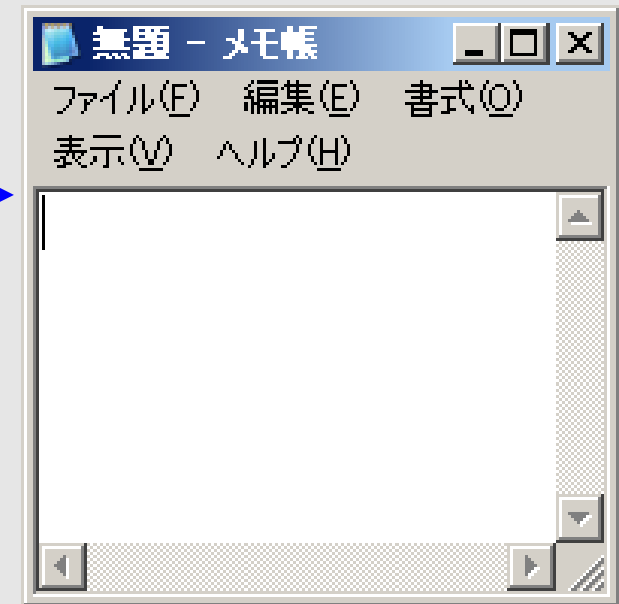


簡単な命令を書いたあと  
ファイルを保存 (.hta)

```
<html>
<title>Sample</title>
<head>
<script type="text/javascript">
resizeTo(400,300);
function MyRun() {
  MyShell = new ActiveXObject("WScript.Shell");
  MyShell.Exec('notepad');
}
</script>
</head>
.....
```

