

PHP に対してデータを送る役割は **FORM 要素**ですが、この要素そのものが持つ属性が WEBアプリケーションにとって重要である事は言うまでもありません。

まず、 属性によってサーバへのデータの送り方が大きく分けて2通りに分けられ、この属性を省略すると実行される通信方法を仮に『**アドレスバー方式**』と呼ぶことにします。この方式では、データは **URL の一部**として作成され、その URLそのものがサーバに到達します。この際、データ部分は と呼ばれ、データとデータの間は **&** で区切られます。

つまり、その URL は A 要素の 属性で指定可能なもので、データ部分が外部に容易に露出する事になるデータ通信方法です。

それに反してもう一つの通信方法は、ブラウザが直接サーバへデータを送る方法で、『**アドレスバー方式**』との大きな違いは送信可能なデータ量に対する制限が無い事です。この場合、FORM 要素の属性に **enctype** を使用して、値に を指定するとファイルのアップロードが可能になります。また、送信されたデータがアドレスバーに残る事も無く、確実にサーバへデータを送る事ができます。

そして、次に重要な属性は です。この属性を省略すると、データの送り先は現在表示している URL に対して送られます。つまり、省略せずに**他の PHP ファイルを指定すれば、その PHP にデータを送信可能**になります。こうする事によって、PHP で記述する処理を単純化する事が可能になり、プログラミングのコストを抑える可能性も大きくなります。

さらに、プログラムの処理そのものにはさほど影響は与えませんが、 属性を使用する事によって、**ブラウザ上のどの部分に表示するか**を決める事ができます。例えば 要素をページ内に埋め込んで、PHP の送信結果をその中に表示する事が可能になります。また、この属性に値として を指定すると、送信のたびに新しいタブを開いてそのウインドウ内に結果が表示されるでしょう。

ここまでは、画面定義として決められた基本機能の指定方法ですが、WEB アプリケーションとして完成度を高くするには、このデータ送信機能を実行する前に**入力チェックを行う必要があります**。たとえ入力チェックが必要無い簡単な送信でも、ユーザの間違いを避ける為に確認ダイアログを表示して**送信のキャンセルをする事ができるようにする必要があります**。

その為の重要な FORM 要素の属性が です。このような属性は、他の要素でも同様に、頭に **on** が付き、JavaScript への橋渡しをするための属性です。よって、属性の値は JavaScript そのものが記述されます。

そして、FORM 要素のこの属性は送信のキャンセルを実現する為に最初に ステートメントを指定して、他で定義した関数の戻り値をシステムに返す事になっています。この他で定義した関数が返す値が の場合にサーバへの送信はキャンセルされます。

```
function check(){  
    if ( !confirm("更新してもよろしいですか?") ){  
        return false;  
    }  
    return true;  
}  
</script>
```

confirm メソッドは、 オブジェクトのメソッドで、OK と キャンセルのボタンを持っており、ユーザが選択できます。JavaScript では、このトップに存在するオブジェクトは省略してメソッドのみを記述する事となっています。