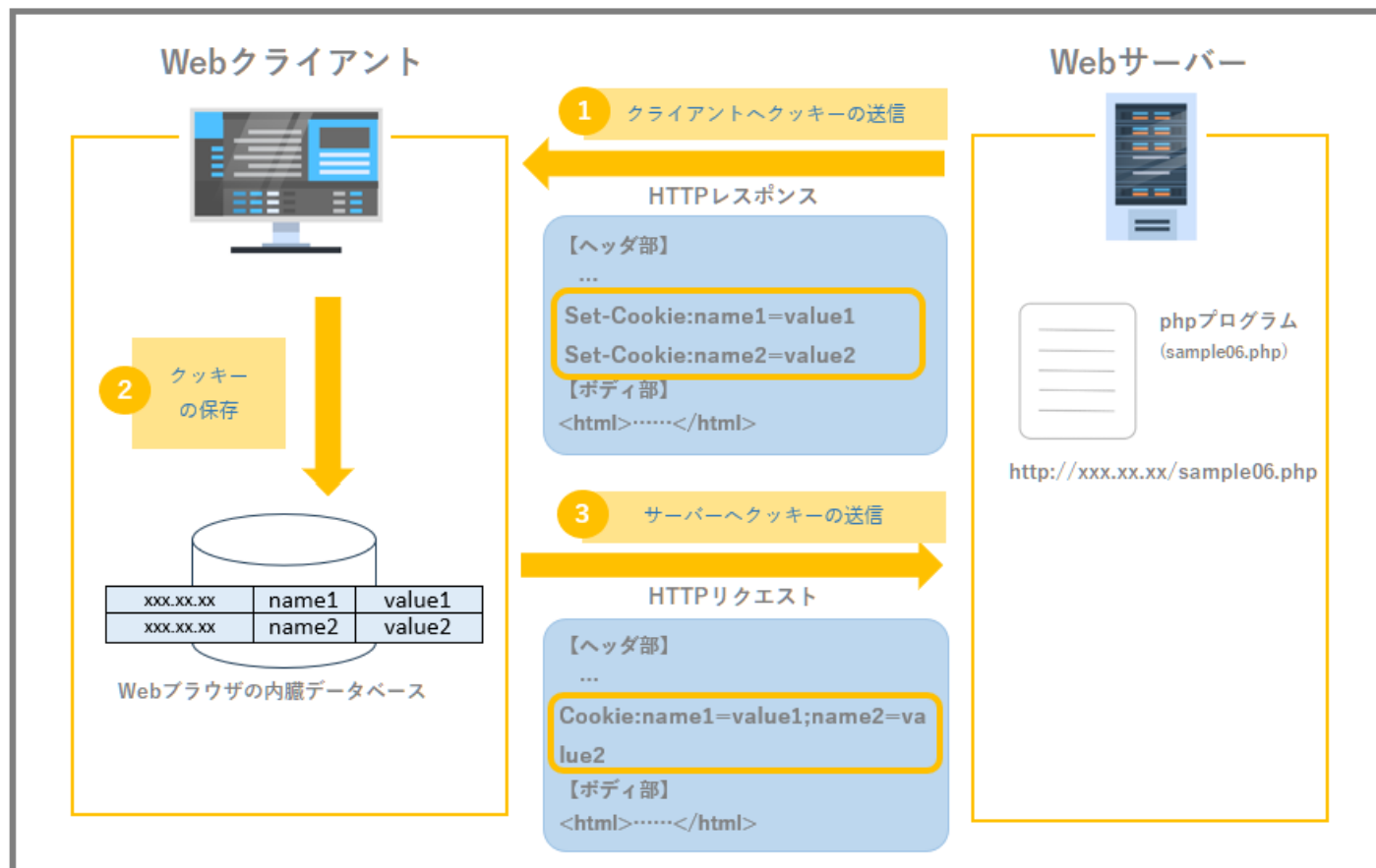


第6回 クッキー・セッション

6-1. Cookie (クッキー)

クッキーとは、サーバー上のデータをブラウザに保存することで、同じ情報を再利用するための仕組みです。



クッキーは、Web ブラウザの設定でクライアント PC への書き込みをブロックすることができます。また、クッキーの中身をクライアント PC 上で書き換えることも可能ですので、パスワードなど重要な情報をクッキーに保存するのは避けましょう。