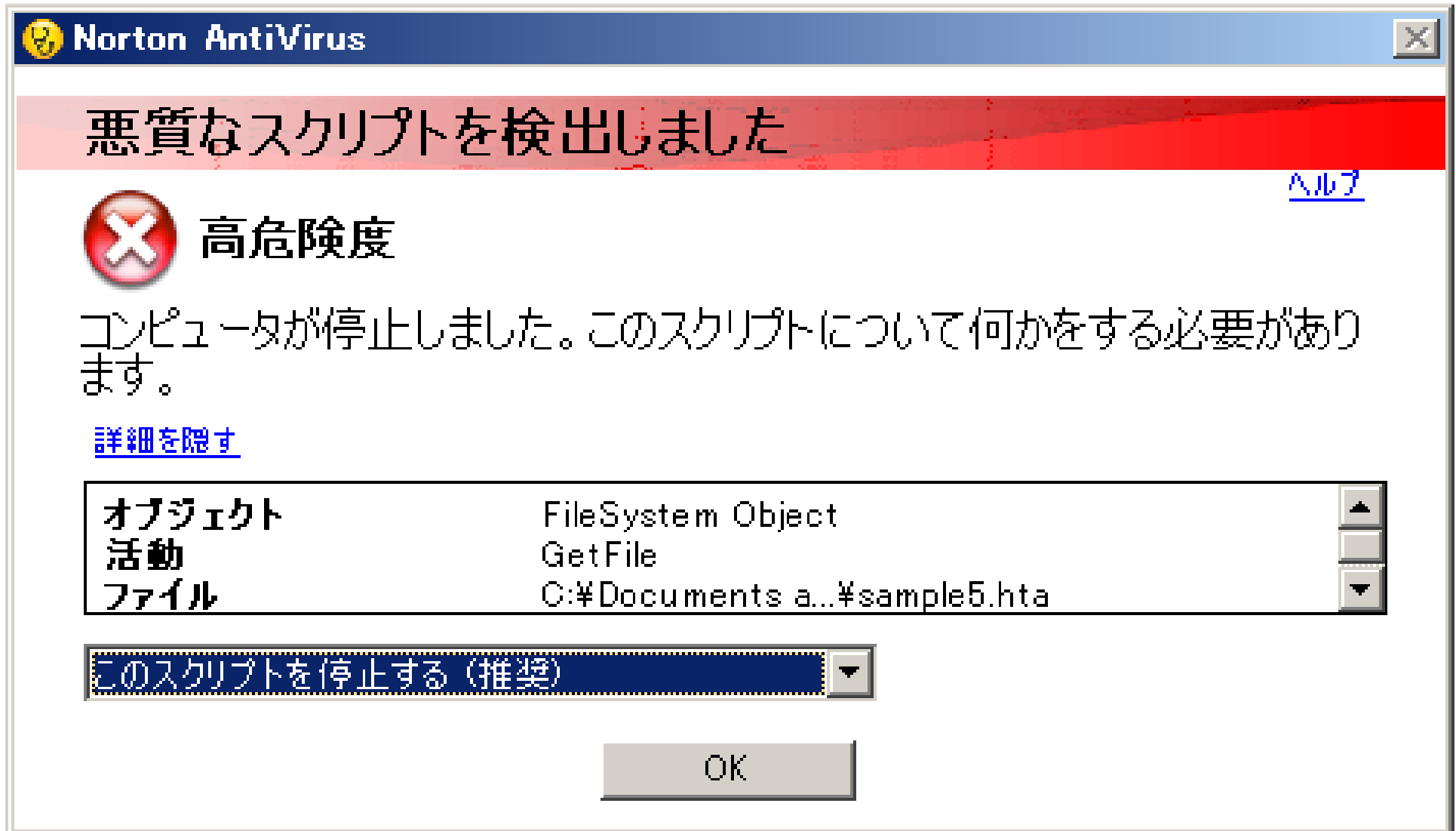


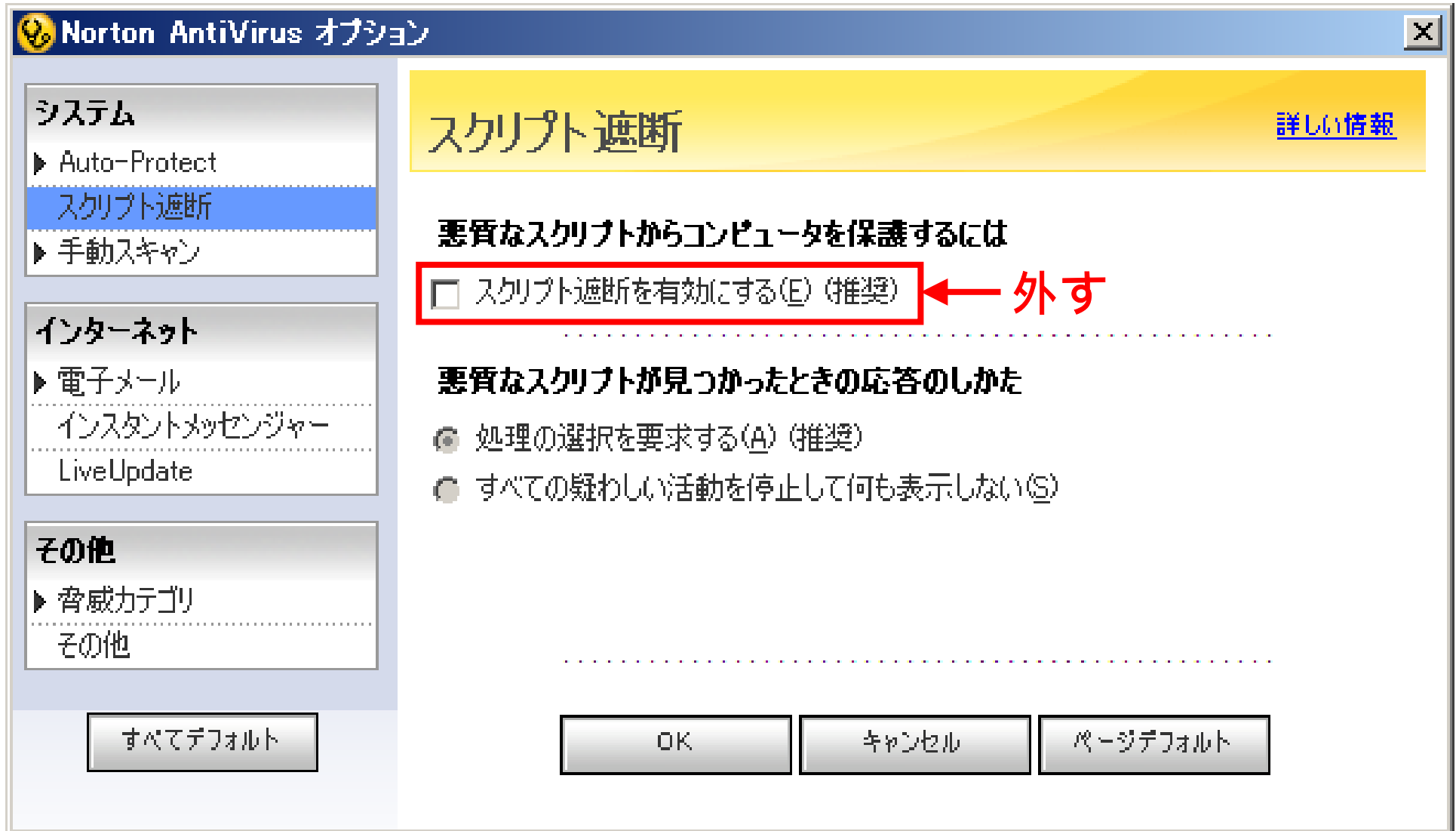
ダブルクリックして  
アプリケーションを起動



## (例 ) シンプルな HTML アプリケーション HTAApplication

1. 作成した HTML アプリケーションを「sample1.hta」という名前で適当なフォルダに保存する
2. 上記 1. をダブルクリックしてアプリケーションを起動
  - 「マウスを右クリック」 「プログラムを実行する」から「Microsoft (R) HTML Application host」を選択でも可
  - アプリケーションを実行すると、ウィンドウが表示される
3. 「Run」ボタンをクリックすると Windows のメモ帳が表示

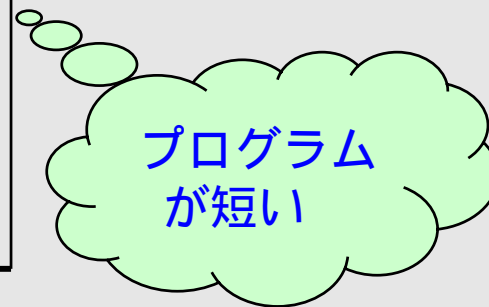




## (例) シンプルな HTML アプリケーション HTAApplication

```
01: <html>
02: <title>Sample</title>
03: <head>
04: <script type="text/javascript">
05:   resizeTo(400,300);
06:   function MyRun() {
07:     MyShell = new ActiveXObject("WScript.Shell") ;
08:     MyShell.Exec('notepad') ;
09:     MyShell = null ;
10:   }
11: </script>
12: </head>
13: <body>
14: <input type="button" value="Run" onclick="MyRun()">
15: </body>
16: </html>
```

{sample1.hta}



## (例) シンプルな HTML アプリケーション HTAApplication

```
01: <html>
02: <title>Sample</title>
03: <head>
04: <script type="text/javascript">
05:   resizeTo(400,300);
06:   function MyRun() {
07:     MyShell = new ActiveXObject("WScript.Shell") ;
08:     MyShell.Exec('notepad') ;
09:     MyShell = null ;
10:   }
11: </script>
12: </head>
13: <body>
14:   <input type="button" value="Run" onclick="MyRun()">
15: </body>
16: </html>
```

