

Cookies & Privacy

Overview

- What is a Cookie? Basic Facts
- Working with Cookies: Code & Demo
- Cookie based Marketing
- Cookies, Privacy & Legislation
- Conclusion

“Google has been bypassing the privacy settings of millions of people using Apple’s Safari web browser on their iPhones and computers - Tracking the web-browsing habits of people who intended for that kind of monitoring to be blocked.” Wall Street Journal, Feb. 17, 2012

“Google to pay \$22.5 million fine for Safari privacy evasion”, CNN Money, July 11, 2012
[largest penalty ever levied on a single company by the FTC]

Copyright 2002 by Randy Glasbergen. www.glasbergen.com



**“Someone got my Social Security number off the internet
and stole my identity. Thank God — *I hated being me!*”**

Copyright 2012-2020 Ellis Horowitz

What is a Cookie?

- Short pieces of text generated during web activity and stored in the user's machine by the user's web browser for future reference
- Cookies are created by website authors who write software for reading and writing cookies
- Cookies were initially used so websites would remember that a user had visited before, allowing customization of sites without need for repeating preferences
- The official 1997 specification for cookies, from Bell Labs, Lucent & Netscape, can be found at:
<https://tools.ietf.org/html/rfc2109>

Elements of a Cookie

- A cookie is associated with a website's domain and contains: **name**, **value**, **path**, and **expiration date**
- For example here is one that was placed on my machine from `research.google.com`
 - Name: `_utma`
 - Content: `180832036.353394603.1325873813.1325873813.1329750652.2`
 - Domain: `.research.google.com`
 - Path: `/`
 - Created: `Monday, February 20, 2012 7:12:45 AM`
 - Expires: `Wednesday, February 19, 2014 7:12:45 AM`
- Such cookies are sometimes referred to as HTTP cookies because they are placed there using the HTTP protocol as the delivery mechanism

Cookie Scope: What They Can Do

- **Store** and manipulate any **information** you explicitly provide to a site
- **Track** your interaction with the **site** such as pages visited, time of visits, number of visits
- Use any information available to the web server including: your IP address, Operating System, Browser Type

Cookie Scope: What They Cannot Do

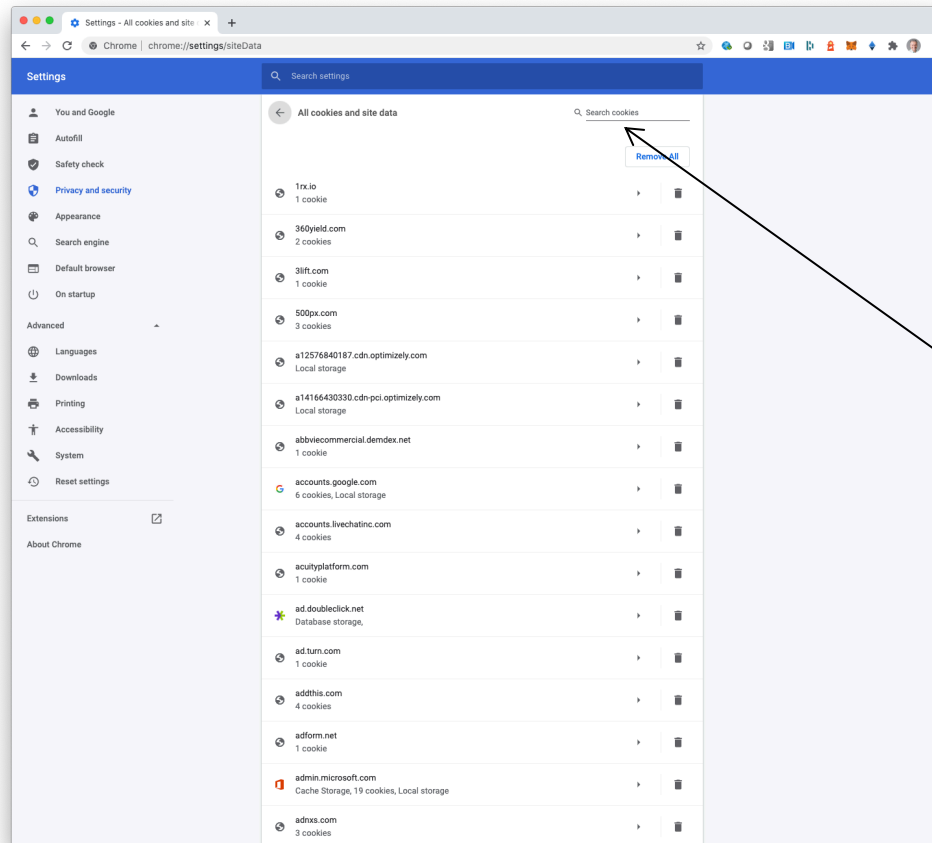
- Have automatic access to Personal Identifiable Information (PII) like name, address, email
- Read or write data to disk
- Read or write information in cookies placed by other sites
- Run programs on your computer
- As a result they
 - Cannot carry viruses
 - Cannot install malware on the host computer

Finding Cookies in Your Chrome Browser

Browsers store their cookies in their own format and files.

