ななちゃんのIT教室

JavaScript超入門の巻

by nara.yasuhiro@gmail.com

JavaScript 超入門者ななちゃんが
ちょっと慣れるまでのお話

第 0.9 版　2017 年 5 月 7 日

もくじ

第1回 はじめてのプログラム

なな：これって、朝日新聞の「ののちゃんのDO科学」のパクリ？

先生：パロディって言ってちょうだい。家政婦のミタ（「家政婦は見た」のパロディ）、クレヨンしんちゃんのダズニーランド（「ディズニーランド」のパロディ）みたいなものよ。

フリー素材
http://freeillustration.net

なな：文部科学省が「2020年からの小学校での『プログラミング教育の必修化』を検討中」と聞いたけど、プログラミングをまったく知らなくて、とても不安なの。

先生：小学生に教えるくらいだから、とっても簡単よ。実用的な JavaScript 言語を勉強しましょう

なな：言語？ マイクに向かって「2 足す 3 はいくつ？」とか聞くの？

;

先生：エディタ（Windows メモ帳、Macintosh テキストエディット、emacs、mi、Atomなど）を使い、右の内容のファイルを作ってみてね。ファイルの名前は「sample1.html」にしてね。

```
<script>
alert ( "Hello, world." ) ;
</script>
```

フリー素材
http://freeillustration.net

なな：はい。できました。

先生：それでは、作ったファイルのアイコンをダブルクリックしてみてね。

なな：こんな画面になったわ！

先生：こういう画面を「アラート画面」というの。「OK」をクリックして、画面を閉じてね。これで、簡単だけど、JavaScript のプログラムを実行したことになるのよ。今後、日本語の文字を表示したり、いろいろなブラウザで確実に表示させたり、普通のウェブページと組み合わせるには、次のような内容にしたほうが良いのよ。→

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>sample</title>
  </head>
  <body>
    <script>
alert ( "Hello, world." ) ;
    </script>
  </body>
</html>
```

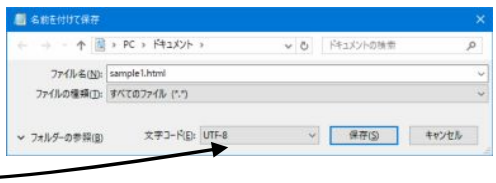
なな：ずいぶん長くなってしまうのね。毎回、こんなにたくさん入力するのは面倒ね。

先生：こういう内容のファイルを作っておいて、新しいプログラムを作るときには、このファイルをコピーして、<script> と </script>の間だけ書き換えるようにするといいのよ。 どこか、1文字だけ書き間違えても正しく動作しなくなる可能性があるからね。

なな：コピーか！ はい、わかりました。ところで、「Hello, world.」って何？ 「世界よこんにちは」？

先生：ひ・み・つ。気になったら、インターネット検索で調べてみてね。

JavaScript プログラムのファイルは「UTF-8」という文字コードで保存してください。



第2回 コンピュータが話をする（ alert ）

先生：前回は、「Hello, world.」というメッセージウィンドウ（アラート）を表示するプログラムを勉強しました。

なな：「alert ("Hello, world.");」ね。 alert かあ。英語を使うのね。

先生：Netscape Communications社という、米国の会社が作った言語だからね。

なな： 英語国民だったら勉強しなくても分かるから得みたいね。

先生：英語国民だと、日常のことばとごっちゃになるという悩みもあるらしいわよ。「マウスの左ボタンをクリック」というのが「ネズミの左ヘソをカチャ！する」 みたいに聞こえるのかも。

なな：ほかに、どんなことができるの？

先生：ひとことで言うのはむづかしいけど、電子計算機を使うから計算が得意なの。

なな：じゃあ、「ルート 2」なんか計算できるの？

先生：「alert (Math.sqrt(2)); 」ね。「1.4142135623730951」と表示されるわ。

なな：「一夜一夜に人見頃」ね。 はじめから「alert(1.4142135623730951);」と書いたほうが簡単なのに。

先生：う〜ん、するどいわね。 でも、Webページの入力枠に「3」と書き込んでから、「計算」ボタンをクリックするとルート 3、「5」なら ルート 5 と、好きな数のルートを表示してくれると便利でしょ。

なな：「Hello, world.」は「alert ("Hello, world.");」と、「"」で囲んだけど、「alert (Math.sqrt(2));」には「"」が無いの？

先生：「alert("Math.sqrt(2)");」だと、「1.4142135623730951」じゃなくて「Math.sqrt(2)」と表示されてしまうの。

なな：「ルート 2 はいくつ？」と聞かれて、「ルート 2 はいくつ？」とオウム返ししてしまうのね。

先生：そうなの。「Math.sqrt(2)」は、「数学(Mathematics) の平方根 (squareroot) の 2 はいくつ？」という計算をお願いしているの。

なな：だから「alert (1.4142135623730951);」と言うのと同じ結果になるのね。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>sample</title>
  </head>
  <body>
    <script>
alert (Math.sqrt(2) ) ;
    </script>
  </body>
</html>
```