HTML の部分  
〔本体 + 見たい目〕

{sample1.hta}

## (例) シンプルな HTML アプリケーション HTAApplication

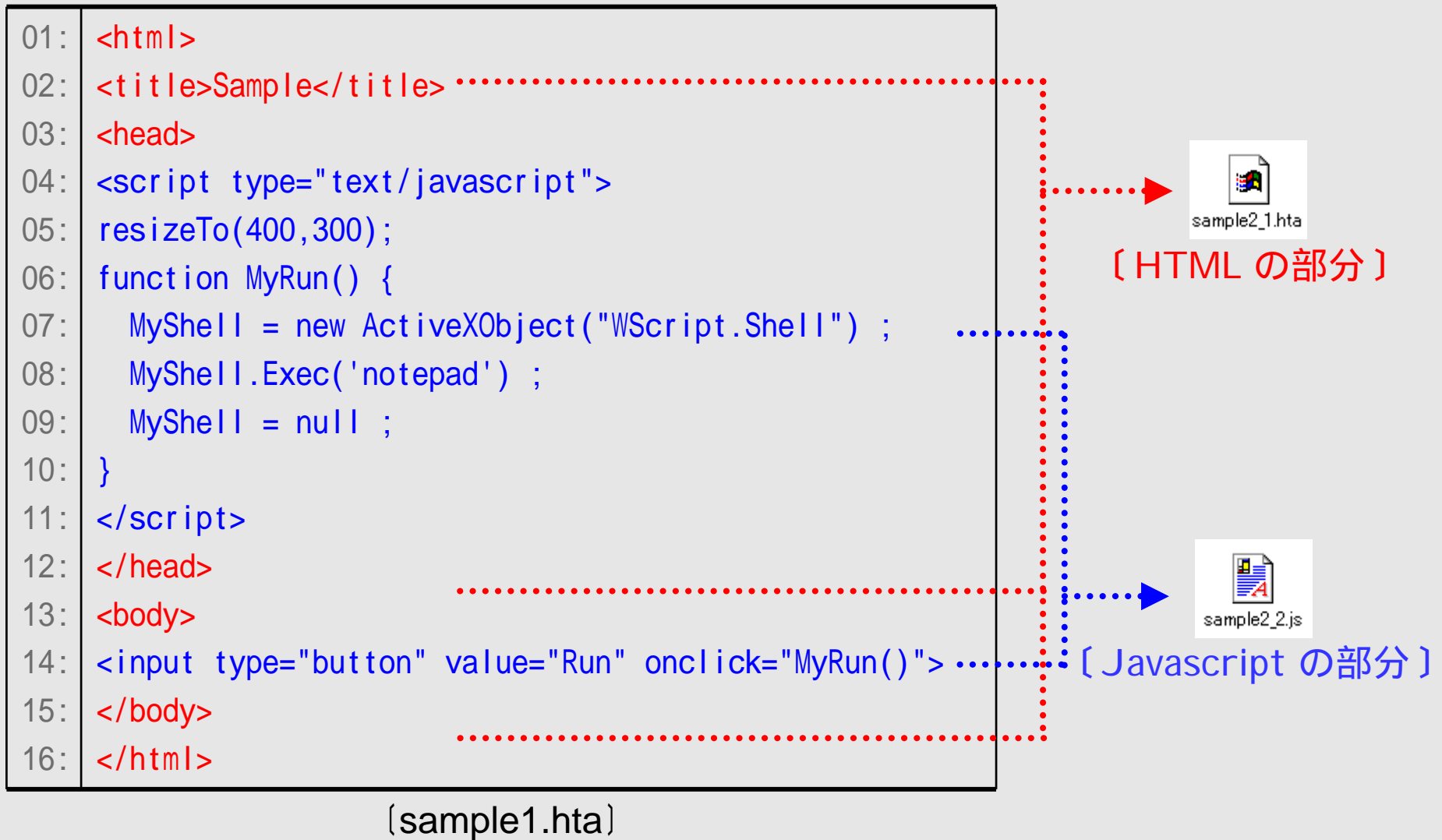
```
01: <html>
02: <title>Sample</title>
03: <head>
04: <script type="text/javascript">
05:   resizeTo(400,300);
06:   function MyRun() {
07:     MyShell = new ActiveXObject("WScript.Shell") ;
08:     MyShell.Exec('notepad') ;
09:     MyShell = null ;
10:   }
11: </script>
12: </head>
13: <body>
14: <input type="button" value="Run" onclick="MyRun()">
15: </body>
16: </html>
```

← Javascript の部分  
(メモ帳の起動)  
[プログラム]

← Javascript の部分  
(ボタンの配置)  
[部品の配置]

{sample1.hta}

# (例) HTML と Javascript の分離



## ■ 分離すると管理が楽！

```
01: <html>
02: <title>Sample</title>
03: <head>
04: <script language="JavaScript"
05:     src="./sample2_2.js"></script>
06: </head>
07: <body>
08: <input type="button" value="Run"
09:     onclick="MyRun()">
10: </body>
11: </html>
```

{ sample2\_1.hta }



[ HTML の部分 ]

```
01: resizeTo(200,200);
02: function MyRun() {
03:     MyShell = new ActiveXObject("WScript.Shell");
04:     MyShell.Exec('notepad');
05:     MyShell = null ;
06: }
07:
08:
09:
10:
11:
```

{ sample2\_2.js }



[ Javascript の部分 ]

# (例) SAS ファイルに文字を追記

The screenshot illustrates the process of generating a SAS file from an HTA application. It features three main windows:

- File Explorer (Left):** Shows the directory `C:\temp` containing a file named `sample3.hta` (HTML Application, 2 KB).
- Sample (Center):** A dialog box with input fields for `n = 10` and `k = 3`, and a button labeled `実行` (Execute).
- File Explorer (Right):** Shows the same directory `C:\temp` after execution, now containing both `sample3.hta` and a newly created `sample3.sas` file (SAS プログラム, 1 KB).

