PHP Cookies

cookie in PHP is a small file with a maximum size of 4KB that the web server stores on the client computer. They are typically used to keep track of information such as a username that the site can retrieve to personalize the page when the user visits the website next time. A cookie can only be read from the domain that it has been issued from. Cookies are usually set in an HTTP header but JavaScript can also set a cookie directly on a browser.

Setting Cookie In PHP: To set a cookie in PHP, the **setcookie()** function is used. The setcookie() function needs to be called prior to any output generated by the script otherwise the cookie will not be set.

Syntax:

```
setcookie(name, value, expire, path, domain, security);
```

Parameters: The setcookie() function requires six arguments in general which are:

- **Name:** It is used to set the name of the cookie.
- **Value:** It is used to set the value of the cookie.
- **Expire:** It is used to set the expiry timestamp of the cookie after which the cookie can't be accessed.
- **Path:** It is used to specify the path on the server for which the cookie will be available.
- **Domain:** It is used to specify the domain for which the cookie is available.
- **Security:** It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

  Below are some operations that can be performed on Cookies in PHP:

- **Creating Cookies**: Creating a cookie named Auction_Item and assigning the value Luxury Car to it. The cookie will expire after 2 days(2 days * 24 hours * 60 mins * 60 seconds).

Example: This example describes the creation of the cookie in PHP.

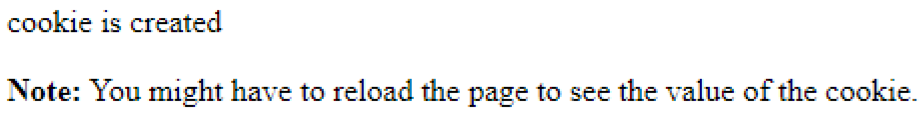
```
<!DOCTYPE html>
<?php
        setcookie("Auction_Item", "Luxury Car", time() + 2 * 24 * 60 * 60);
?>
<html>
<body>
```

```
    <?php
        echo "cookie is created."
    ?>
    <p>
        <strong>Note:</strong>
        You might have to reload the
        page to see the value of the cookie.
    </p>

</body>
</html>
```

Note: Only the name argument in the setcookie() function is mandatory. To skip an argument, the argument can be replaced by an empty string(“”).

Output:



cookie is created

Note: You might have to reload the page to see the value of the cookie.

Cookie creation in PHP

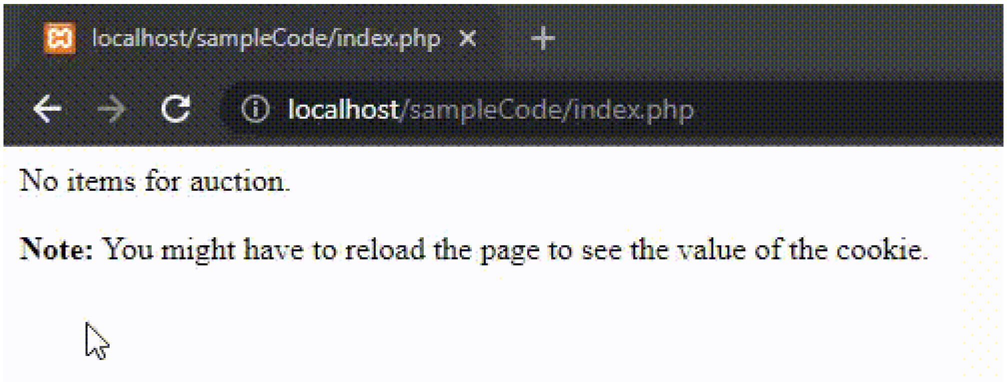
Checking Whether a Cookie Is Set Or Not: It is always advisable to check whether a cookie is set or not before accessing its value. Therefore to check whether a cookie is set or not, the PHP isset() function is used. To check whether a cookie “Auction_Item” is set or not, the isset() function is executed as follows:

Example: This example describes checking whether the cookie is set or not.

```
<!DOCTYPE html>
<?php
    setcookie("Auction_Item", "Luxury Car", time() + 2 * 24 * 60 * 60);
?>
<html>
<body>
    <?php
    if (isset($_COOKIE["Auction_Item"]))
    {
        echo "Auction Item is a " . $_COOKIE["Auction_Item"];
        }
        else
        {
                echo "No items for auction.";
        }
        ?>
        <p>
                <strong>Note:</strong>
                You might have to reload the page
                to see the value of the cookie.
        </p>

</body>
</html>
```

Output:


localhost/sampleCode/index.php
localhost/sampleCode/index.php

```
No items for auction.

Note: You might have to reload the page to see the value of the cookie.
```

Checking for the cookie to be set

Accessing Cookie Values: For accessing a cookie value, the PHP $_COOKIE superglobal variable is used. It is an associative array that contains a record of all the cookies values sent by the browser in the current request. The records are stored as a list where the cookie name is used as the key. To access a cookie named “Auction_Item”, the following code can be executed.

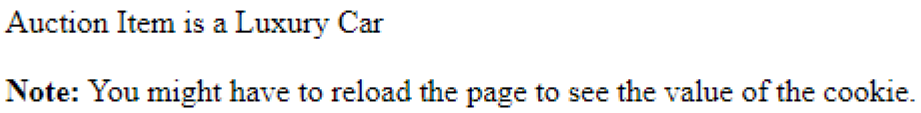
Example: This example describes accessing & modifying the cookie value.

```
<!DOCTYPE html>
<?php
        setcookie("Auction_Item", "Luxury Car", time() + 2 * 24 * 60 * 60);
?>
<html>
```

```
<body>
<?php
    echo "Auction Item is a " . $_COOKIE["Auction_Item"];
?>
    <p>
        <strong>Note:</strong>
        You might have to reload the page
        to see the value of the cookie.
    </p>

</body>
</html>
```

Output:



```
Auction Item is a Luxury Car

Note: You might have to reload the page to see the value of the cookie.
```

Accessing the Cookie value

Deleting Cookies: The setcookie() function can be used to delete a cookie. For deleting a cookie, the setcookie() function is called by passing the cookie name and other arguments or empty strings but however this time, the expiration date is required to be set in the past. To delete a cookie named "Auction_Item", the following code can be executed.

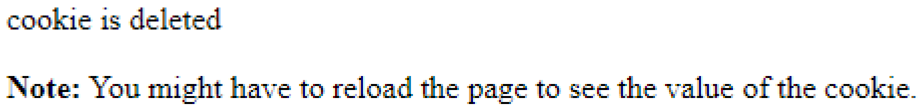
Example: This example describes the deletion of the cookie value.

```
<!DOCTYPE html>
<?php
    setcookie("Auction_Item", "Luxury Car", time() + 2 * 24 * 60 * 60);
?>
<html>
<body>
    <?php
    setcookie("Auction_Item", "", time() - 60);
```

```
        ?>
        <?php
            echo "cookie is deleted"
        ?>
        <p>
            <strong>Note:</strong>
            You might have to reload the page
            to see the value of the cookie.
        </p>

    </body>
    </html>
```

Output:



```
cookie is deleted

Note: You might have to reload the page to see the value of the cookie.
```

Deleting the Cookie

Important Points:

- If the expiration time of the cookie is set to 0 or omitted, the cookie will expire at the end of the session i.e. when the browser closes.
- The same path, domain, and other arguments should be passed that were used to create the cookie in order to ensure that the correct cookie is deleted.