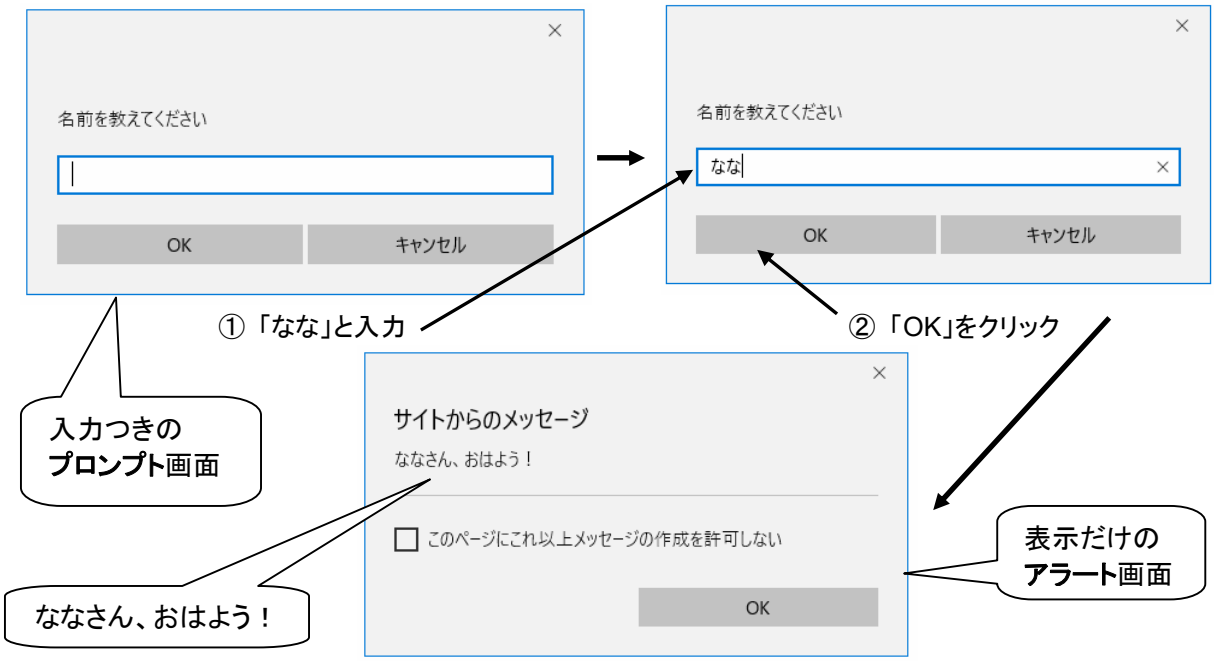
http://www.irasutoya.com/

第3回 コンピュータが人間に耳を傾ける（ prompt ）

先生： 前回は、コンピュータが何かを表示してくる、というのをやりました。今回は、人間がコンピュータに語りかけるというのをやりましょう。ななちゃん、次のプログラムを動かしてみてね。

```
var namae = prompt ( "名前を教えてください" ) ;
alert ( namae + "さん、おはよう！" ) ;
```

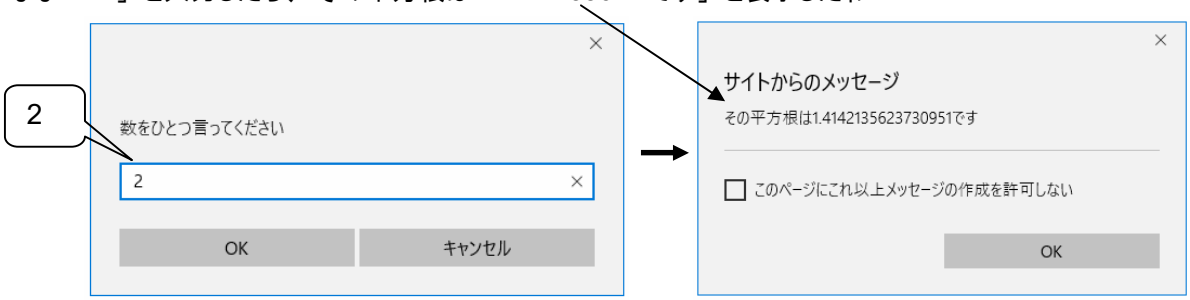
なな： 「なな」と入力したら、「ななさん、おはよう！」というアラートが出たわ！



先生： 「prompt（“名前を教えてください”）」は、「名前を教えてください」と表示した「**プロンプト**画面」を出して、ユーザの入力を待つの。ユーザが文字列を入力して、「OK」ボタンをクリックすると、その文字列をJavaScript に送ってくるのよ。 今回のプログラムでは、その文字列を「namae」という名前の箱に記憶するの。「var」は「variable」（**変数**）の略語で、箱を用意してね、という意味なの。「variable」は、英語で「変化し得るもの」というような意味で、いろいろなデータを次から次へと記憶するということなの。 次の行では、今 記憶した文字列と、「さん、おはよう！」という文字列を「結合」しているのよ。JavaScript の「+」は、文字列なら「結合、連結」をする命令、数字なら「足し算する」という命令なの。 場合によって、機能が異なるので、入門者はちょっと とまどうかも。 そして、結合した文字列全体を alert で表示しているのよ。じゃあ、ななちゃん、次のプログラムはどうかしら。

```
var kazu = prompt("数をひとつ言ってください") ;
alert ( "その平方根は" + Math.sqrt( Number(kazu) ) + "です") ;
```