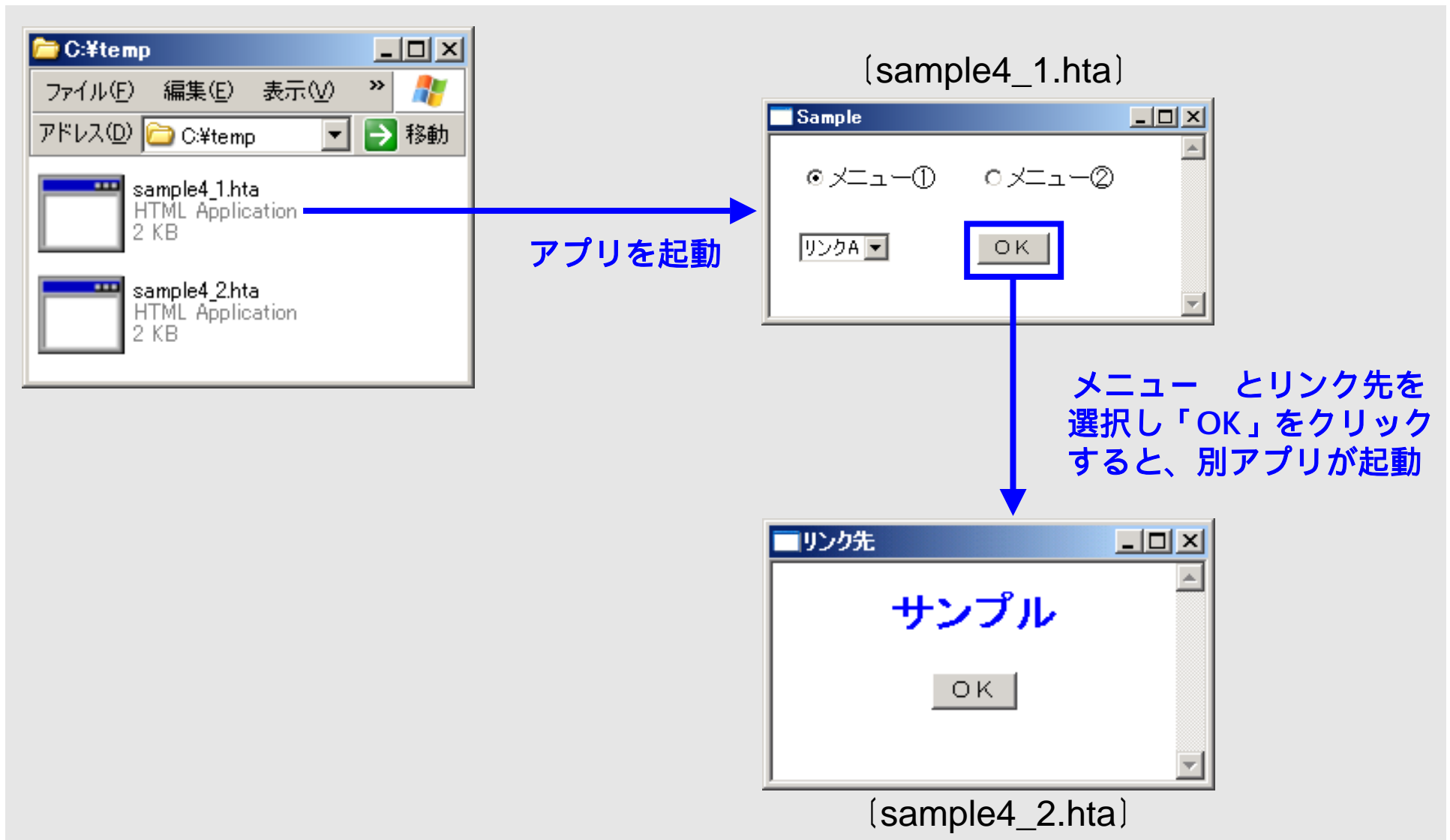
A large cyan arrow points from the `sample3.hta` file in the left window to the `実行` button in the `Sample` dialog box, with the text `アプリを起動` (Start application) below it.

A blue arrow points from the `実行` button to the `sample3.sas` file in the right window, with the text `「実行」をクリックすると .SAS が生成` (Clicking 'Execute' generates .SAS) below it.