In **Chrome**

Customize > Settings > Privacy and security > Cookies and site data -> See all cookies and site data > Search cookies



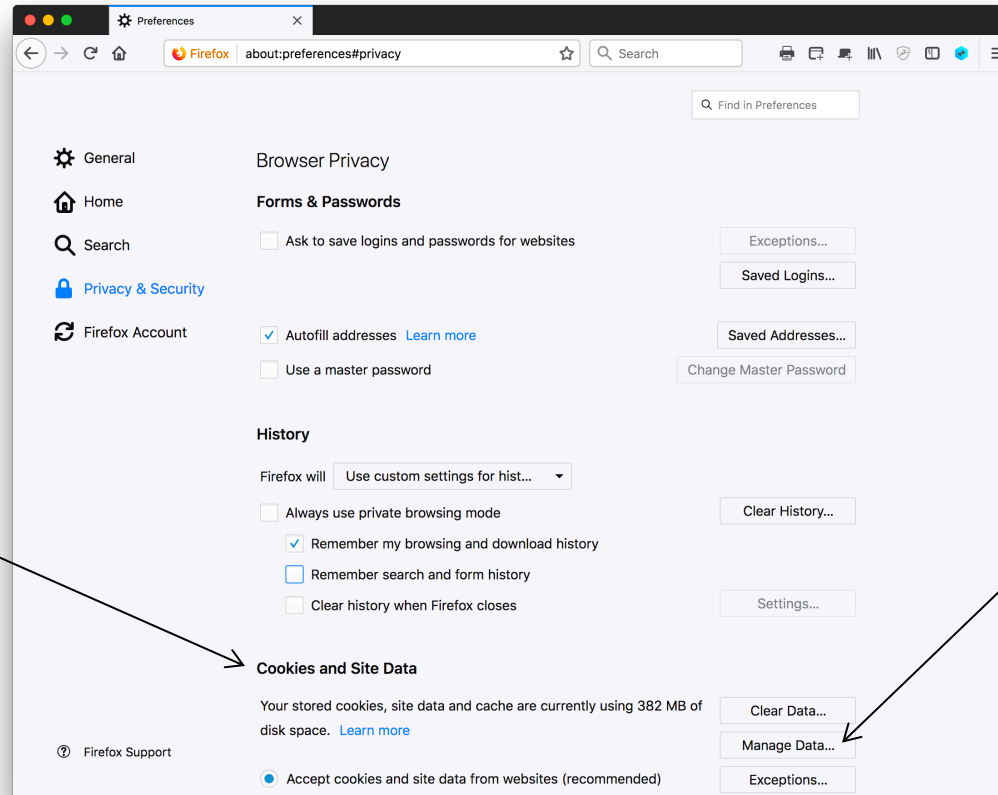
Search cookies

Finding Cookies in Your Firefox Browser

In **Firefox**

Open menu > Preferences > Privacy & Security > Browser Privacy > Cookies and Site Data > Manage Data... > Search websites

Here are the
Firefox settings
for handling
cookies

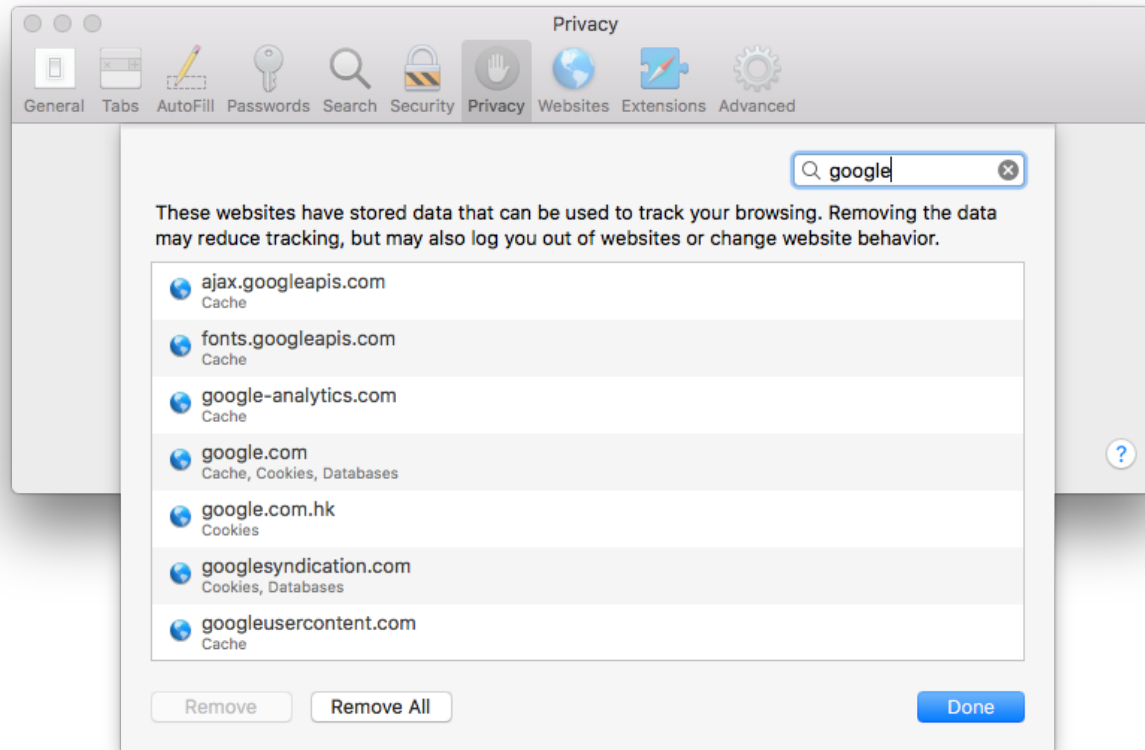


Manage Data...