なな： 「2」と入力したら、「その平方根は1.41421356….です」と表示したわ！



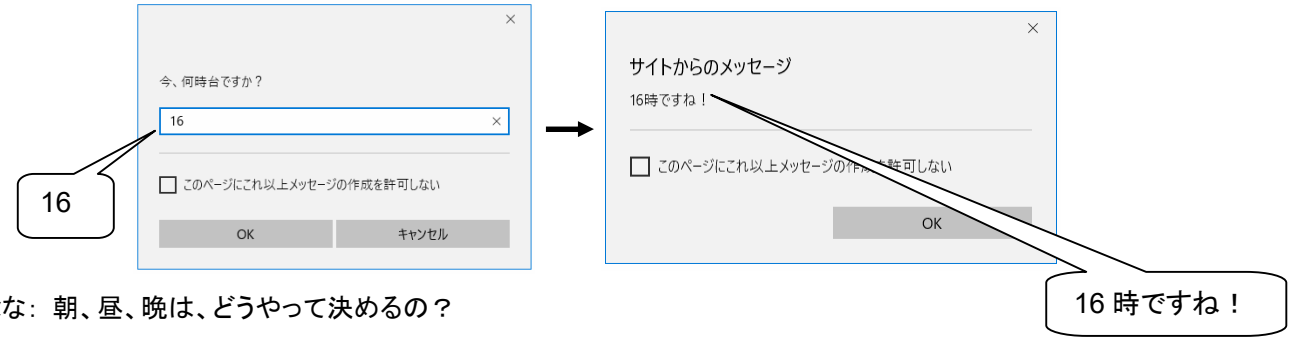
先生： 「Number」は、文字列を数字に変換する命令だけど、ルート（sqrt）は数字が前提で、Number をつけなくても「自動変換」してくれます。ですから「alert("その平方根は" + Math.sqrt(kazu) + "です");」でも大丈夫。

第4回 コンピュータが判断をする（判断/分岐）

先生： 前回は、人間とコンピュータが対話するようなプログラムを作ったわね。今回は、コンピュータに「判断」をしてもらうプログラムに挑戦しましょう。現在の時間を入力すると、その時間に合わせて、「おはよう」「こんにちは」「こんばんは」のいずれかアラートしてくるプログラムを作ってみましょう。 最初は、準備として、次のプログラムを実行してみてね。

```
var h = prompt ( "今、何時台ですか？" ) ;
alert ( h + "時ですね！" ) ;
```

なな：「16」と入力すると、「16時ですね！」と表示するわ。



なな：朝、昼、晩は、どうやって決めるの？

先生：朝、昼、晩の区別は、人によって意見が分かれるので、ななちゃんが自分で決めないといけないわ。

なな：じゃあ、9 時台までが朝、15 時台までが昼、16 時台以降が晩かな。

先生：まず、昼の「こんにちは！」から。「alert」 の行を、次のように変えるの。

```
if ( h < 16 ) { alert ( "こんにちは！" ) ; }
```

なな：「h < 16」って何？

先生：この行の代わりに「alert (h < 16) ;」と書くとどうなるかしら？

なな：「true」と表示されるわ。

先生：これは「真」、つまり、「『今は 16 時より前』である」ということね。

16 時を過ぎると「false」、つまり「偽」、「『今は 16 時より前』ではない」ということになるわ。
「if (true) {...}」は、「...」の部分を実行し、
「if (false) {...}」は、「...」の部分を実行しない、無視するということなの。
「if」は「もし 真 なら」という意味なの。

なな：16 時以降だったら「こんばんは！」と表示するのを追加するには、次のようにすれば良いの？

```
if ( h < 16 )  { alert ( "こんにちは！" ) ; }
if ( h > 16 )  { alert ( "こんばんは！" ) ; }
```

先生：これだと、16 時台の時に何も実行されないの。次のようにする必要があるの。

```
if ( h >= 16 )  { alert ( "こんばんは！" ) ; }
```

午前 10 時台より前だったら「おはよう！」とするのを追加するには？

なな：

```
if ( h < 10 ) { alert ( "おはよう！" ) ; }
if ( h < 16 ) { alert ( "こんにちは！" ) ; }
if ( h >= 16 ) { alert ( "こんばんは！" ) ; }
```

先生：これだと、9 時で「おはよう！」と「こんにちは！」の両方表示されてしまうわ。

```
if ( h < 10 )                    { alert ( "おはよう！" ) ; }
if ( ( h >= 10 ) && ( h < 16 ) ) { alert ( "こんにちは！" ) ; }
if ( h >= 16 )                   { alert ( "こんばんは！" ) ; }
```

とするの。「&&」は、「かつ」とか「しかも」という意味なの。

なな：「&」ひとつではダメなの？

先生：ダメよ。「&」と「&&」は別物なの。「& &」のように、隙間を空けて書いてはいけないの。

なな：面倒ね。

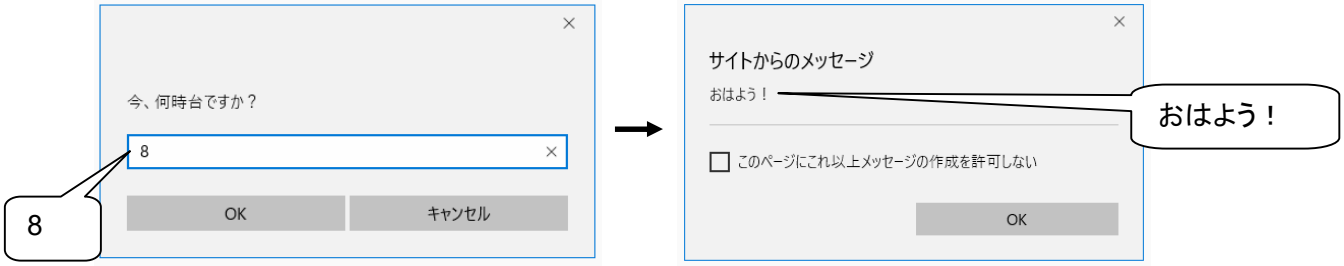
先生：次のような書き方もできるわ。「else」は「そうではなくて」という意味よ。

