

JavaScriptプログラミング入門

2. JavaScriptの概要

08T4067L

横田 翔

2-1 オブジェクトベース言語としての JavaScript

2-1-1 オブジェクト指向言語と オブジェクトベース言語

オブジェクト指向言語

- ・ 対象となるオブジェクトがどのようなデータ、操作方法を持っているかというようにモデル化してプログラミングを行う。
- ・ オブジェクト指向の概念の中でも基本的なものだけを採用していて、初心者優しい。⇒ オブジェクトベース

メソッド — オブジェクトに対する操作のこと。

継承 — オブジェクトのプロパティやメソッドを引き継ぎ、新たなオブジェクトを定義する機能。

2-1-2 オブジェクトの取り扱い

new 演算子 — オブジェクトの生成

書式 `new コンストラクタ`

コンストラクタ — オブジェクトを生成するために使用する特別なメソッド

※ コンストラクタには生成するオブジェクトの名前と同じ名前がつく。

生成されたオブジェクトには、オブジェクトそのものではなく、オブジェクトを指し示している値が格納される。

インスタンス — 生成されたオブジェクトの実体のこと。

メソッドの利用

- ・いったんオブジェクトを生成するとそのオブジェクトに用意されているさまざまなメソッドが使用可能に。

書式

インスタンス名. メソッド名()

たとえば、Dateオブジェクトには、
時間を取り出すgetHours()、
分を取り出すgetMinutes()といったメソッドがある。

例

```
var now ; //変数nowを宣言
now = new Date(); //現在の日付時間を管理するDateオブジェクトを作成
var hour, minute; //変数hour,minuteを宣言
hour = now.getHours(); //時を変数hourに入れる
minute = now.getMinutes(); //分を変数minuteに入れる
```

これを元に、Webブラウザの画面に現在時刻を表示するサンプル(time.html)を示す。

time.html

```
<html>
<head>
<meta http-equiv="Content-Script-Type" content="text/javascript">
<title>現在時刻を表示する</title>
</head>
<body bgcolor="#ffffff">
<script Language="JavaScript" type=text/javascript>
var now, hour, minute;
now = new Date();
hour = now.getHours();
minute = now.getMinutes(),
document.write("<h1>現在時刻:");
document.write(hour,"時",minute,"分","</h1>");
</script>

</body>
</html>
```

インスタンスメソッドとクラスメソッド

- ・getHours()のようなメソッドは、あらかじめ生成されているインスタンスに対して実行される。インスタンスが管理するデータは個々のインスタンスごとに異なるため、以下のようにオブジェクトの型名であるDateに対してメソッドを呼び出すことはできない。

```
hour = Date.getHours(); //エラー
```