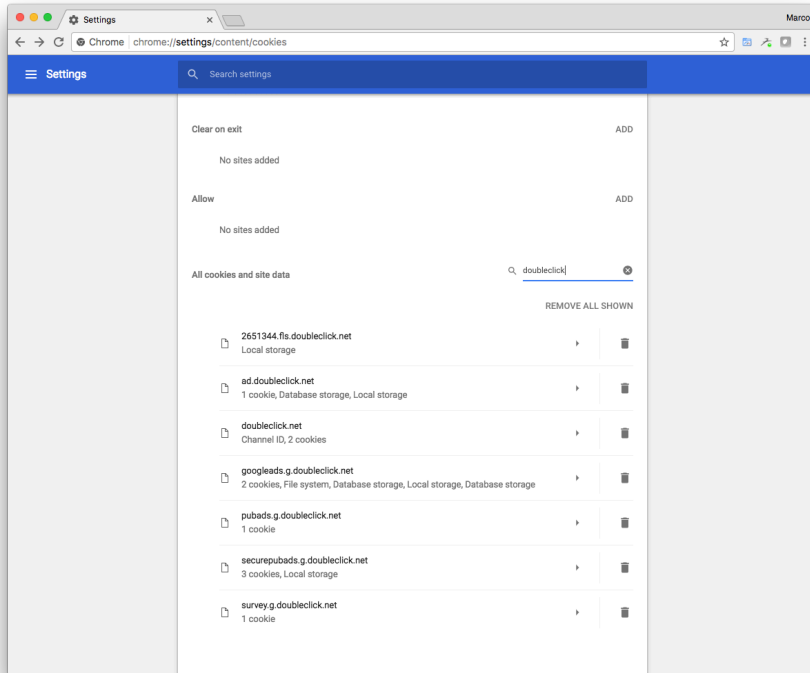
Finding Cookies in Your Safari Browser

In **Safari**

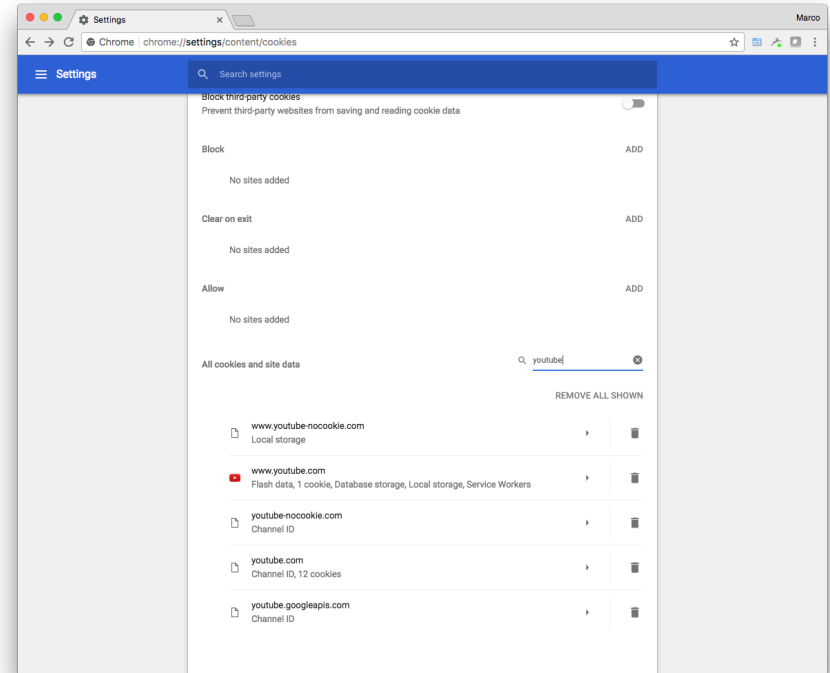
Safari menu > Preferences... > Privacy > Manage Website
Data... > Search



Other Google Property Cookies



DoubleClick Cookies



YouTube cookies

DoubleClick and YouTube are two companies owned and operated by Google; Doubleclick cookies are referred to as **3rd party cookies** because the user never actually visits the Doubleclick site