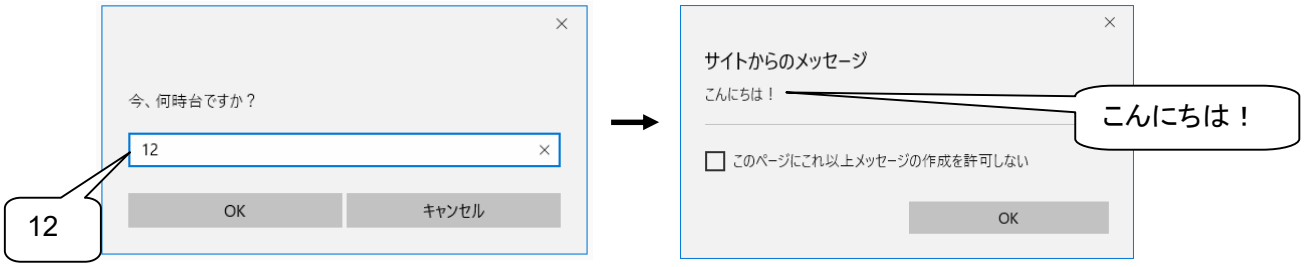
```
if      ( h < 10 ) { alert( "おはよう！" ) ; }
else if ( h < 16 ) { alert( "こんにちは！" ) ; }
else               { alert ( "こんばんは！" ) ; }
```

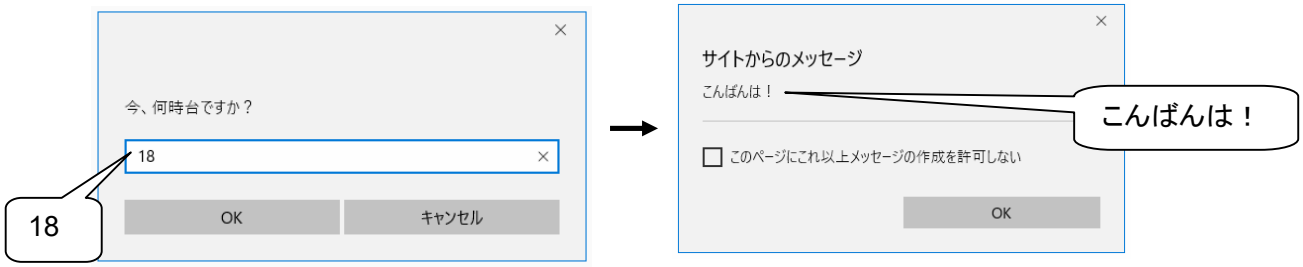
なな：ずいぶん、すっきりするのね！

```
var h = prompt ( "今、何時台ですか？" ) ;
if ( h < 10 )                    { alert ( "おはよう！" ) ; }
if ( ( h >= 10 ) && ( h < 16 ) ) { alert ( "こんにちは！" ) ; }
if (h >= 16)                     { alert ( "こんばんは！" ) ; }
```

この3つは
独立、
別々








```
var h = prompt ( "今、何時台ですか？" ) ;
if      (h < 10)  { alert ( "おはよう！" ) ; }
else if (h < 16)  { alert ( "こんにちは！" ) ; }
else              { alert ( "こんばんは！" ) ; }
```

もし … なら

そうでなくて
もし … なら

そうでなければ

この3つは
つながって
いる
else が
カギ

よりスマートなプログラム

ちょっとおまけ

先生: if (…) の、「 … 」の部分に書けるものの例を、下の表に、まとめてみました。

なな: 「if (x %2 == 0) 」と書いたら、「もし x を 2 で割った余りが 0 だったら」という意味になるということね。

先生: それだけだったら、どうということはないんだけど、大切なのは、「もし x を 2 で割った余りが 0 だったら」というのが、「もし、x が 偶数だったら」という意味でもあるということなの。ここでは、x が、小数点以下をもたない、「整数」だという前提の場合だけどね。ユーザが入力した数で、その後、割り算を使っていないとか。

なな: 「if (x == 偶数」とか、「if (x == even)」とかいう書き方ができないということ?

先生: そうなのよ。そこが、コンピュータが「融通が利かない」ところなの。でも、実際にプログラムを書く時に必要になるのは、たとえば、「ページ数が偶数だったら 内容を左に寄せる」なんていうことだったりするわけね。

なな: ふうん。「もし x が偶数だったら」というのを「if (x %2 == 0)」と表現できないといけないのね。

先生: そう。プログラムを書く時に、その場でアイデアを思いつくということも大切だけど、いくつかのパターンというか、常套句みたいなのがあるから、慣れておく必要があるの。下の表で、特に、**太字で下線のある部分**から、表の左側の表現が書けるように練習しておいてね。