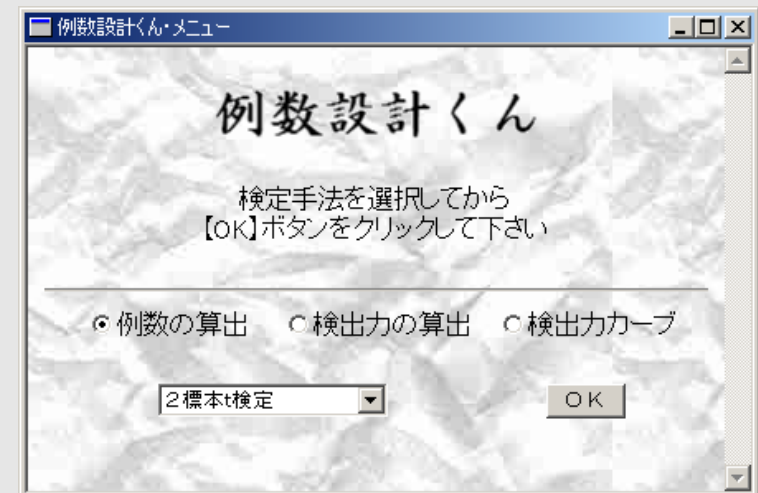
A yellow thought bubble next to the `sample3.sas` file contains the SAS code:

```
%let n = 10 ;  
%let k = 3 ;
```

