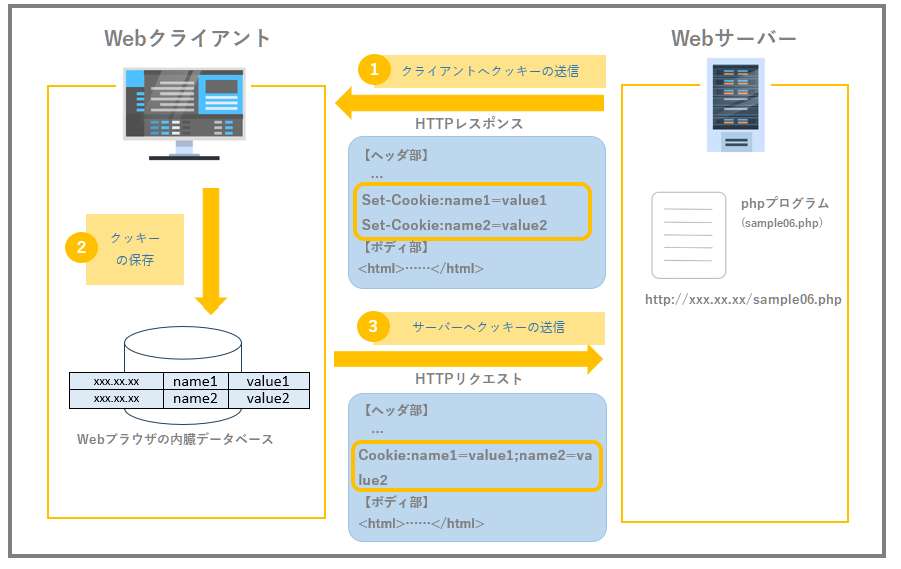
第６回　クッキー・セッション

６－１．Cookie（クッキー）

　クッキーとは、Webサイト閲覧時に端末上のWebブラウザに保存されるテキストファイルのことである。Webサイトを閲覧する際に「Cookie取得に同意しますか」という表示を目にしたことがある人もいるだろう。Webサーバーが発行・配付し、クライアント端末のWebブラウザが受け取る仕組みである。

Cookieを用いることで、IDやパスワードなどの認証情報がWebブラウザに保管されるため、スムーズな認証が可能となる。しかし一方で、Cookieをめぐるセキュリティやプライバシーの懸念についても、近年は注目されるようになっている。



　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。

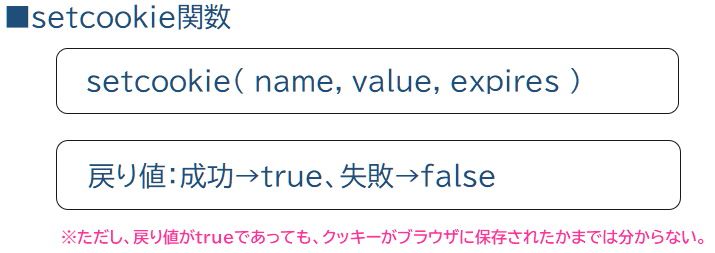
■Cookieの注意点

|  |  |
| --- | --- |
| 警告 単色塗りつぶし | ・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



■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"];

?>

■一つのクッキーに複数の値を格納する

サーバーからクライアントに送ることができるクッキーは１つだけではない。クッキー名を違う名前で指定すれば、複数の値をクッキーに保存することができる。

ただし、その数は制限があるので、多くの値をクッキーに保存したい場合は工夫が必要となる。

複数の値をクッキー値に保存する場合、配列に格納して保存することもできる。

また、クッキー値の取得は、$\_COOKIEでもできるが、課題４，５，で使用したfilter\_input関数を使用したほうが安全である。

今回の課題は、仕様の内容的に配列を使用したほうが簡潔に書ける。

<<例>>sample\_cookie.php　クッキー値を配列に格納する

<?php

//クッキー値を配列で格納する

setcookie ("buy[1]", "Televison",time() + ( 60 \* 1 ) );

setcookie ("buy[2]", "Video",time() + ( 60 \* 1 ) );

setcookie ("buy[3]", "Audio",time() + ( 60 \* 1 ));

//配列で格納されたクッキー値を取得する。

//FILTER\_DEFAULT・・・・・フィルターチェックをしない（なくてもOK）

//FILTER\_REQUIRE\_ARRAY・・・・・・・配列でない場合、falseを返す

$cookie\_array = filter\_input (INPUT\_COOKIE,"buy", FILTER\_DEFAULT,FILTER\_REQUIRE\_ARRAY);

//配列でないクッキー値のセットと取得。

setcookie ("buy2","apple",time() + ( 60 \* 1 ));

$cookie=filter\_input (INPUT\_COOKIE,"buy2", FILTER\_DEFAULT);

echo "<pre>";

var\_dump($cookie\_array);

echo "</pre>";

echo $cookie."<br>";

