

Java Scriptプログラミング入門

3-6～3-7

茨城大学工学部情報工学科

08T4018Y 小幡智裕

3-6 組み込み関数

- ・組み込み関数とは、JavaScriptの内部にあらかじめ用意されている関数のこと。ユーザ定義の関数と同様に、関数名のみで呼び出すことができる。

3-6-1 文字列を式として評価する関数

・eval()関数

引数: string → 式として評価する文字列

戻り値: 式を評価した値

説明: 引数として渡された文字列stringを式として評価してその値を戻す。

例: var result;

```
result = "2 * 5 + 1";
```

変数resultには 2 * 5 + 1という文字列が代入される。

```
result = eval("2 * 5 + 1");
```

変数resultには計算結果の11が代入される。

3-6-2 文字列を数値に変換する

- parseInt()関数

引数: string(数値に変換する文字列)

radix(基数)

戻り値: 整数値

説明: 数値を示す文字列である引数stringを、引数radixを基数とする整数値に変換する。Radixを省略した場合、先頭が"0x"もしくは"0X"で始まる場合には16進数、"0"で始まる場合には8進数、それ以外の場合には10進数と判断される。変換できない場合、NaNが戻される。

3-6-2 文字列を数値に変換する(続き)

▪ `parseFloat()`関数

引数: `string`

戻り値: 数値

説明: 数値を示す文字列である引数`string`を、浮動召集点数値に変換する。指数表記も可能。数値に変換できない場合は`NaN`が戻される。

例: `parseFloat("3.14")` → 3.14

`parseFloat("1.4e3")` → 1400

`parseFloat("A1")` → NaN

NaN(非数)かどうかを調べる関数

- `isNaN(testValue)`

引数: `testValue` → 調べる値

戻り値: `true`または`false`

説明: 引数`testValue`が非数であるかどうかを調べ、非数なら`true`、そうでなければ`false`を戻す。ダイアログボックスやフォームに数値が入力されたかどうかを調べるためにしばしば使われる。

3-6-3 文字列をエンコード/デコードする

- `escape()`関数

引数: `string` → エンコードする文字列

戻り値: エンコードされた文字列

説明: 引数`string`内のスペースや日本語などの文字列をエンコードする。

- `unescape()`関数

引数: `string` → エンコードされた文字列

戻り値: デコードされた文字列

説明: `escape()`関数とは逆に、エンコードされた文字列をデコードする。

escape()関数/unescape()関数

例: `escape("K&K")` → `K%26K`

中央の文字「&」が「%26」に変換される。

`unescape("K%26K")` → `K&K`

エンコードされた文字がデコードされる。

URLエンコード形式でエンコードする

- `encodeURIComponent()`
エンコードする(予約文字を含まない)
- `encodeURIComponent()`
エンコードする(予約文字を含む)
- `decodeURI()`
デコードする(予約文字を含まない)
- `decodeURIComponent()`
デコードする(予約文字を含む)

URLエンコード形式でエンコードする(続き)

`encodeURI()`と`encodeURIComponent()`の違いは、

「`,` `/` `?` `:` `;` `@` `&` `=` `+` `$`」

といった予約文字を含めて変換するかしないかである。

`encodeURI()`は予約文字を変換せず、

`encodeURIComponent()`は予約文字を変換する。

3-7 正規表現オブジェクト

- ・正規表現オブジェクトは、JavaScriptのバージョン1.2以降から搭載された。正規表現オブジェクトとは、正規表現を扱うRegExpオブジェクトのことである。正規表現とは、文字列の検索、置換などを柔軟に行うための表記法である。

3-7-1 正規表現の基礎知識

・正規表現とメタキャラクタ

「先頭がJavaScriptで始まる行だけを検索したい」あるいは「先頭が二桁の数字で始まる行だけを検索したい」

→単純な検索機能ではこれらができない。そこで正規表現と呼ばれる表記法を用いる。

正規表現とメタキャラクタ

- ・具体例

先頭が「JavaScript」で始まる文字列は正規表現で、以下のよう
に表わされる。

`^JavaScript`

先頭の「^」というのは、特殊な意味を持つ記号で「メタキャラ
クタ」と呼ばれる。「^」は、先頭を表わす。

また、「JavaScript」、「Java Script」どちらかを検索したい場合は

`Java ?Script`

で表わされる。

「?」は「直前の文字の0回もしくは1回の繰り返し」を表わす。

正規表現オブジェクトの生成

- JavaScriptで正規表現を利用するには、正規表現のパターンをあらかじめ「正規表現オブジェクト」として生成する必要がある。生成は、RegExpコンストラクタの引数にパターンとして使用する文字列を指定することによって行う。