なな: は〜〜い。

x < 10	「x は 10 未満である」、「**x は 1 桁である**」
x <= 10	「x は 10 以下である」
	「=<」と書いてはいけない。エラーになる。
	「< =」のように、間にスペースを入れてはいけない。エラーになる。
x > 0	「x は 0 より大きい」、「**x は、正数**」
x >= 0	「x は 0 以上」、「**x は負ではない**」
	「=>」と書いてはいけない。エラーになる。
	「> =」のように、間にスペースを入れてはいけない。エラーになる。
x < 0	「x は 0 より小さい」、「**x は負である**」
! (x < 0)	「x は 0 より小さくない」、「**x は 負ではない**」
(x >= 1) && (x <= 12)	「x は 1 以上、かつ、12以下」、「**x は 1 〜 12**」
(x >= 10) && (x <= 99)	「x は 10 以上、かつ、99 以下」、「**x は 2 桁**」
x == 2	「x は 2 に等しい」、「x は 2 である」
	「=」(代入)と混同しないように。
	「= =」のように、間にスペースを入れてはいけない。エラーになる。
x != 2	「x は 2 に等しくない」、「x は 2 ではない」
(x == 1) || (x == 3) || (x == 5) || (x == 7) || (x == 8) || (x == 10) || (x == 12)	「**x は 大の月**」
(x <= 10) && (y <= 10)	「x は 10 以下、かつ、y は 10 以下」、「**x も y も 10 以下**」
(x < 10) || (y <10)	「x は 10 未満、または、y は 10 未満」、「**x か y が 10 未満**」
	「**x か y のどちらか、または両方が 10 未満**」
(x < 10) && (x > 0)	「x は 10 未満、かつ、0 より大きい」、「**x は 1 桁の正数である**」
x % 4 == 0	「x を 4 で割った余りが 0」、「**x は 4 で割り切れる**」、
	「**x は 4 の倍数**」(「**x = 4 * n**」とは書けない)
x % 4 != 0	「x を 4 で割った余りが 0 ではない」、「**x は 4 で割り切れない**」、
	「**x は 4 の倍数ではない**」
x % 2 == 0	「x を 2 で割った余りが 0」、「**x は 偶数である**」
x % 2 != 0	「x を 2 で割った余りが 0 ではない」、「**x は 奇数である**」
x == “はい”	「x は 文字列の はい である」
x != “はい”	「x は 文字列の はい ではない」

注意:上記は「x が 小数点を含まない数(整数)の場合」

第5回 根に持つタイプ（変数）

先生：次の JavaScript プログラムを実行すると、どんな表示になるかな?

```
var x; x = 1; alert(x);
x = x + 1;    alert(x);
```

なな：「x と 1 は等しい」、「x と x+1 が等しい」?

先生：JavaScript の「=」は「等しい」という意味じゃなくて、「=」の右側の計算をやってから、その計算結果を「=」の左側の「変数」に記憶させるという意味なの。

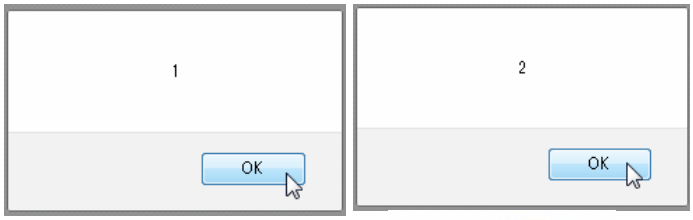
なな：じゃあ「x = 1;」は「数 1 を 変数 x に記憶する」ということ?

先生：正解。では「x = x + 1;」は?

なな：「x + 1」は「1 + 1」のことだから「2」のことね。じゃあ、それを変数 x に記憶するの?

先生：その通り！「x = x + 1;」は「x が記憶している数を 1 だけ増やす」という意味になるの。

なな：じゃあ、最初の「alert(x);」は alert ウィンドウで「1」を表示して、「OK」をクリックすると、次の「alert(x);」で「2」と表示するのね。



先生：そう。では、次のプログラムは?

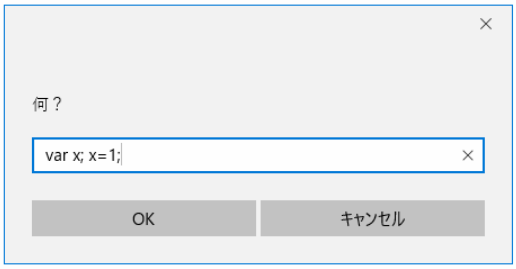
```
var x; x = 1; alert(x);
x = x + 1;    alert(x);
x = x + 1;    alert(x);
```

なな：「1」、「2」、「3」という表示になるのね！

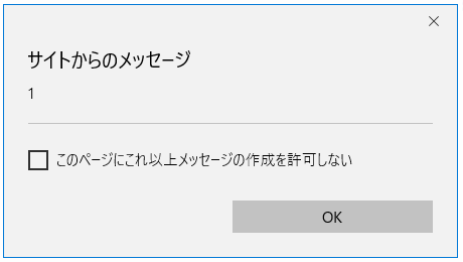
先生：今は、この(↓)プログラムの意味は理解できなくて良いけど、今の変数の仕組みを簡単に体験できるわ。試してみてね。「bye」と入力すると終了。

```
for(;;) alert(eval(prompt("何？")));
```