Sample Chrome Cookie Storage

For example, **google.com** has stored many cookies on this browser:

plus.google.com

plusone.google.com

productforums.google.com

research.google.com

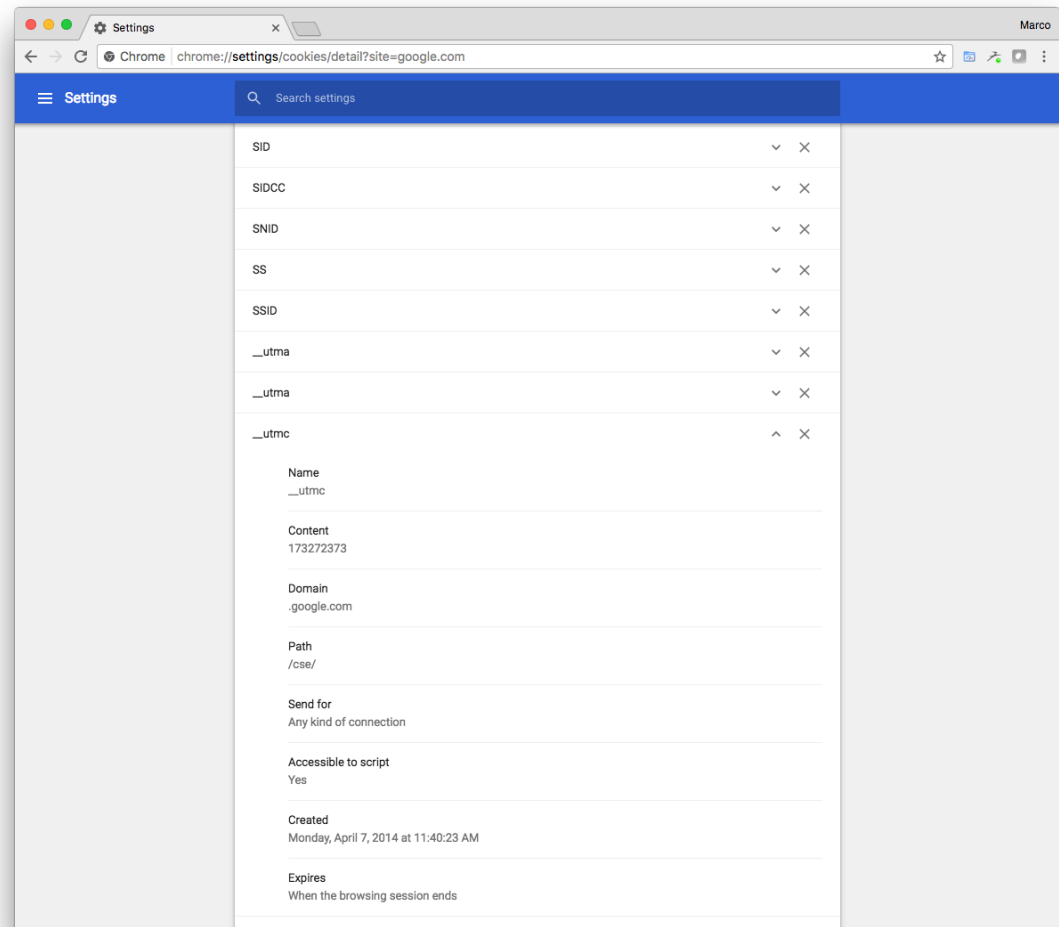
support.google.com

talkgadget.google.com

wallet.google.com

www.google.com

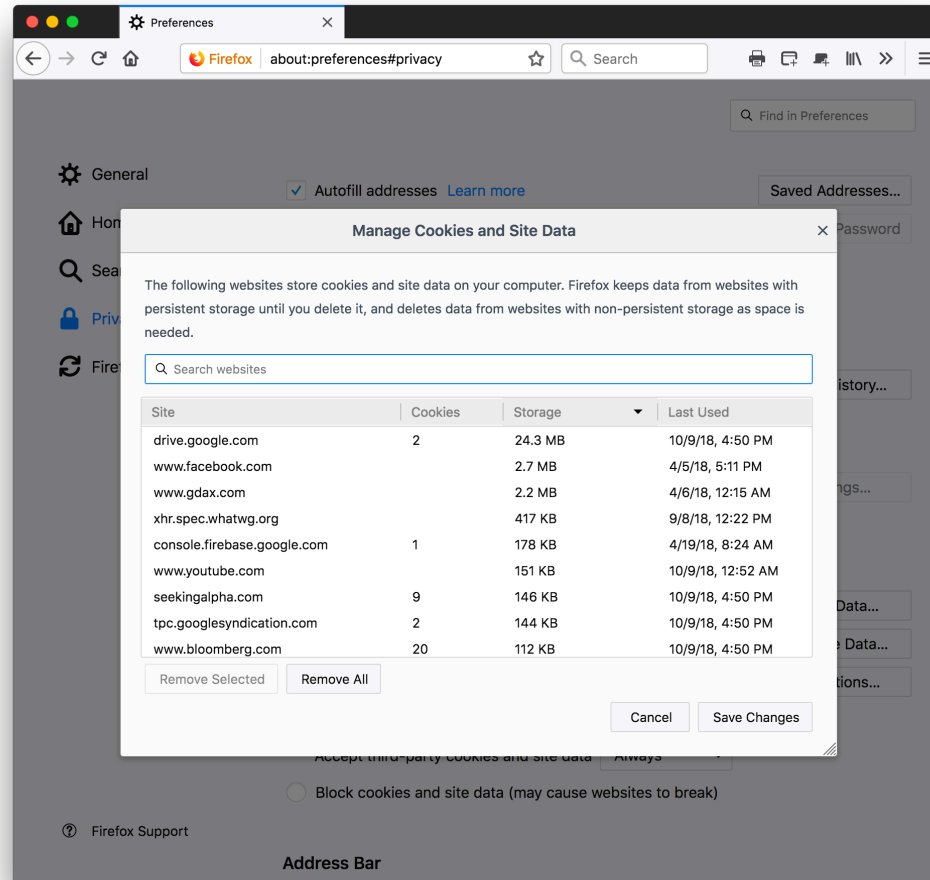
And many other



Sample Firefox Cookie Storage

For example, google.com has stored many cookies on this browser:

accounts.google.com
google.com
mail.google.com
plus.google.com
plusone.google.com
etc.



Cookie Types and Taxonomy

- **By Lifespan**
 - *Session* Cookies (stored in RAM)
 - *Persistent* Cookies (stored on disk)
- **By Read-Write Mechanism**
 - *Server-Side* Cookies (included in HTTP Headers)
 - *Client-Side* Cookies (manipulated with JavaScript)
- **By Structure**
 - *Simple* Cookies
 - *Array* Cookies
- **Session cookies** exist only while the user is reading and navigating the website; browsers normally delete session cookies when the user exits the browser
- **Persistent cookies**, also known as tracking cookies have an expiration date
- **Secure cookies** have the secure attribute enabled and are only used via **https**, so the cookie is always encrypted
- **Third-party cookies** are not from the “visited” site

Third Party Cookies

- **Third party cookies** are cookies set with a different domain than the one in the browser's address bar
 - These cookies may be placed by an **advertisement** on the page or an image on the page
 - RFC 6265 allows browsers to implement whatever policy they wish regarding third party cookies
 - <https://tools.ietf.org/html/rfc6265>
 - Advertisers use third party cookies to **track a user** across multiple sites
 - A user visits `www.site1.com` which sets a cookie with domain `ad.adtracking.com`; later the same user visits `www.site2.com` which also sets a cookie with domain `ad.adtracking.com`; Eventually both cookies will be sent to the advertiser, that will match them
- Wikipedia has a nice description of cookies, see http://en.wikipedia.org/wiki/HTTP_cookie