インスタンスメソッド — 特定のインスタンスに対して実行するメソッド

➡ 実行するにはあらかじめインスタンスが存在している必要がある。

クラスメソッド — インスタンスに依存しないで汎用性が高い関数のように使用できる。

実行形式 ➡

オブジェクト名. メソッド

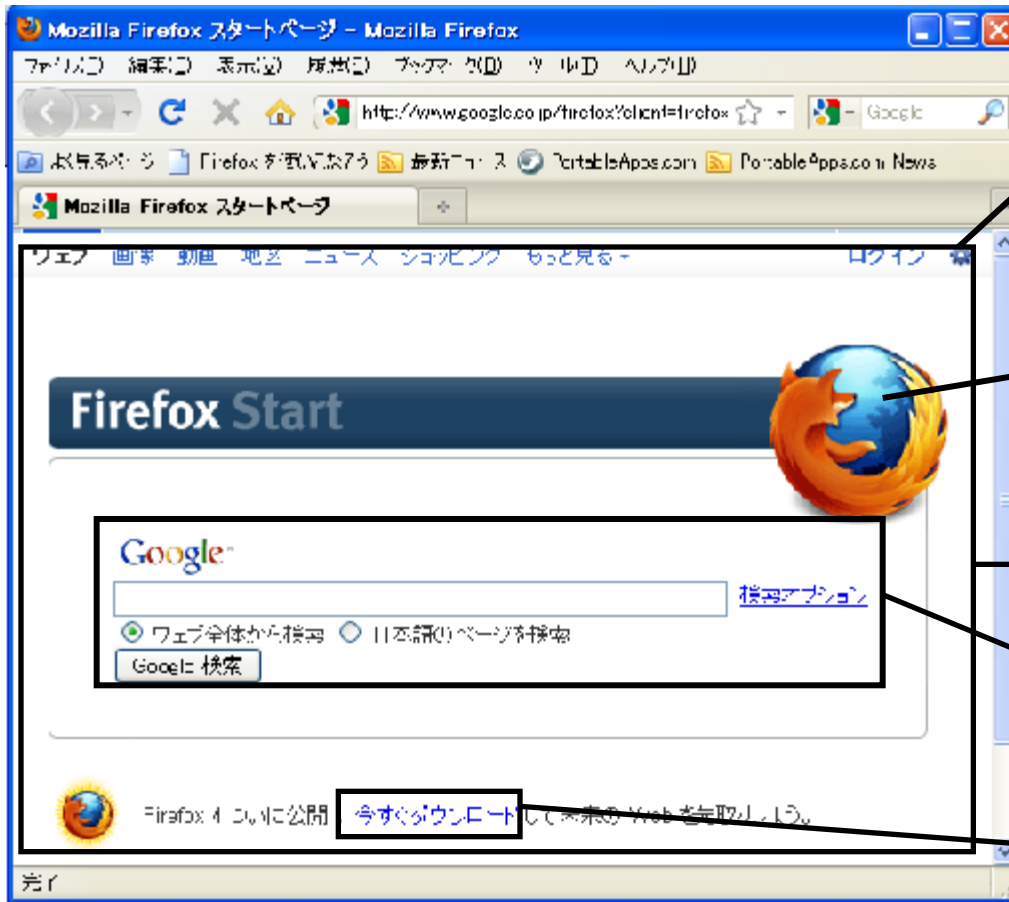
引数について

引数 — メソッドに渡す何らかの値。

引数を複数個とる場合もあり、その場合はカンマ「,」で区切って指定。

ちなみに引数がない場合でも()は必要。

2-1-3 Webブラウザ上のオブジェクト



windowオブジェクト

imageオブジェクト

documentオブジェクト

formオブジェクト

linkオブジェクト

Webブラウザ画面のオブジェクト

オブジェクト	説明
windowオブジェクト	Webブラウザのウィンドウ
documentオブジェクト	HTMLドキュメント
imageオブジェクト	イメージ
formオブジェクト	フォーム
linkオブジェクト	リンク

documentオブジェクトについて

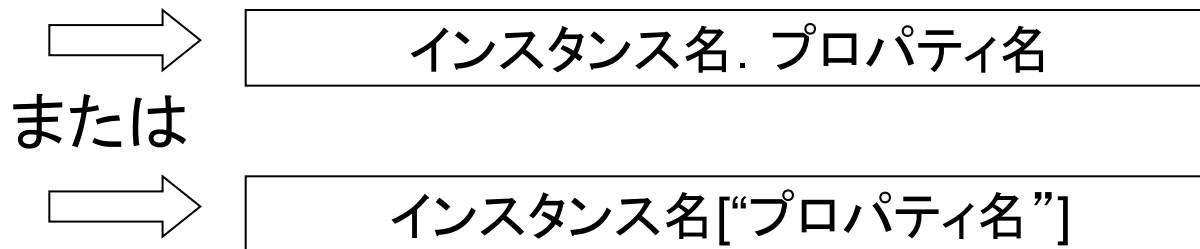
documentオブジェクト — Webブラウザに表示されているHTMLドキュメント自身。
プロパティとメソッドで構成される。

WebブラウザにHTMLドキュメントがロードされると、そのHTMLドキュメントを管理するオブジェクトのインスタンスである、「document」が自動的に生成される。

