

HTMLオブジェクト

HTMLタグオブジェクトは、Webページの構成要素であるHTMLタグをドリトル内で直接扱うことができるオブジェクトである。HTMLオブジェクトを使うことで、動的な画面更新が可能な双方向コンテンツを簡潔に記述できる。HTML+JavaScriptやPHPベースのCGIプログラムでは処理の混在によりプログラムの理解が難しいことがあるが、HTMLオブジェクトを使うことで可読性の高いプログラムを作成できる。

HTMLオブジェクトは、角括弧 (<>) を使って直接記述する。次のプログラムでは、 というHTMLの順序なしリスト [unordered list] を作成している。

```
メッセージボード<ul></ul>
```

また、入力欄やボタンなどのフォーム要素も同様に記述できる。

```
入力欄<input>  
送信ボタン<button>送信</button>
```

HTMLオブジェクトには、それぞれの要素に応じた様々なメソッドが用意されている。例えば、リスト要素には**クリア**や**いれる**、フォーム要素には**値?**などのメソッドがある。

```
メッセージボード!クリア。 // リストの内容をクリアする  
メッセージ<li></li> ""( "こんにちは" )テキスト // liタグに文字列を設定  
メッセージボード! (メッセージ)いれる // ulタグにliタグを追加  
入力された内容 = 入力欄!値?。 // 入力欄の値を取得
```

HTMLオブジェクトを使うと、**webapi**オブジェクトと組み合わせることで、サーバとの通信を行う双方向コンテンツを簡潔に記述できる。以下は、簡単なチャットプログラムの例である。

```
メッセージボード<ul></ul>  
入力欄<input>  
送信ボタン<button>送信</button>
```

```
受信_wa webapi!作る。  
受信_wa.url "read.php"
```

```
メッセージ表示 = 「 | メッセージ配列 |  
メッセージボード!クリア。  
メッセージ配列! 「 | 内容 |  
メッセージ<li></li> ""(内容)テキスト。  
メッセージボード! (メッセージ)いれる。  
」それぞれ実行。  

```

```
メッセージ表示! (受信_wa読む) 実行。
```

```
送信_wa webapi!作る。  
送信_wa.url "write.php"  
名前 = "ユーザー"。
```

```
送信ボタン: 動作 = 「  
送信_wa.内容 = 名前 + " : " + (入力欄!値?)。</pre>
```

```
送信_wa□読む。  
メッセージ表示！（受信_wa□読む）実行。  
□□
```

このプログラムでは、サーバ側で動作するPHPプログラム□read.phpとwrite.php□とクライアント側で動作するドリトルプログラムを組み合わせている。サーバ側のPHPプログラムは以下ようになる。

snippet.php

```
//掲示板にメッセージを書き込むwrite.php  
<?php  
  $message = $_GET['message'] . "\n";  
  file_put_contents("chat.txt", $message, FILE_APPEND);  
?>  
  
//掲示板の書き込み一覧を取得するread.php  
<?php  
  $message = file("chat.txt");  
  echo json_encode($message);  
?>
```

この例では□WebAPIによるサーバ側とクライアント側で処理を分離している。クライアント側のドリトルプログラムではHTMLタグを直接記述し、サーバ側のPHPプログラムはWebAPIの形でモジュール化されている。このように処理を分離することで、各処理の責任範囲が明確になり、プログラムの可読性が高くなる。

HTMLオブジェクトは、タグの種類に応じて様々なメソッドを持つ。一般的なものとしては以下がある。

- **テキスト** - タグの内容のテキストを設定する
- **いれる** - 子要素としてHTMLタグを追加する
- **クリア** - 内容をクリアする
- **値?** - フォーム要素の値を取得する□input, textareaなど）
- **値** - フォーム要素の値を設定する

また□HTMLオブジェクトはイベント処理も行える。例えば、ボタンタグには動作メソッドを定義することでクリック時の処理を記述できる。

```
送信ボタン：動作 = 「  
// ボタンがクリックされたときの処理  
□□
```

HTMLオブジェクトを使った双方向コンテンツの開発では、クライアント側とサーバ側の処理を明確に分けることで、プログラムの構造が理解しやすくなる。クライアント側では主にユーザインターフェースと画面更新処理を記述し、サーバ側ではデータの保存や取得などの処理を記述する。これにより、プログラムの保守性や拡張性が向上する。

From:

<https://dolittle.eplang.jp/> - プログラミング言語「ドリトル」

Permanent link:

https://dolittle.eplang.jp/ch_html_js?rev=1741328089

Last update: **2025/03/07 15:14**