- Google Chrome will kill support for 3rd party cookies, see <https://www.cnbc.com/2020/01/14/google-chrome-to-end-support-for-third-party-cookies-within-two-years.html>

Cookie Processing Algorithm

1. A URL is requested, (either by entering one into the address field or clicking on a link)
2. The browser scans its Cookie database for any cookies whose domain and path matches the requested URL
3. If any are found, all the cookies are sent along with the request as part of the HTTP headers (value of **Cookie**)
4. The server-side programs may/may not make use of any cookies from the client to determine what page to return
5. The server-side program may generate one (or more) cookies and send them along with the requested page; cookies are included in the HTTP headers returned to the browser (value of **Set-Cookie**)
6. The browser stores any new cookies into its database; cookies can be accessed on the client using the `document.cookie` object in JavaScript

Cookie Processing Example

https://www.google.com/?gws_rd=ssl

GET /?gws_rd=ssl HTTP/1.1

Host: www.google.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:43.0) Gecko/20100101 Firefox/43.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Connection: keep-alive

Cache-Control: max-age=0

HTTP/2.0 200 OK

Date: Mon, 29 Feb 2016 22:02:31 GMT

Content-Type: text/html; charset=UTF-8

Content-Encoding: gzip

Server: gws

Set-Cookie: NID=77=AITJ83oyT_00AB8c4ogH1JKOxUwf3w9SMg5tcZUjnqq_3mKK1AQTMPPIET1Q2FL1jaKpK-NFJ_v-HT469S0DK15SYn6Ct_1bGdn0xbbU-dLABnqUDneClbdgsGliFcKqZdfur3w9nN3VyQ; expires=Tue, 30-Aug-2016 22:02:31 GMT; path=/; domain=.google.com; HttpOnly

https://www.google.com/images/hpp/ic_wahlberg_product_core_48.png8.png

GET /images/hpp/ic_wahlberg_product_core_48.png8.png HTTP/1.1

Host: www.google.com

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:43.0) Gecko/20100101 Firefox/43.0

Accept: image/png,image/*;q=0.8,*/*;q=0.5

Accept-Encoding: gzip, deflate

Cookie: NID=77=AITJ83oyT_00AB8c4ogH1JKOxUwf3w9SMg5tcZUjnqq_3mKK1AQTMPPIET1Q2FL1jaKpK-NFJ_v-HT469S0DK15SYn6Ct_1bGdn0xbbU-dLABnqUDneClbdgsGliFcKqZdfur3w9nN3VyQ