2-1-4 プロパティ

プロパティ — オブジェクトの外観や性質などを管理する値。

プロパティへのアクセス



プロパティには値を設定できるものと、読み出しのみ可能なものの2通りある。値を設定するときは以下のように記述。

インスタンス名. プロパティ名 = 値;

2-1-5 with文によるプロパティ、メソッドのアクセス

with文 — 最初にインスタンス名を宣言することで、with文内では、インスタンス名を省略してメソッドやプロパティを指定できる。

例

```
with(インスタンス名){  
    メソッド名();  
    メソッド名();  
    .....  
    ...プロパティ名.....  
}
```

2-2 変数とリテラル

2-2-1 変数を使うには

変数の宣言について

- ・型名の宣言の必要はない。
 - ⇒ ただし間違いを防ぐため、できるだけ宣言しておいたほうがいい。
- ・宣言しただけで値を代入していない変数には「undefined」という特殊な値が設定される。
- ・複数の変数を1行で宣言するに変数名をカンマ「,」で区切る。

2-2-2 変数の型

JavaScriptの代表的な変数は以下の4つ

- 数値型 (数値を表す)
- 文字列型 (文字列を格納)
- ブーリアン型 (trueかfalseのどちらかの値をとる)
- オブジェクト型 (オブジェクトのインスタスを格納する)

2-2-3 リファレンス型とプリミティブ型

リファレンス型

- ーオブジェクト型、配列など
- ー変数にオブジェクトの場所を指し示す値を格納される

プリミティブ型

- ー数値型、ブーリアン型
- ー変数に数値そのものが格納される

2-2-4 JavaScriptにおける変数名のつけ方

JavaScriptで使用する変数名の決まりごと

- ・JavaScriptの予約語を名前として使用できない
- ・変数名は、アルファベットかアンダースコア「_」で始まらなくてはならない
- ・2文字目以降には、数字またはアルファベットの中から任意の文字を使用できる

※大文字と小文字が区別されるので注意

2-2-5 リテラル

リテラル — プログラム中に記述する整数や文字列などの、そのままの値のこと

例: num = 2 ;

↑ ↑

変数 リテラル

リテラルの種類

・整数リテラル

— 整数値を表記する

— 10進数、8進数、16進数で表記できる(通常は10進数)

— 「0」から始まるリテラルは8進数

例: 013 ← 8進数のリテラル(10進数で13)

— 「0x」または「0X」から始まるリテラルは16進数

例: 0xff 16進数のリテラル(10進数で10)

0x0a 16進数のリテラル(10進数で10)

- ・浮動小数点リテラル

- 整数以外の数値

- 例: 3.14

- $1.3e3 \longleftarrow 1.3 \times 10^3$

- $.3 \longleftarrow 0.3$

- $1.666e-9 \longleftarrow 1.666 \times 10^{-9}$

- ・ブーリアンリテラル

- true (真) または false (偽) の論理値 (ブール値) を表す

- ・文字列リテラル

- 文字列をあらわすリテラル。ゼロ個以上の文字をダブルクォーテーション「”」もしくはシングルクォーテーション「'」で囲む。

・null

—何もない状態を表す特殊な値

・特殊な文字列の表記法

エスケープシーケンスによる特殊文字

エスケープシーケンス	内容
\\n	改行
\\t	タブ
\\b	バックスペース
\\r	キャリッジリターン
\\f	フォームフィールド(紙送り)
\\\\	文字「 \\ 」
\\'	シングルクォーテーション
\\“	ダブルクォーテーション
\\XXX	8進数で指定したASC II 文字コード
\\xXX	16進数で指定したASC II 文字コード
\\uXXXX	Unicode(ユニコード)で指定した文字コード