例:「^JavaScript」をパターンとする正規表現オブジェクトの生成

```
var regexp = new RegExp("^JavaScript");
```

もしくは

```
var regexp = /^JavaScript/;
```

test()メソッドによるパターンマッチ

- このメソッドは、RegExpオブジェクトにあるパターンマッチ用のメソッドの中でも最も基本的なメソッド。
- test()メソッド
 - 書式: 正規表現オブジェクト.test(str)
 - 引数: str → 検索対象の文字列
 - 戻り値: マッチした結果
 - 説明: 引数として与えられた文字列と、正規表現オブジェクトのパターンがマッチしているかどうかを調べ、マッチして入ればtrueを、そうでなければfalseを戻す。

test()メソッドを用いた例

- ・文字列「JavaScript入門」とパターン「^JavaScript」がマッチするか調べる。

var regexp = /^JavaScript/; 正規表現オブジェクト生成

var str = "JavaScript入門"; 検索対象の文字列

var result = regexp.test(str); パターンマッチ実行

この場合、パターンマッチは成功するので、変数resultはtrueとなる。簡略化して以下のようにすることもできる。

```
var result = /^JavaScript/.test("JavaScript入門");
```

正規表現オブジェクトのオプション

- ・オプション `i`
大文字/小文字を区別しない。
- ・オプション `g`
全てを検索する。

これらは、RegExpコンストラクタの2番目の引数として指定するか、/パターン/の後に指定する。

3-7-2 stringオブジェクトのメソッドによる パターンマッチ

▪ match()

書式: 文字列.match(regex)

引数: regex → 正規表現オブジェクト

戻り値: マッチした文字列

説明: 文字列と引数で指定したregexがマッチすれば、マッチした部分文字列を返し、マッチしなければnullを返す。

▪ search()

書式: 文字列.search(regex)

引数: regex → 正規表現オブジェクト

戻り値: マッチした文字列

説明: 文字列と引数で指定したregexがマッチすれば、マッチした文字列の先頭からの位置(先頭を0とする)を返す。見つからなければ-1を返す。

3-7-3 メタキヤラクタ

メタキヤラクタ	説明
^	行の先頭にマッチ
\$	行の終わりにマッチ
.	任意の1文字にマッチ
?	直前の文字の0回もしくは1回の繰り返しにマッチ
*	直前の文字の0回以上の繰り返しにマッチ
+	直前の文字の1階以上の繰り返しにマッチ
\d	数字にマッチ
\D	数字以外にマッチ
\s	空白文字にマッチ
\S	空白文字以外にマッチ
\w	英数字とアンダーバーにマッチ
\W	英数字とアンダーバー以外にマッチ
\b	単語の区切りにマッチ

文字クラス

- 「java」「cava」「sava」の全てにマッチさせたい場合には文字クラスと呼ばれる表記が便利である。文字クラスでは、「[」と「]」の間にマッチさせたい文字の並びを記述する。上記の例は、[jcs]avaと表す。

文字クラス	説明
[a-z]	アルファベットの小文字
[0-9]	数字(\dと同じ)
[a-zA-Z]	アルファベット全て
[a-zA-Z0-9_]	英数字と「_」(\wと同じ)
[^a-zA-Z0-9]	英数字と「_」以外(\Wと同じ)

3-7-4 正規表現による文字列の置換

- Stringオブジェクトのreplace()メソッドを使うと、正規表現によるパターンにマッチした文字列を別の文字列で置換できる。

- replace()

書式: 文字列.replace(regex, str)

引数: regex → 正規表現オブジェクト

str → 文字列

戻り値: 置換の文字列

説明: 1番目の引数で指定した正規表現regexにマッチした部分を、2番目の引数である文字列strに置換する。

3-7-5 マッチした文字列を後から使う

- ・パターンの一部を「(」と「)」で囲んでグループ化することでマッチした文字列を後から使うことができる。マッチした文字列は、Regexオブジェクトのスタティック変数「\$1」、「\$2」、「\$3」と順に格納される。

例:

```
var yourmail = makoto@peachpress.co.jp;
var mailRegex = /^([\w\.-]+)@([\w\.-]+)$/;
```

とすると、

Regex.\$1には、makoto

Regex.\$2には、peachpress.co.jpが格納される。

課題

- メールアドレス(学校のアカウント)を入力して、送信ボタンを押すと学籍番号とドメインに分かれた結果が出力されるプログラムを作る。
- 正規表現を用いて分ける。