

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

متغیر Cookie

مدرس : مهندس افشین رفوآ

متغیر Cookie

کوکی ها به شما اجازه می دهند اطلاعات کاربر را در صفحه ی وب ذخیره کنید.

کوکی ها اغلب برای تشخیص هویت کاربر، بکار می روند.

cookie چیست؟

کوکی ها داده یا اطلاعاتی هستند که در فایل های مبتنی بر متن، بر روی رایانه ی کاربر ذخیره می گردند.

cookie متغیری است که در رایانه ی کاربر ذخیره می شود و هر زمان که مرورگر رایانه ی نام برده، درخواستی

را به سرور می دهنده (**server**) ارسال کرد، **cookie** نیز همراه با آن فرستاده خواهد شد. با زبان جاوا

اسکریپت این قابلیت وجود دارد که یک کوکی ایجاد و همچنین بازیابی کرد.

هنگامی که وب سرور، یک صفحه ی وب را به مرورگر کاربر یا سرور می گیرنده ارسال کرد، اتصال کاملاً قطع

گشته و سرور تمامی اطلاعات مربوط به کاربر را فراموش می کند.

Cookie برای چه منظور اختراع شد؟

در اصل کوکی برای بیاد آوردن اطلاعات کاربر طراحی شد:

هنگامی که یک کاربر از صفحه ی وب بازدید می کند، اسم کاربر در یک کوکی ذخیره می گردد.

دفعه ی بعد که کاربر از صفحه مذکور بازدید می کند، **cookie** اسم وی را بخاطر می آورد.

کوکی ها به صورت جفت های اسم و مقدار (**name/value pair**) ذخیره می شوند:

```
username=John Doe
```

هنگامی که مرورگر کاربر از سرویس دهنده (**server**) یک صفحه وب را درخواست می کند، کوکی های متعلق به صفحه نیز به **request** اضافه می شوند. از این طریق سرور داده های لازم برای بیاد آوردن اطلاعات کاربر را بدست می آورد.

ایجاد کوکی با استفاده از جاوا اسکریپت

جاوا اسکریپت قادر است با استفاده از خاصیت **document.cookie**، کوکی هایی ایجاد کرده، آن ها را بخواند و نیز در صورت لزوم کوکی ها را حذف کند.

نحوه ی ایجاد کوکی:

```
document.cookie="username=John Doe";
```

همچنین می توانید یک تاریخ انقضا اضافه کنید. به صورت پیش فرض، هنگامی که مرورگر بسته می شود، کوکی نیز حذف می گردد:

```
document.cookie="username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC";
```

به وسیله ی پارامتر **path**، می توان به مرورگر اعلام کرد (اطلاع داد) کوکی به کدام آدرس (**path**) تعلق دارد. به صورت پیش فرض، کوکی به صفحه جاری تعلق دارد.

```
document.cookie="username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC;  
path="/";
```

خواندن کوکی به وسیله ی جاوا اسکریپت

به منظور خواندن یک کوکی از دستور زیر استفاده می کنیم:

```
var x = document.cookie;
```

تغییر دادن کوکی

می توانید یک کوکی را به همان روشی که ایجاد کردید، تغییر دهید:

```
document.cookie="username=John Smith; expires=Thu, 18 Dec 2013 12:00:00 UTC; path="/;
```

کوکی قبل بازنویسی (overwrite) می شود.

پاک کردن یک کوکی

حذف کوکی امر بسیار ساده ای است. برای این منظور پارامتر `expire` را به تاریخ ارسال شده، متصل کنید:

```
document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00 UTC";
```

Cookie String

اگرچه خاصیت `document.cookie`، شبیه به یک رشته ی متنی (`text string`) به نظر می رسد، اما در حقیقت این خاصیت یک رشته نیست.

حتی اگر هم یک `cookie string` کامل را در خاصیت `document.cookie` بنویسید، بازهم وقتی آن را می خوانید، مشاهده خواهید کرد که تنها جفت اسم-مقدار آن قابل رویت می باشد.

در صورت مقداردهی یک کوکی جدید، لازم است بدانید کوکی های قدیمی بازنویسی نمی شوند، بدین معنا که کوکی جدید به `document.cookie` اضافه می گردد. بنابراین اگر `document.cookie` را دوباره بخوانید، نتیجه ای مشابه نمونه ی زیر را دریافت خواهید کرد:

```
cookie1=value; cookie2=value;
```

اگر می خواهید مقدار یک کوکی معین را بدست بیاورید، باید یک تابع جاوا اسکریپت بنویسید که در `cookie string` بدنبال مقدار `cookie` مورد نظر بگردد.

مثال ایجاد یک کوکی

در مثال پیش رو یک کوکی ایجاد می کنیم که اسم بازدید کننده را در خود ذخیره می کند.