■Cookie の注意点



- ・ 容量の大きい情報を入れない
- ・ 重要な情報を入れない
- ・ 内容が書き換えられると困る情報を入れない

■ Cookie の仕様と制限

項目	説明
保存ファイル	ブラウザごとにテキストファイルで保存される
Cookie の全容量	300 個が上限 ※300 個を超えると古い Cookie から削除される。
1 サイトの制限数	20 個が上限 ※上限個数を超えると保存できない。ブラウザによって上限が異なる
1 つの Cookie の容量	4KB ※4KB を超えた部分は削除される
有効期限	保存時に有効期限を設定 ※有効期限が未設定の場合は、ブラウザを閉じた時に削除される。
ブラウザ間	ブラウザごとに保存されるため Cookie の共有はできない

■Cookie の保存

■setcookie関数

```
setcookie( name, value, expires )
```

戻り値: 成功→true、失敗→false

※ただし、戻り値がtrueであっても、クッキーがブラウザに保存されたかまでは分らない。

■構文

name	クッキー名
value	クッキー値
expires	クッキーの有効期限 UNIX タイムで指定する UNIX タイム（1970 年 1 月 1 日からの経過秒数）での指定となるため、time 関数と組み合わせて使用します。

■time 関数

戻り値：現在時刻を UNIX タイム (1970 年 1 月 1 日 00:00:00) からの通算秒として返します。

■例

```
time() + 60 /* 有効期限 1 分 */
```

```
time() + ( 60 * 60 * 24 ) /* 有効期限 1 日 */
```

■SampleCode (Cookie をセット)

```
<?php
//変数 test のデータを「sample04」というクッキー名で保存

$test = "あいうえお";

setcookie("sample04", $test, time() + (60 * 1));

?>
```

■Cookie の読み込み

\$_COOKIE

HTTP COOKIE 変数

現在のスクリプトに HTTP クッキーから渡された変数の連想配列です。

■SampleCode (Cookie を読み込み)

```
<?php
//「sample04」という名前のクッキーの値を、変数 cookieData に格納

$cookieData = $_COOKIE["sample04"];

?>
```