Connection: keep-alive

If-Modified-Since: Wed, 09 Sep 2015 23:04:49 GMT

HTTP/2.0 304 Not Modified

Date: Mon, 29 Feb 2016 22:02:32 GMT

Expires: Mon, 29 Feb 2016 22:02:32 GMT

Last-Modified: Wed, 09 Sep 2015 23:04:49 GMT

Server: sffe

Additional Facts About Cookies

- **Scope:** by default, cookie scope is limited to all URLs on the current host name. Scope may be limited with the ***path=*** parameter to specify a specific path prefix to which the cookie should be sent, or broadened to a group of DNS names, rather than single host only, with ***domain=.***
- **Time to live:** by default, each cookie has a lifetime limited to the duration of the current browser session. Alternatively, an ***expires=*** parameter may be included to specify the date at which the cookie should be dropped
- **Overwriting cookies:** if a new cookie with the same ***NAME***, ***domain***, and ***path*** as an existing cookie is encountered, the old cookie is discarded
- **Deleting cookies:** There is no specific mechanism for deleting cookies, although a common hack is to overwrite a cookie with a bogus value as outlined above, plus a backdated or short-lived ***expires=***
- **"Protected" cookies:** as a security feature, some cookies set may be marked with a special ***secure*** keyword, which causes them to be sent over HTTPS only

Client-Side Cookies

- JavaScript has a property of the document **object** named `cookie`:
`document.cookie`
- This is a string variable that can be read and written using the JavaScript string functions
- A cookie can be removed from the cookie database either because it expires or because the cookie file gets too large
 - browsers need not store more than 300 cookies, nor more than 20 cookies per web server, nor more than 4K per cookie
- Setting `document.cookie` creates a new cookie for the web page
- Reading `document.cookie` retrieves all defined cookies (array)

escape(s) and unescape(s)

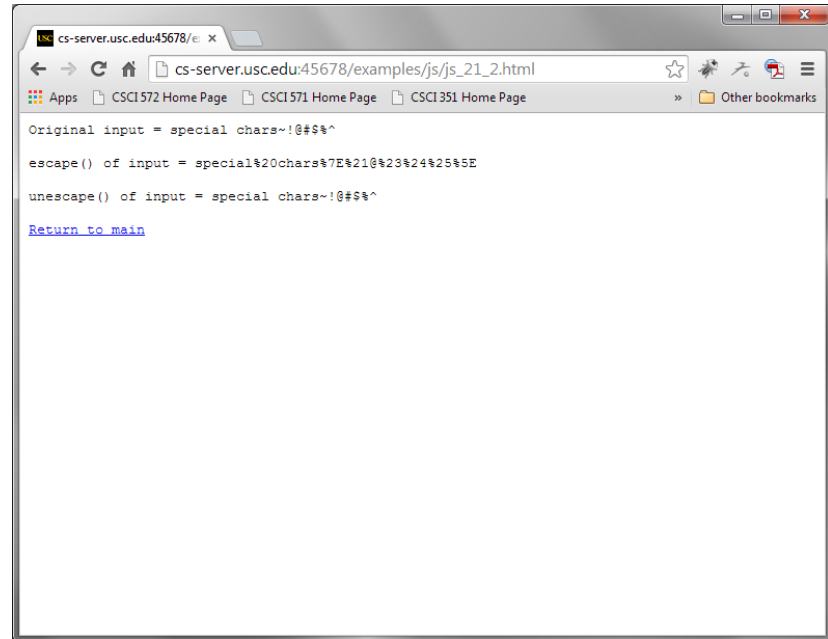
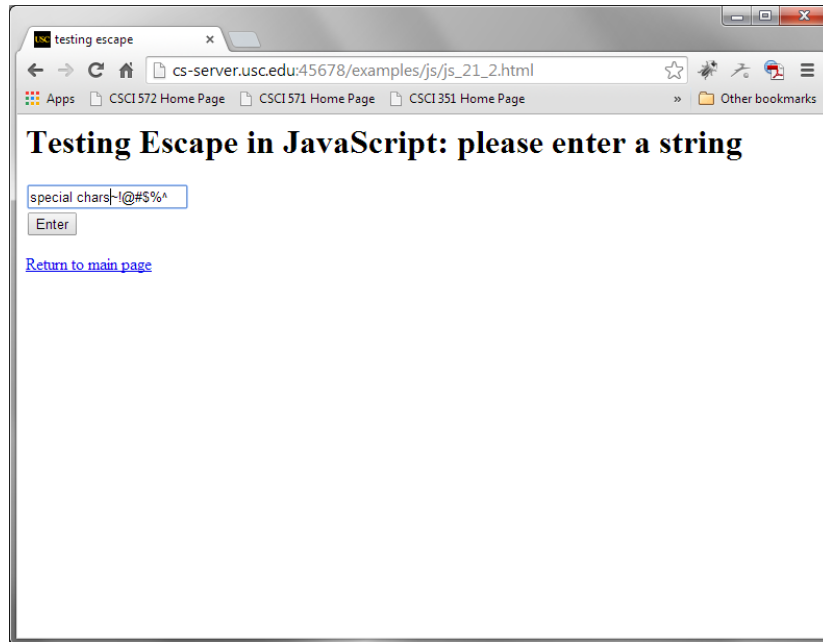
- Cookie 'values' should not contain white space, brackets, parentheses, equals signs, commas, double quotes, slashes, question marks, at signs, colons, and semicolons
 - Cookie values are encoded into their hex equivalents
- escape() and unescape() functions are properties of the "global object"
 - https://developer.mozilla.org/en-US/docs/Glossary/Global_object
- escape(s) returns a new version of string 's' that is encoded
 - all spaces, punctuation, accented characters, and other non-ASCII letters or numbers are converted to %xx format (ISO-8859-1)
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/escape
- unescape(s) returns a new version of string 's' that is decoded
 - all %xx are replaced by their character equivalent
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/unescape