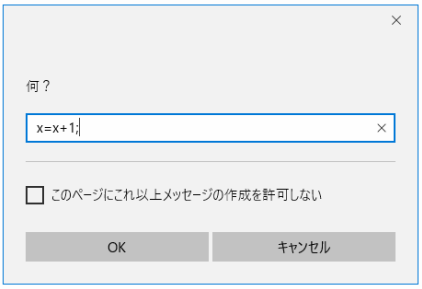
実は
エラー終了



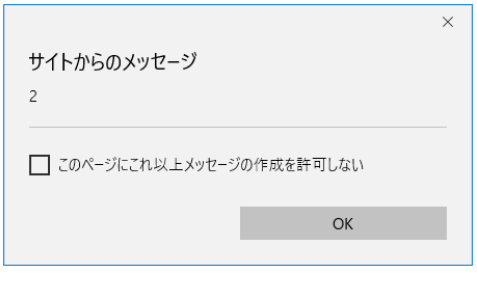
var x; x=1; と入力して「OK」クリック



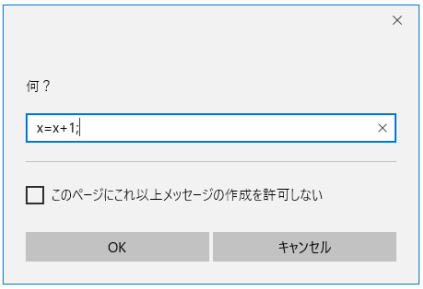
「1」と表示



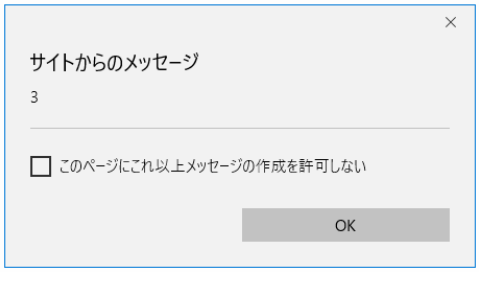
x = x + 1 と入力して「OK」



「2」と表示



x = x + 1 と入力して「OK」



「3」と表示

第6回　まわって、まわって、まわって、まわる…（くりかえし）

先生：次のプログラムを実行してみてね。

```
var x ; x = 1 ;
while (true) {
    alert ( x ) ;
    x = x + 1 ;
}
```

古い！

な な：つぎつぎにアラート画面が出て、「1」、「2」、「3」、「4」、「5」..... いつまで続くの？

先生：永遠に続くの。「10」までで終わりにしたければ、「while(true)」のところを 「while(x <= 10)」 に変えるの。

```
var x; x = 1;
while ( x <= 10 ) {
    alert ( x ) ;
    x = x + 1 ;
}
```

な な：「while (x <= 10) { ... }」と「if (x <= 10) { ... }」は似ているけど、違いは？

先生：「if (x <= 10) { ... }」は、

「『変数 x が記憶している数が 10 以下なら ... を実行する』、ということを 1 回やってしておしまい、
変数 x が記憶している数が 10 以下でないなら何もしないでおしまい」、

「while (x <= 10) { ... } 」は、

「『変数 x が記憶している数が 10 以下なら ... を 1 回実行する』ということを永遠に繰り返す、
変数 x が記憶している数が 10 以下でないなら何もしないでおしまい」

という違いがあるのよ。「if は場合分け」、「while は繰り返し」。

な な：じゃあ、この「while (x <= 10) 」の繰り返しは永遠に続くの？

先生：「 … 」を実行した結果、x が記憶している数が変化して 10 より大きくなれば それでおしまいになるということ。

な な：「while (x <= 10) { ... }」で、最初に変数 x が記憶している数が 10 以下でない場合にはどうなるの？

先生：「 ... 」の部分が 1 回も実行されないでおしまいになるの。

「while」は、英語で「…する間」という意味ね。「Strike while the iron is hot. 」（鉄は熱いうちに打て、好機を逃がすな）という格言があるわね。英語の世界では「人は純粋な気持ちを失わない若いうちに鍛錬すべきである」という意味は無いんだって。.

```
var x ; x = 1 ;
while ( x <= 10 ) {
    alert ( x ) ;
    x = x + 1 ;
}
```

は、

```
for ( var x = 1; x <= 10; x = x + 1 ) {
    alert ( x ) ;
}
```

と書くことができます。「while (true) { … } は、「for (; ;) { … }」と書くこともできます（永久繰り返し）。

第7回　成績表（配列）

なな：成績表のデータが、

abe = 100;　ito = 90;　eto = 95;　のように、100 人分あるんだけど、合計を計算するのに、total = abe + ito + eto + ...　のように書くのは大変ね。

先生：そういう時は「配列」を使うの。出席番号順に

seiseki = [100, 90, 95, ...];

のようにするの。abe さんの成績は seiseki[0]、ito さんの成績は seiseki[1] として使えるのよ。[ ] の中が 0 から始まることに注意してね。

なな：合計は？

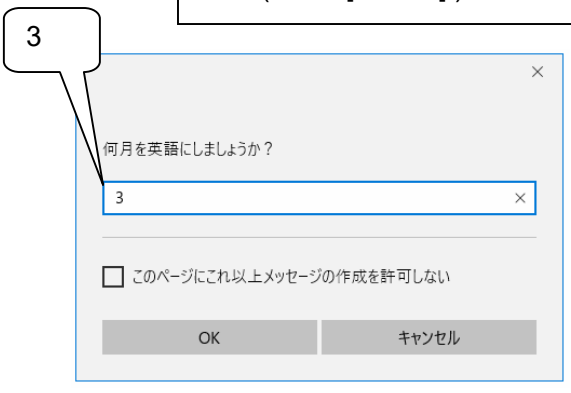
先生：前に勉強した「while」が使えるわ。

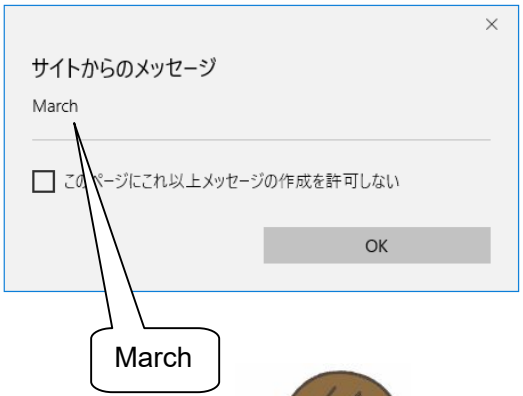
```
var number = 0 ;
while ( number <= 99 ) {
    total = total + seiseki [ number ] ;
    number = number + 1 ;
}
alert ( total ) ;
```