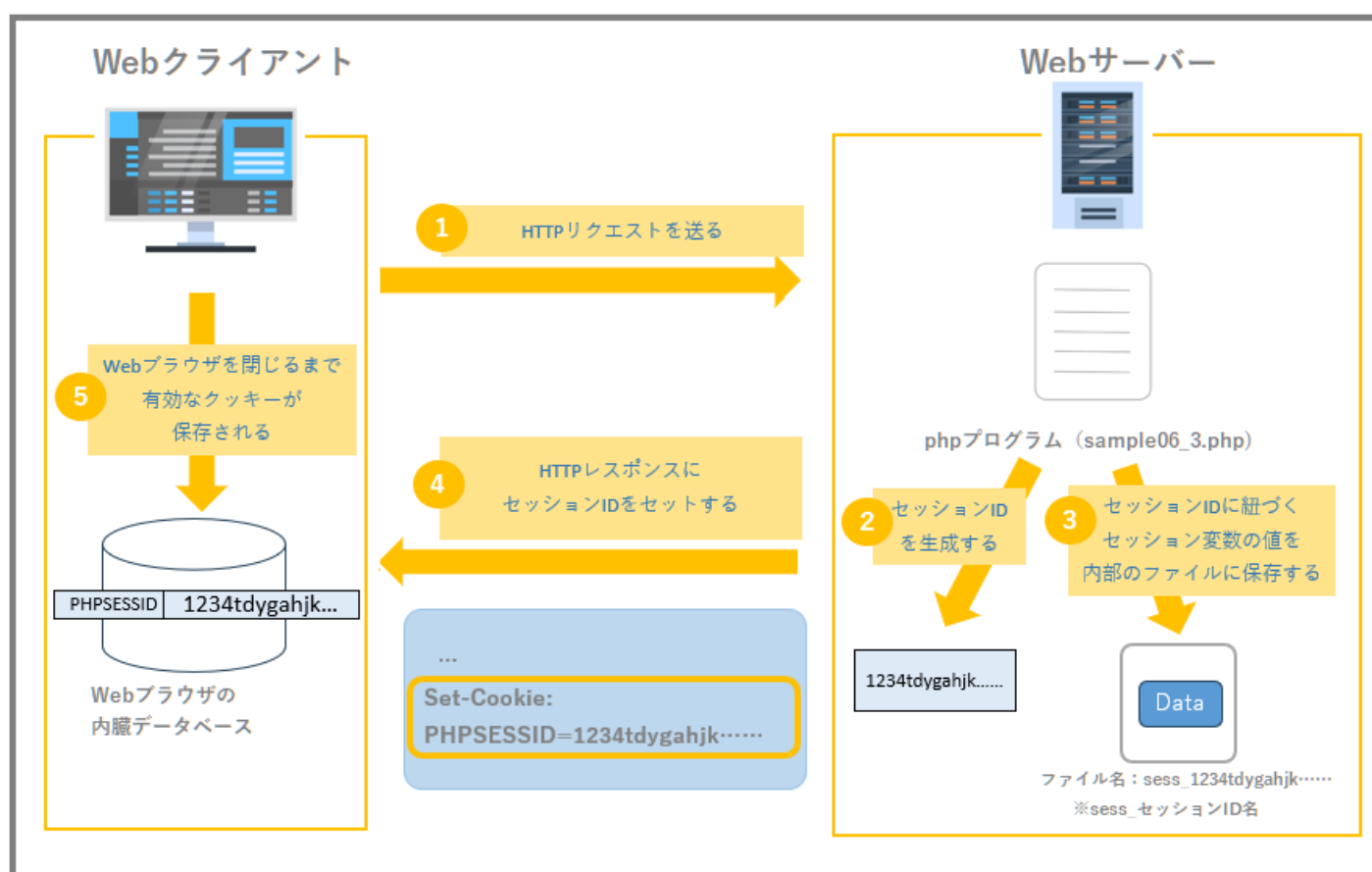
6-2. セッション

セッションとは、ユーザーが Web サイトにアクセスし、ページ遷移し、ブラウザを閉じるまでの期間、値を保持できる特殊な変数です。

クッキーとは異なり、セッション変数のデータは、サーバーに保存されます。

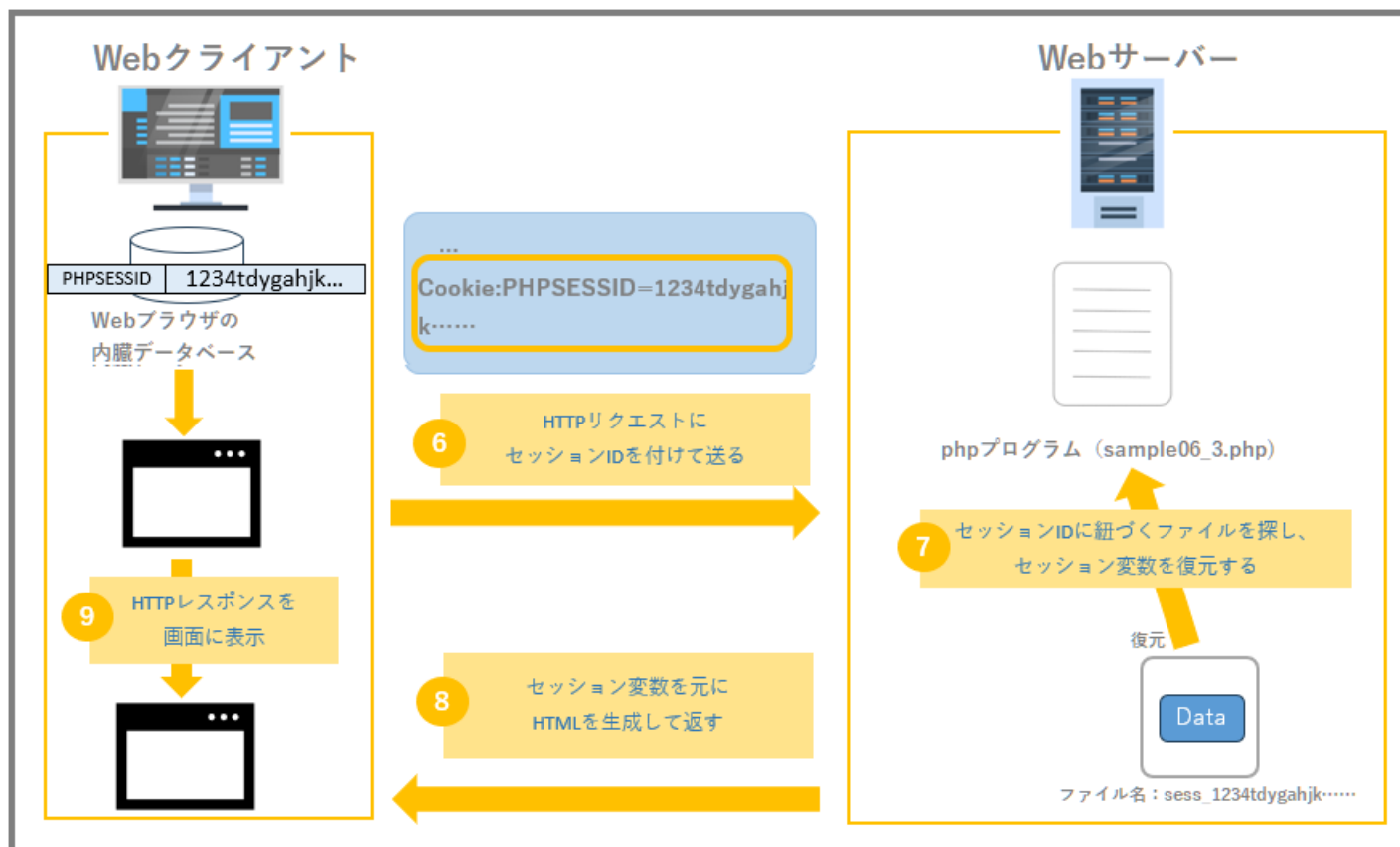
セッション変数は、セッションクッキー（ブラウザを閉じるまで有効なクッキー）を内部的に使用します。