Test of escape(), unescape()

```
<html><head><title>testing escape</title>
<script language=javascript>
function printout(x) {
    document.write("<PRE>")
    document.write("Original input = " + x + "<BR>")
    document.write("escape() of input = " + escape(x) +
        "<BR>")
    document.write("unescape() of input = " + unescape(x) +
        "<BR>")
    document.write("</PRE></BODY></HTML>" ) }
</script>
</head><body><H1>Testing Escape in JavaScript</h1>
<form name="myform">  <input name="sample"><br>
<input type="button" value="Go"
    onClick="printout(myform.sample.value)">
</form></body></html>
```

See <https://csci571.com/examples.html#cookies>

Browser Output Showing Use of Escape(), Unescape()



http://csci571.com/examples/js/js_21_2.html

Creating a Cookie in JavaScript

// Original JavaScript code by Chirp Internet:

www.chirp.com.au

<http://www.the-art-of-web.com/javascript/setcookie/>

// Please acknowledge use of this code by including this header.

```
var today = new Date();
```

```
var expiry = new Date(today.getTime() + 30 * 24 * 3600 *  
    1000); // plus 30 days
```

```
function setCookie(name, value, expiry)  
{  
    document.cookie=name + "=" + escape(value) + "  
    path=/; expires=" + expiry.toGMTString();  
}
```

produces a cookie that looks like

```
name= value; path=/; expires= date;
```

Retrieving a cookie in JavaScript

// Original JavaScript code by Chirp Internet:

www.chirp.com.au

// <http://www.the-art-of-web.com/javascript/getcookie/>

// Please acknowledge use of this code by including this header.

```
function getCookie(name)
{
    var re = new RegExp(name + "=([^\;]+)");
    var value = re.exec(document.cookie);
    return (value != null) ? unescape(value[1]) : null;
}
```


Removing a cookie in JavaScript

```
function removeCookie(name)
{
    var expired = "=; expires=Thu, 01 Jan 1970
    00:00:00 UTC; path=/;";

    document.cookie=name + expired;
}

// creates an early date;
// attaches it to the expires directive and
// assigns the name to the null string
```

JavaScript Cookie Libraries

- We have defined three JavaScript functions for handling cookies
 - **setCookie**(name, value, expiry)
 - **getCookie**(name)
 - **removeCookie**(name)
- Instead of including them in every html page that manipulates cookies, one can save them in a file, e.g. cookies.js and include the line
<SCRIPT language=JavaScript src=cookies.js>
.
.
.
</SCRIPT>