みたいになるわ。

この応用で、次のようなプログラムも作れるわ。

```
var table = [ "January",    "February", "March",     "April",
              "May",        "June",     "July",      "August",
              "September", "October",  "November", "December" ] ;
var m = prompt ( "何月を英語にしましょうか？" ) ;
alert ( table [ m – 1 ] ) ;
```





なな：「3」と入力すると、「March」と表示するのね！

table [m] じゃなくて table [m – 1] になるのがミソね。

第8回 まとめて名前をつける（関数）

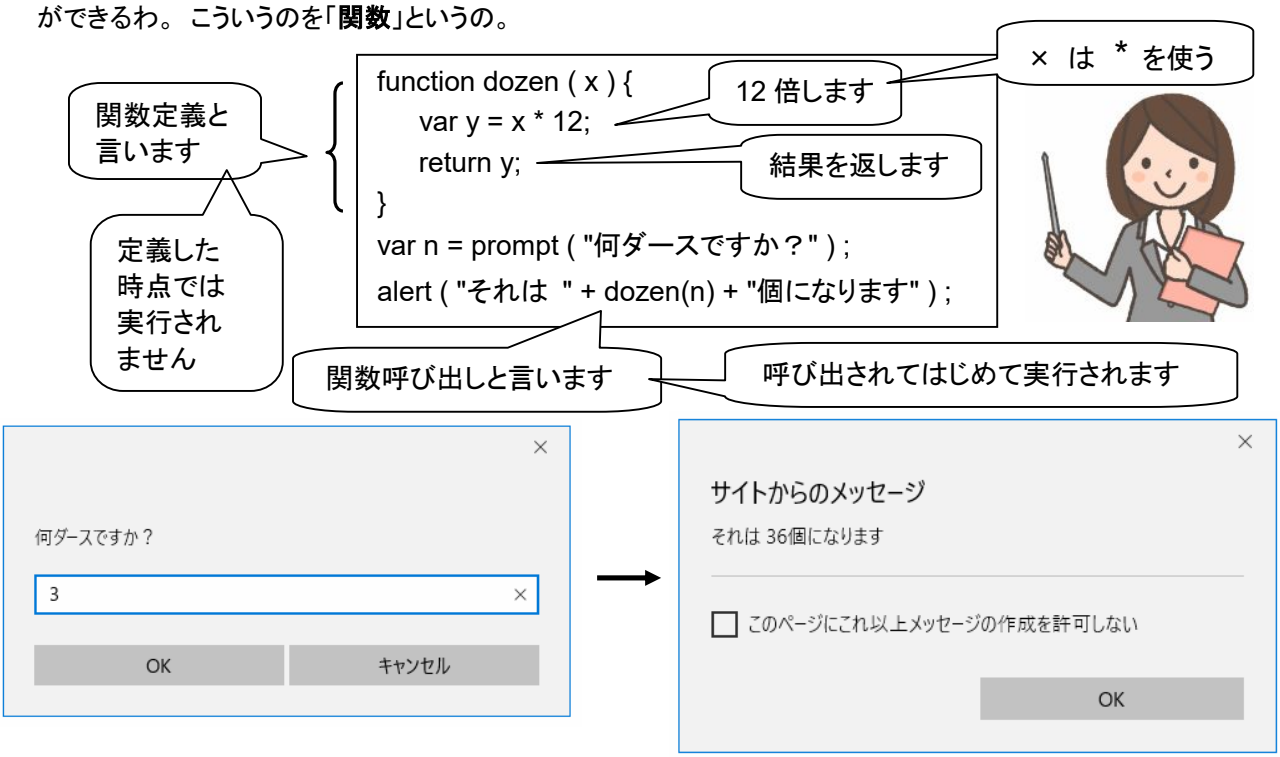
先生： プログラムの中に、同じ処理が何回も出てくるとき、ひとつだけ用意して、名前をつけておき、それを名前で呼び出すことができます。

なな： ？？？？？

先生： たとえば、「ABBBBC」という構造のプログラムなら、「B を 4 回繰り返す」ということで、while や for が使えるけど、「ABCBDBE」という構造のプログラムだと、while や for が使えないわね。「B」がいくつも出てくるのに。

なな： ？？？？？

先生： たとえば、ダース から 個数を計算するプログラムを、「dozen」という名前でまとめておくと、こんな使い方ができるわ。 こういうのを「**関数**」というの。