?>

<html>

<head><title>PHP TEST</title></head>

<body>

<?php

//クッキー配列が存在する場合はその内容を表示し、存在しなければ、値をセットしましたと表示する。

if (isset($cookie\_array)){

print('<p>');

print('購入したものは下記の通りです<br>');

foreach($cookie\_array as $v){

print($v.'<br>');

}

print('</p>');

}else{

print('<p>値をセットしました</p>');

}

?>

</body>

</html>

**<<参考　Chromeでのクッキー値参照方法>>**

Chromeの開発者ツールから

1.開発者タブを開く

開き方

右クリックから検証 or **F12**

2.Applicationタブを開く

3.左バーのCookiesから各Cookieの詳細の確認が可能

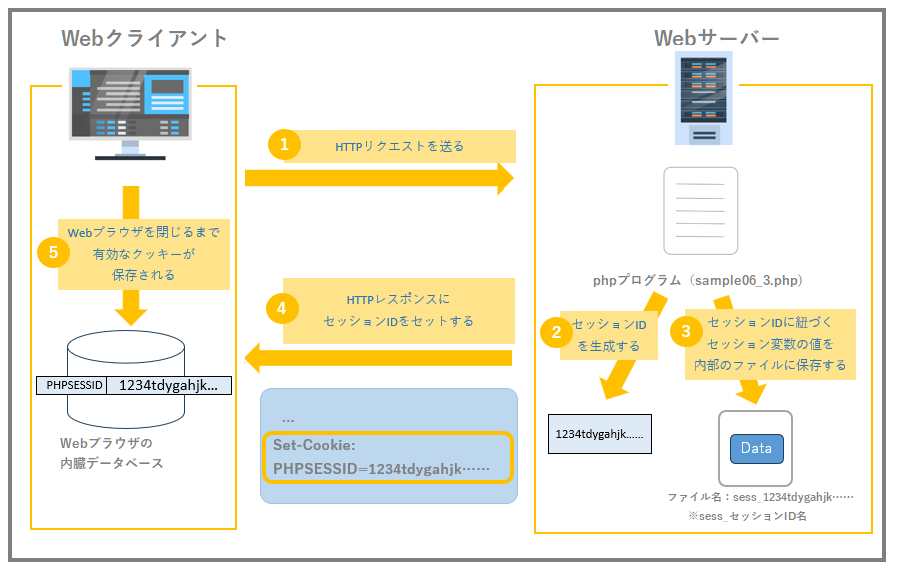
６－２．セッション

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。（ショッピングサイトのカートなど）

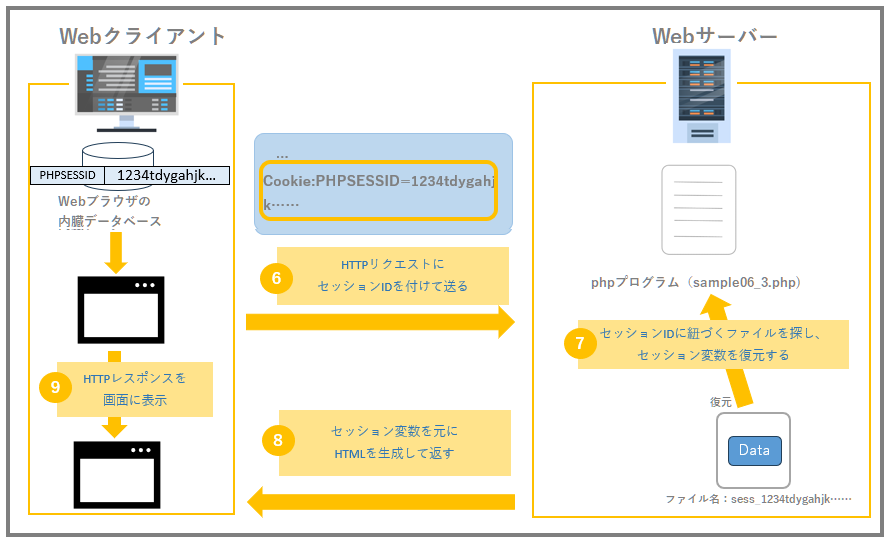
クッキーとは異なり、セッション変数のデータは、サーバーに保存されます。

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



■ページ遷移時



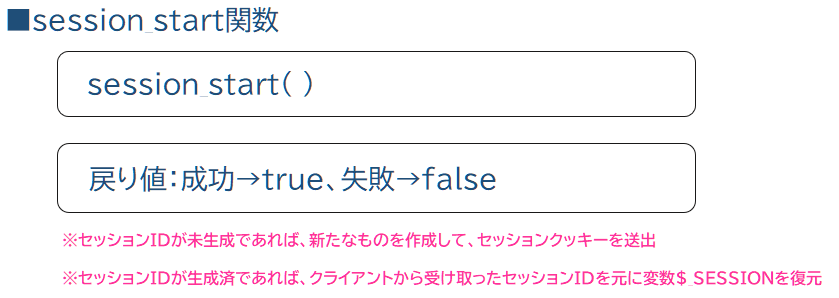
■Session変数の注意点

|  |  |
| --- | --- |
| 警告 単色塗りつぶし | **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 |