2-3 式と演算子

2-3-1 式の概要

- ・他のプログラミング言語同様、右辺に計算式を記述し、左辺にその値を代入する変数を記述する。

例:

```
num = num + 5 ;
```

//変数numの値に5を加え、再びnumに代入する

```
newStr = myName + "is taro." ;
```

//myNameという変数と「is taro.」という文字列をつなげた文字列を変数newStrに代入する

・式の組み合わせ

例:

$$b = (a = 3) * 2;$$

変数aに「3」が代入され、変数bには $3 * 2$ の結果「6」が代入される

・二項演算子と単項演算子

二項演算子

— 二つの値に対し演算を行う。

例: $i + 1$, x / y ,

単項演算子

— ひとつの値に対し演算を行う。

例: $++i$, $-b$,

2-3-2 さまざまな演算子

・算術演算子

— 数値型のデータに対し演算を行う。

●二項演算子

演算子	説明
+	加算
-	減算
*	乗算
/	除算
%	剰余算

●単項演算子

演算子	説明
-	マイナス符号
++	インクリメント
--	デクリメント

・等価比較演算子

一式の値が等しいか等しくないかを調べ、真偽の
ブール値を返す二項演算子

・比較演算子

一式の大小を比較して、真偽のブーリアン値を返す
二項演算子

●等価比較演算子

演算子	説明
==	等しければtrue、そうでなければfalseを返す
!=	等しくなければtrue、そうでなければfalseを返す

●比較演算子

演算子	例	説明
<	$a < b$	aの値がbより小さければtrue、そうでなければfalseを返す
>	$a > b$	aの値がbより大きければtrue、そうでなければfalseを返す
<=	$a \leq b$	aの値がb以下ならばtrue、そうでなければfalseを返す
>=	$a \geq b$	aの値がb以上ならばtrue、そうでなければfalseを返す

・論理演算子

ーブーリアン型のデータに対し論理演算を行う

●論理演算子

演算子	説明
!	否定:ブール値の値を反転する単項演算子
&&	論理積:左右の値がともにtrueならばtrueを そうでなければfalseを戻す二項演算子
!!	論理輪:左右の値のどちらかがtrueであればtrue そうでなければfalseを戻す二項演算子

・ビット演算子

— 数値を32ビットの整数とみなして、2進数に変換しビット演算を行う演算子

●ビット演算子

演算子	説明
&	論理積 (and)
	論理和 (or)
^	排他的論理和 (xor)

例: ●数値9と14の論理和、論理積、排他的論理和

演算	10進数	2進数
	9	1001
	14	1110
9 & 14 (論理積)	8	1000
9 14 (論理和)	15	1111
9 ^ 14 (排他的論理和)	7	0111

・シフト演算子

— 整数型のデータのビットを右方向、もしくは左方向にシフトする二項演算子

●シフト演算子

演算子	説明
<<	左シフト
>>	右シフト
>>>	右論理シフト

例: $a = 10 \ll 1$;

	2進数	10進数
	00001010	10
<<1	00001010	
(左へ1ビットシフト)	000010100	20

2-3-3 演算子の優先順位

優先順位	演算子
1	. [] ()
2	! ++ --
3	* / %
4	+ -
5	<< >> >>>
6	< <= > >=
7	== !=
8	&
9	^
10	
11	&&
12	
13	? :
14	= += -= /= %= &= ^= = <<= >>= >>>=

2-3-4 値の型を調べるtypeof演算子

例:

```
var today = new Date();  
var month = today.getMonth();  
var myNum = 15 ;  
var myName = "WildDog";
```

●上記リストの場合の例

typeof演算子	戻り値
typeof today	object
typeof Date	function
typeof myNum	number
typeof month	string
typeof true	boolean
typeof ng	undefined
typeof eval	function
typeof null	object

今週の課題

以下の条件を満たすプログラムを作成してください。

- ・new演算子を使用してDateオブジェクトを生成する
- ・変数hour,minuteを宣言して、それぞれに時間と分を代入する
- ・時間が6～12時なら「おはようございます」
12～18時なら「こんにちは」
18～24時なら「こんばんは」
0～6時なら「よい子は寝ている時間です」

と表示されるようにする

- ・使用した変数の型をtypeof演算子を用い、調べ表にまとめてみる