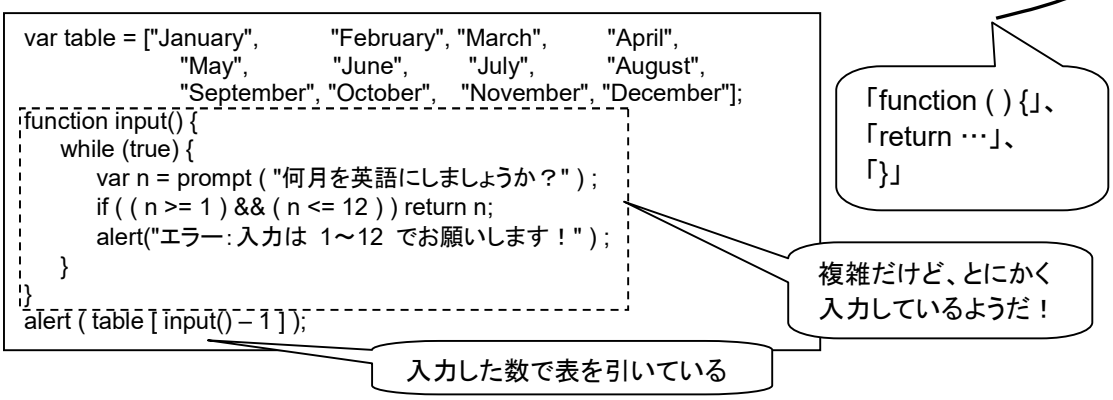


なな： dozen (n) と書くのと、n * 12 と書くのと、あまり手間は変わらないような。

先生： これは、説明のための例であって、もっと複雑な処理だったら、プログラムの行数が減るということなの。たとえば、20 行あるプログラムをまとめるとして、それを 5 回使うことを考えてみましょう。

なな： 関数を使わないと、20 行 × 5 ＝ 100 行。関数を使うと、定義に 20 行 + 3 ＝ 23 行。呼び出しが 1 行 × 5 ＝ 5 行、合わせて 28 行ということね。

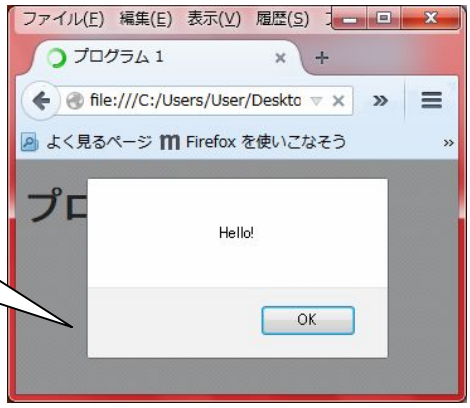
先生： それに、関数の名前を 内容が分かりやすい名前にしておくと、プログラムが読みやすくなるの。下のプログラムは、1 から 12 の数が正しく入力されるまで、何度も入力を繰り返す例。複雑だけど、とにかく「ここで入力を行っている」というのが分かりやすいと思います。



いろいろな出力方法

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>プログラム 1</title>
  </head>
  <body>
    <h1> プログラム 1 </h1>
    <script src="prog1.js"></script>
  </body>
</html>
```

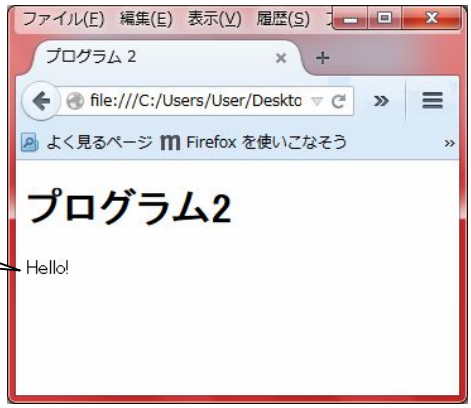
alert
ポップアップ




```
alert("Hello!");
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>プログラム 2</title>
  </head>
  <body>
    <h1> プログラム 2
    <script src="prog2.js"></script>
  </body>
</html>
```

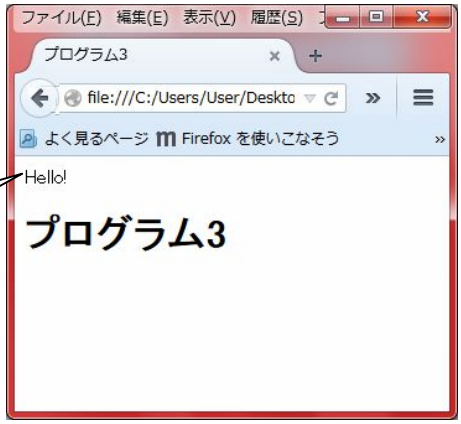
画面末尾に
追加




```
document.write("Hello!");
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>プログラム 3</title>
  </head>
  <body>
    <div id="area"></div>
    <h1> プログラム 3 </h1>
    <script src="prog3.js"></script>
  </body>
</html>
```

div 領域
に出力




```
var areap = document.getElementById("area");
areap.innerHTML = "Hello!";
```

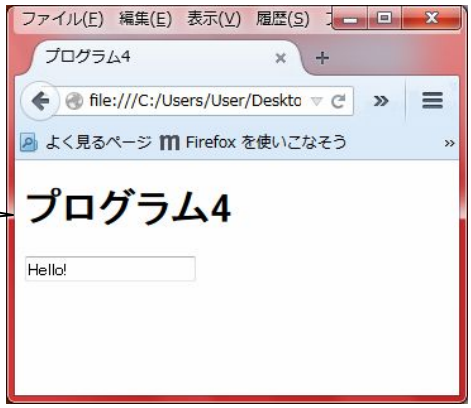
文字列中に HTML の
タグも使える

```
document.getElementById("area") .innerHTML = "Hello!";
```

1 行にまとめることもできる

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>プログラム 4</title>
  </head>
  <body>
    <h1> プログラム 4 </h1>
    <input type="text" id="area">
    <script src="prog4.js"></script>
  </body>
</html>
```

input 領域
に出力




```
var areap = document.getElementById("area");
areap.value = "Hello!";
```

```
document.getElementById("area").value = "Hello!";
```

1 行にまとめることもできる