اولین باری که یک بازدید کننده به صفحه ی وب مراجعه می کند، وی یک پیغام **welcome** دریافت می کند.

برای این مثال، سه تابع (**function**) تعریف می کنیم:

1. یک تابع که مقدار کوکی را تنظیم می کند
2. یک تابع که مقدار کوکی را بازیابی می کند
3. یک تابع که مقدار کوکی را بررسی می کند

یک تابع که کوکی را مقداردهی می کند

ابتدا یک تابع ایجاد می کنیم که اسم بازدیدکننده را در متغیر کوکی ذخیره می کند:

```
function setCookie(cname, cvalue, exdays) {  
    var d = new Date();  
    d.setTime(d.getTime() + (exdays*24*60*60*1000));  
    var expires = "expires="+d.toUTCString();  
    document.cookie = cname + "=" + cvalue + ";" + expires;  
}
```

شرح مثال:

پارامترهای مثال فوق عبارتند از: اسم کوکی (**cname**)، مقدار کوکی (**cvalue**) و در نهایت تعداد روزهایی که پس از سپری شدن آن ها کوکی مربوطه منقضی می شود (**exdays**).

تابع با اضافه کردن اسم کوکی، مقدار کوکی و رشته ی **expire** به هم، یک کوکی را مقداردهی می کند.

تابع بازیابی کوکی

سپس یک تابع ایجاد می کنیم که مقدار کوکی مشخص شده را بازگردانی نماید:

```
function getCookie(c_name) {  
    var c_value = document.cookie;  
    var c_start = c_value.indexOf(" " + c_name + "=");  
    if (c_start == -1) {  
        c_start = c_value.indexOf(c_name + "=");  
    }
```

```

}
if (c_start == -1) {
    c_value = null;
}
else {
    c_start = c_value.indexOf("=", c_start) + 1;
    var c_end = c_value.indexOf(";", c_start);
    if (c_end == -1) {
        c_end = c_value.length;
    }
    c_value = unescape(c_value.substring(c_start, c_end));
}
return c_value;
}
}

```

در مثال فوق، به وسیله ی متد `indexOf()` داخل `cookie string`، نام `cookie` مورد نظر را جستجو می کنیم (`c_name`: نام `cookie`).

در متد `indexOf()` اولی، چنانچه مقدار (" `" + c_name + "="`") در `cookie string` یافت نشد، در تابع `indexOf()` داخل `if` مقدار ("`c_name + "="`") جستجو می شود (بدون " `" + c_name + "="`") بدین ترتیب در نهایت مکان شروع نام `cookie` در متغیر `c_start` ذخیره می شود.

در ادامه، در صورتی که مقدار متغیر `c_start` مخالف "`-1`" باشد، محل شروع و پایان مقدار `cookie` را بدست می آوریم و با استفاده از تابع `substring()` آنرا بازیابی می کنیم.

تعریف تابعی برای بررسی (چک کردن) کوکی

سر انجام به ایجاد تابعی می پردازیم که بررسی می کند آیا کوکی مقداردهی شده است خیر.

چنانچه کوکی تنظیم شده بود، یک پیغام خوش آمد گویی نمایش داده می شود.

اگر `cookie` مقداردهی نشده بود پنجره ی `Prompt` پدیدار شده و نام کاربر سوال می شود و با فراخوانی و

اجرای تابع `setCookie()` نام کاربر در `cookie` ذخیره می گردد.

```

function checkCookie() {
    var username=getCookie("username");
    if (username!="") {
        alert("Welcome again " + username);
    }else{

```

```

username = prompt("Please enter your name:", "");
if (username != "" && username != null) {
    setCookie("username", username, 365);
}
}
}

```

تنظیم کوکی برای تمامی موارد ذکر شده:

```

function setCookie(cname, cvalue, exdays) {
    var d = new Date();
    d.setTime(d.getTime() + (exdays*24*60*60*1000));
    var expires = "expires=" + d.toUTCString();
    document.cookie = cname + "=" + cvalue + ";" + expires;
}

```

```

function getCookie(cname) {
    var name = cname + "=";
    var ca = document.cookie.split(';');
    for(var i=0; i<ca.length; i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1);
        if (c.indexOf(name) == 0) return c.substring(name.length, c.length);
    }
    return "";
}

```

```

function checkCookie() {
    var user = getCookie("username");
    if (user != "") {
        alert("Welcome again " + user);
    } else {
        user = prompt("Please enter your name:", "");
        if (user != "" && user != null) {
            setCookie("username", user, 365);
        }
    }
}

```

مثال فوق تابع (`checkCookie()`) را پس از اتمام بارگذاری صفحه، راه اندازی می کند.

Tahlildadeh.com