PHPでは、Sessionを扱うための関数が用意されています。

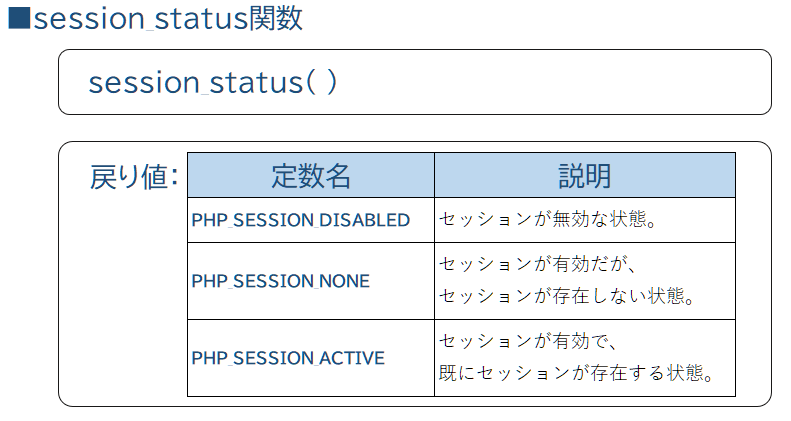
■Sessionの開始

　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。



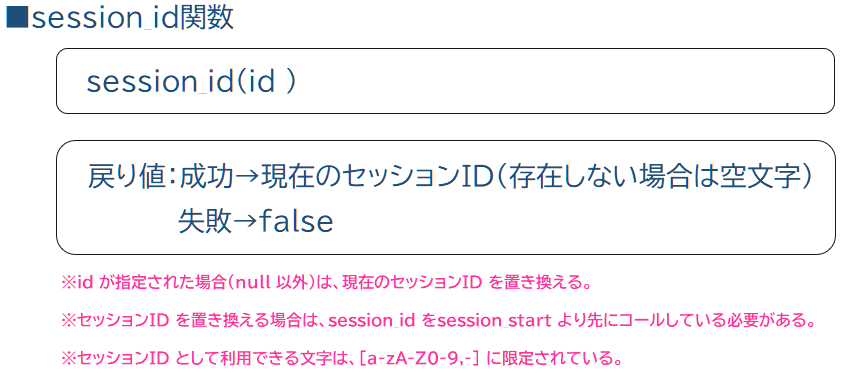
■Sessionの確認

　　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。



■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。

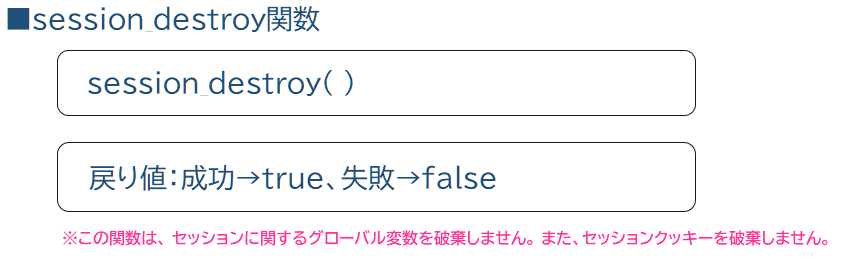


■引数

|  |  |
| --- | --- |
| 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されているすべてのデータを破棄します。



セッションは、ブラウザを閉じれば破棄されるが、その前に明示的にセッションを破棄したい（ログアウトしたときなど）場合に上記関数を使用する。ただし、セッション情報を格納したクッキー情報までは破棄されないので、session\_destroyを行う前に、該当クッキーの値を破棄しておくことも忘れないようにする。

**＜＜セッションデータの一部の内容を破棄したい＞＞**

$\_SESSION変数は、連想配列なので複数の値を持たせることができる。その中で、ある一部のセッション変数を破棄したい場合は、変数を破棄するunset関数を使う。(下記プログラム参照)

$\_SESSION内のすべての値を破棄したい場合は、

$\_SESSION = array();

のように空配列を代入すればよい。（$\_SESSIONそのものをunsetしないよう注意）

■Sessionの読み込み

|  |  |
| --- | --- |
| $\_SESSION | セッション変数  現在のスクリプトで使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"];

//セッション変数「old」の破棄

unset($\_SESSION["old"]);

}

?>