

# ライブラリ

## ▼ユーティリティライブラリ

sample08/lib フォルダを作成し、その中に下記ファイルを作成しましょう。

util.php

```
<?php
// XSS対策のためのHTMLエスケープ
function es($data){
    // $dataが配列のとき
    if (is_array($data)){
        // 再帰呼び出し
        return array_map(__METHOD__, $data);
    } else {
        // HTMLエスケープを行う
        return htmlspecialchars($data, ENT_QUOTES, 'UTF-8');
    }
}

// 配列の文字エンコードのチェックを行う
function cken(array $data){
    $result = true;
    foreach ($data as $key => $value) {
        if (is_array($value)){
            // 含まれている値が配列のとき文字列に連結する
            $value = implode("", $value);
        }
        if (!mb_check_encoding($value)){
            // 文字エンコードが一致しないとき
            $result = false;
            // foreachでの走査をブレイクする
            break;
        }
    }
    return $result;
}
// ?>
```

## ▼エスケープ処理チェックプログラム

sample08 フォルダ内に下記ファイルを作成しましょう。

esSample.php

```
<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="utf-8">
  <title>XSS対策 es()</title>
</head>
<body>
<?php
// util.phpを読み込む
require_once("lib/util.php");
// HTMLタグの入ったデータを用意する
$myCode = "<script>alert('hello')</script>";
// es()でHTMLエスケープして表示する
echo '$myCodeの値：', es($myCode);
?>
</body>
</html>
```

## ▼文字エンコードチェックプログラム

sample08 フォルダ内に下記ファイルを作成しましょう。

ckenSample.php

```
!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="utf-8">
  <title>不正なエンコーディングによる攻撃対策 cken()</title>
</head>
<body>
<?php
// util.phpを読み込む
require_once("lib/util.php");
// Shift-JISのデータを用意する
$utf8_string = "こんにちは。";
$sjis_string = mb_convert_encoding($utf8_string, 'Shift-JIS');
// 内部エンコーディングを調べる
$encoding = mb_internal_encoding();
// cken()でチェックする
if (cken([$sjis_string])) {
  echo '配列の値は、', $encoding, 'です。';
} else {
  echo '配列の値は、', $encoding, 'ではありません。';
}
?>
</body>
</html>
```