■セッション ID 生成時



※セッション ID 生成後のページ遷移時の図は、次ページにあります。

■ ページ遷移時



■ Session 変数の注意点



Session 変数のデータは、閲覧ユーザーごとに作られます。

【破棄のタイミング】

- ・ そのユーザーが Web ブラウザを閉じたとき
- ・ そのユーザーが Web ブラウザを開いたまま放置して、一定時間が経過したとき

Session を扱うための関数が用意されています。

■Session の開始

セッションを開始するための関数です。**セッション変数を利用するすべてのページの冒頭でコール**する必要があります。

■session_start関数

```
session_start( )
```

戻り値: 成功→true、失敗→false

※セッションIDが未生成であれば、新たなものを作成して、セッションクッキーを送出

※セッションIDが生成済であれば、クライアントから受け取ったセッションIDを元に変数\$_SESSIONを復元

■Session の確認

現在のセッションの状態を取得するための関数です。セッションの状態を表す定義済み定数が用意されています。

■session_status関数

```
session_status( )
```

戻り値:

定数名	説明
PHP_SESSION_DISABLED	セッションが無効な状態。
PHP_SESSION_NONE	セッションが有効だが、セッションが存在しない状態。
PHP_SESSION_ACTIVE	セッションが有効で、既にセッションが存在する状態。

■SessionID の取得（または設定）

現在のセッション ID を取得または、設定するための関数です。

■session_id関数

```
session_id(id )
```

戻り値: 成功→現在のセッションID(存在しない場合は空文字)
失敗→false

※id が指定された場合(null 以外)は、現在のセッションID を置き換える。

※セッションID を置き換える場合は、session_id をsession_start より先にコールしている必要がある。

※セッションID として利用できる文字は、[a-zA-Z0-9,-] に限定されている。

■引数

id	指定され、null でない場合、現在の セッション ID を置換する。 (id を置換する場合、この関数は、session_start 関数より先にコールしている必要があります。)
----	---

■Session データの破棄

セッションに登録されているすべてのデータを破棄するための関数です。現在のセッションに関連付けられているすべてのデータを破棄します。

■session_destroy関数

```
session_destroy( )
```

戻り値: 成功→true、失敗→false

※この関数は、セッションに関するグローバル変数を破棄しません。また、セッションクッキーを破棄しません。

■Session の読み込み

<code>\$_SESSION</code>	セッション変数 現在のスクリプトで利用できるセッション変数を含む連想配列です。
-------------------------	--

■SampleCode （セッション変数に値を格納）

```
<?php
//セッション変数の配列に「old」というキーで値を格納

session_start();

$_SESSION["old"] = "かきくけこ";

?>
```

■SampleCode （セッション変数の値を取得）

```
<?php
//セッション変数「old」に値があれば、ローカル変数$oldData に値を格納

session_start();
if (isset($_SESSION["old"])) {
    $oldData = $_SESSION["old"];
}

?>
```