# (例) メニュー選択を実装



- ツール作成の動機
- HTML アプリケーションって何？
- 「例数設計くん」の紹介



〔本稿の方法の適用例として例数設計ソフト作成例を紹介〕

「例数設計くん」の機能概要

- 例数の算出：各種パラメータと検出力を入力
- 検出力の算出：各種パラメータと例数を入力
- 検出力カーブの作成：各種パラメータと例数を入力

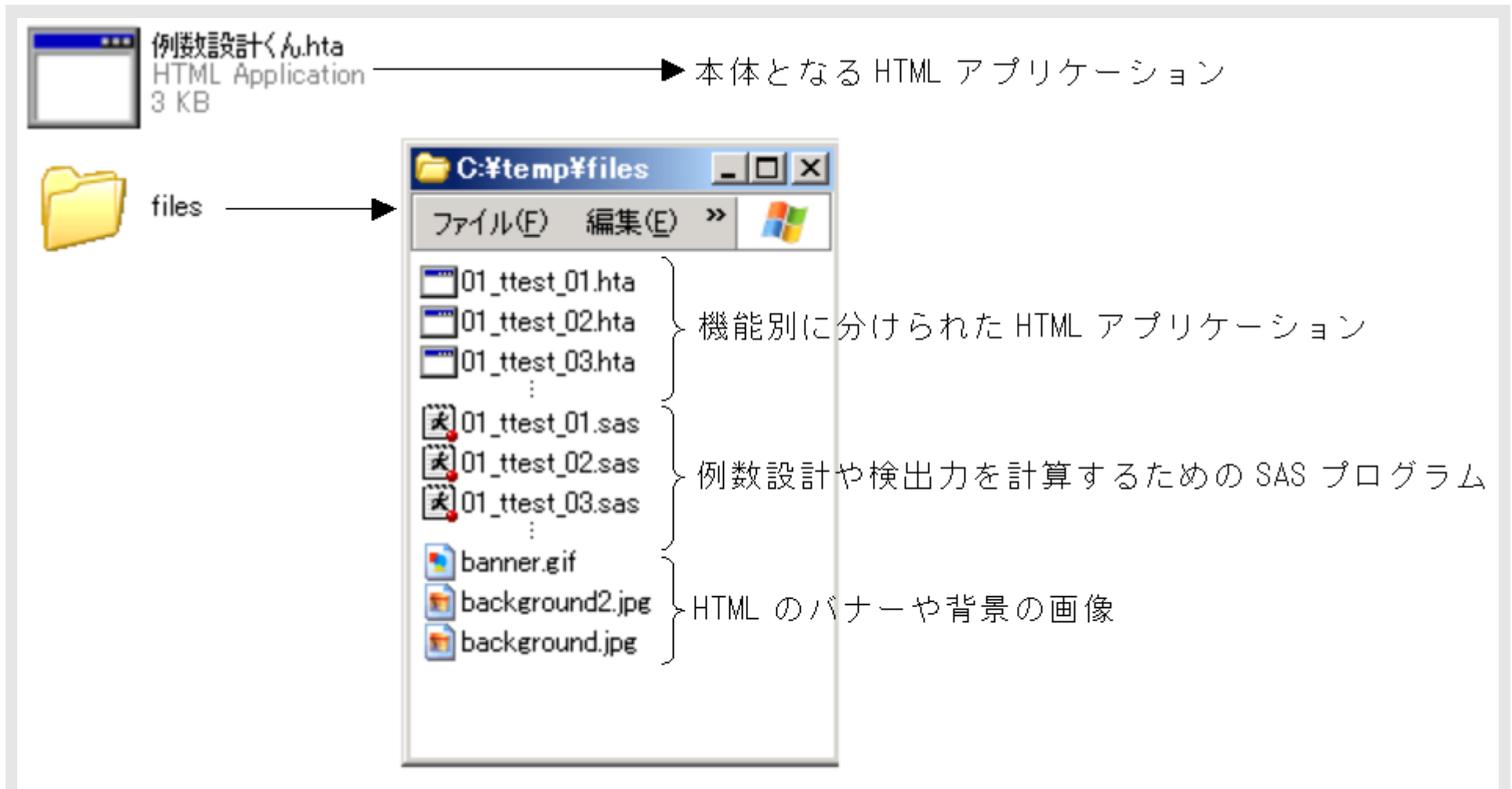
本報告の主張は「お手軽にソフトが作成できます」がメインです

例数設計ソフトの紹介がメインではありません！

例えば、臨床試験に関するデータを解析するためのツールを構築

するときも同様の方法が適用できます

# 簡易ツール「例数設計くん」



〔「例数設計くん.hta」と「files」フォルダの構造〕

# 「例数設計くん」で例数の算出

例数設計くん-メニュー

例数設計くん

検定手法を選択してから  
【OK】ボタンをクリックして下さい

例数の算出    検出力の算出    検出力カーブ

2標本t検定

2標本t検定  
2標本比率の検定  
2標本生存時間解析  
対比較定

OK

2標本t検定

各パラメータを設定してから  
【OK】ボタンをクリックして下さい

$\Delta = 1.0$

$\sigma = 1.0$

$\alpha = 0.05$

Power = 0.9

OK

SAS システム

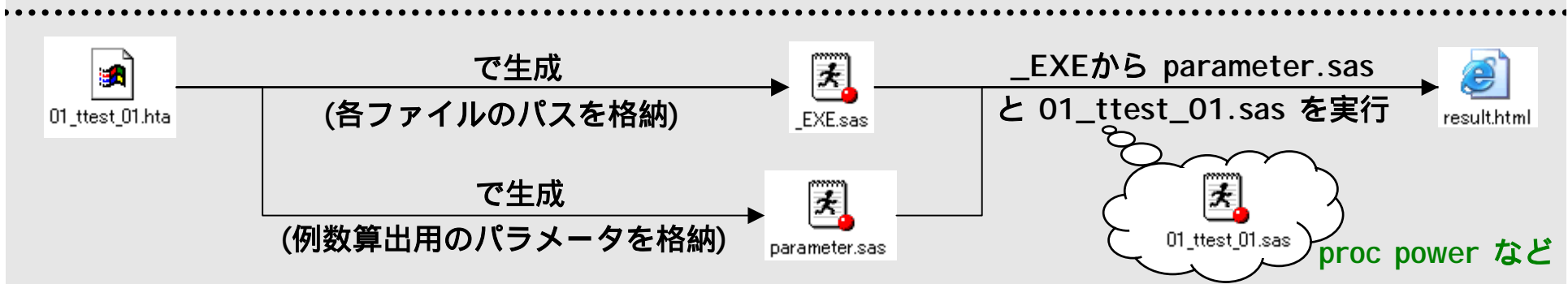
The POWER Procedure  
Two-sample t Test for Mean Difference

Fixed Scenario Elements	
Distribution	Normal
Method	Exact
Alpha	0.05
Mean Difference	1
Standard Deviation	1
Nominal Power	0.9
Number of Sides	2
Null Difference	0
Group 1 Weight	1
Group 2 Weight	1

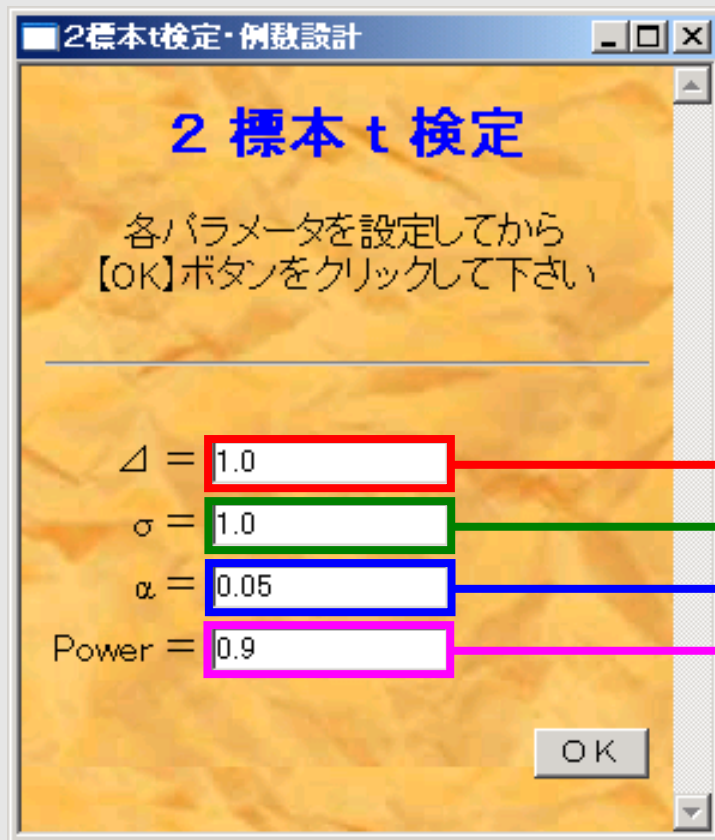
Computed N Total	
Actual Power	N Total
0.912	46

メニューと検定手法を選択して「OK」をクリック

各種パラメータと検出力を入力して「OK」をクリック



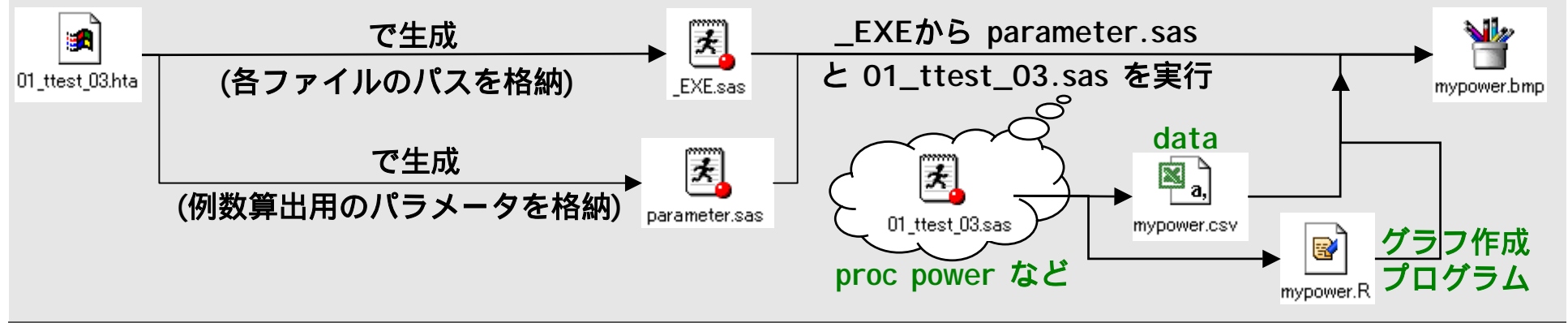
# 「例数設計くん」で例数の算出



(01\_ttest\_01.sas)

```
01: ods html file = 'result.html'
02:       path = "&_path1" style=styles.sasweb ;
03: proc power; *--- proc power で例数設計 ;
04:     twosamplemeans
05:     test      = diff
06:     meandiff  = &_meandiff.
07:     stddev    = &_stddev.
08:     alpha     = &_alpha.
09:     power     = &_power.
10:     ntotal    = . ;
11: run ;
12: ods html close;
13: x "&_path1¥result.html" ;
```

# 「例数設計くん」で検出力カーブ作り HTA Application





## 〔メリット〕

- GUI を備えたアプリが簡単に作成できる  
GUI 上で SAS を実行してレポートが自動出力できる  
SAS を知らない人でも SAS によるデータ解析ができる
- メンテナンスが容易、開発・保守費用が比較的かからない

## 〔デメリット〕

- 他言語に比べて HTA に関する資料が少ない
- OS のバージョンアップによる仕様変更が生じる可能性あり

## 〔謝辞〕

長谷川 伸弘さん（株式会社 JIEC ）