Complete JavaScript Example

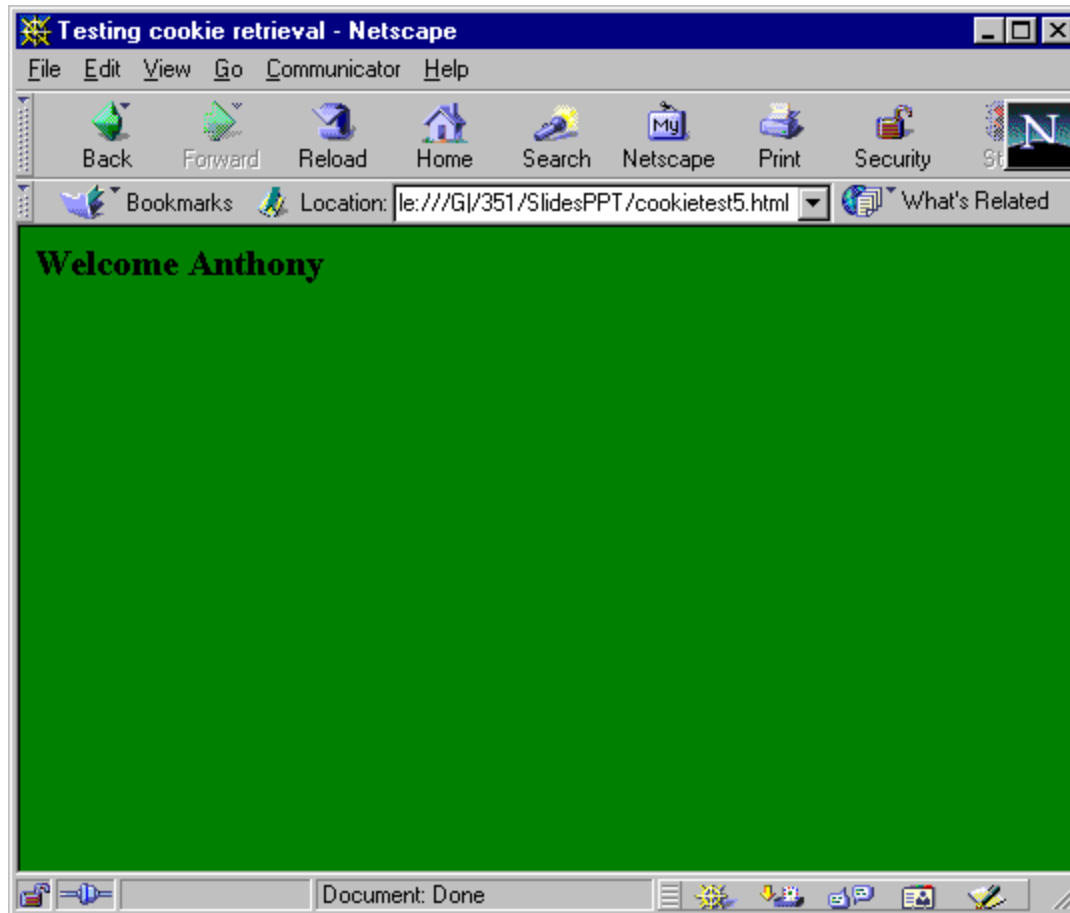
```
<HTML><HEAD><TITLE>Set Cookies</TITLE>
<SCRIPT language=JavaScript src=cookies.js></SCRIPT>
<SCRIPT language=JavaScript>
//cname: cookie name
    //cvalue: cookie value
    //expdays: the cookie expires after a certain number of days
See Next Slide
</SCRIPT></HEAD>
<BODY>
    <h2>Full Cookie Example</h2>
    <FORM method="GET" name="myform">
        What is your name: <input name="yourname" type="text" /><BR>
        Choose your background:<br>
        <input type="radio" name="backg" value="green">green<br>
        <input type="radio" name="backg" value="red">red<br>
        <input type="button" value="submit" onclick="saveCookies()">
    </FORM>
    <a href="http://csci571.com/examples.html#cookies">
Return to main page</a>
<noscript></BODY></HTML>
```

Complete JavaScript Example (2) - saveCookies

```
function saveCookies()
{
    // Remove any previous cookies
    //To delete a cookie, just set the expires parameter to a passed date:
    document.cookie = "personColor=; expires=Thu, 01 Jan 1970 00:00:00 UTC";
    document.cookie = "personName=; expires=Thu, 01 Jan 1970 00:00:00 UTC";

    var hisName = document.myform.yourname.value;
    var hisColor;
    if (document.myform.backg[0].checked)
    {
        hisColor=document.myform.backg[0].value;
    }
    else if (document.myform.backg[1].checked)
    {
        hisColor=document.myform.backg[1].value;
    }
    setCookie('personName', hisName, 30);
    setCookie('personColor', hisColor, 30);
    alert("personName =" + hisName + "\npersonColor = " + hisColor);
    window.location="js_21_5.html";
} </SCRIPT>
```

Browser Output



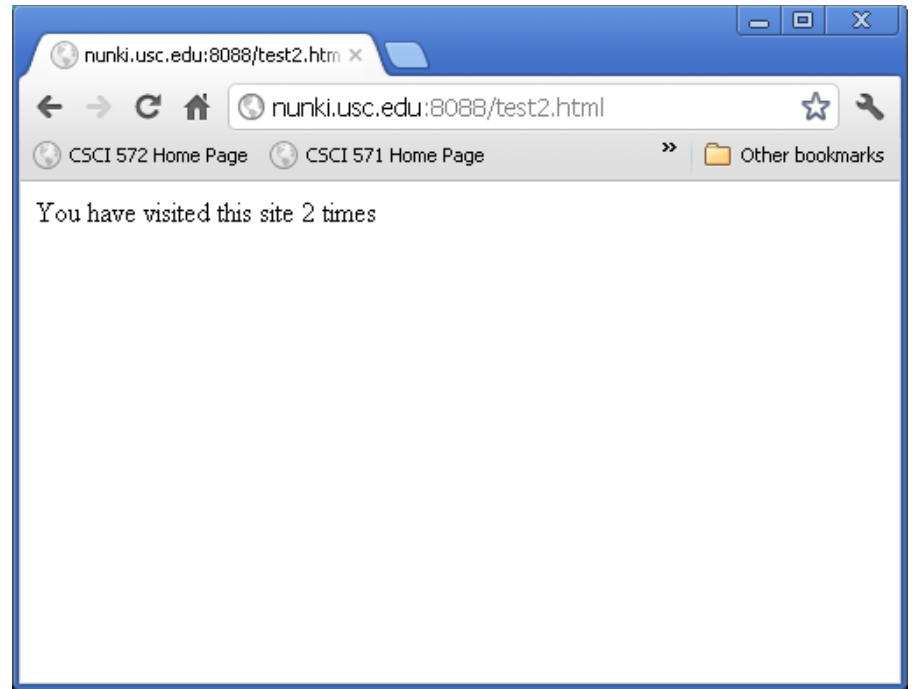
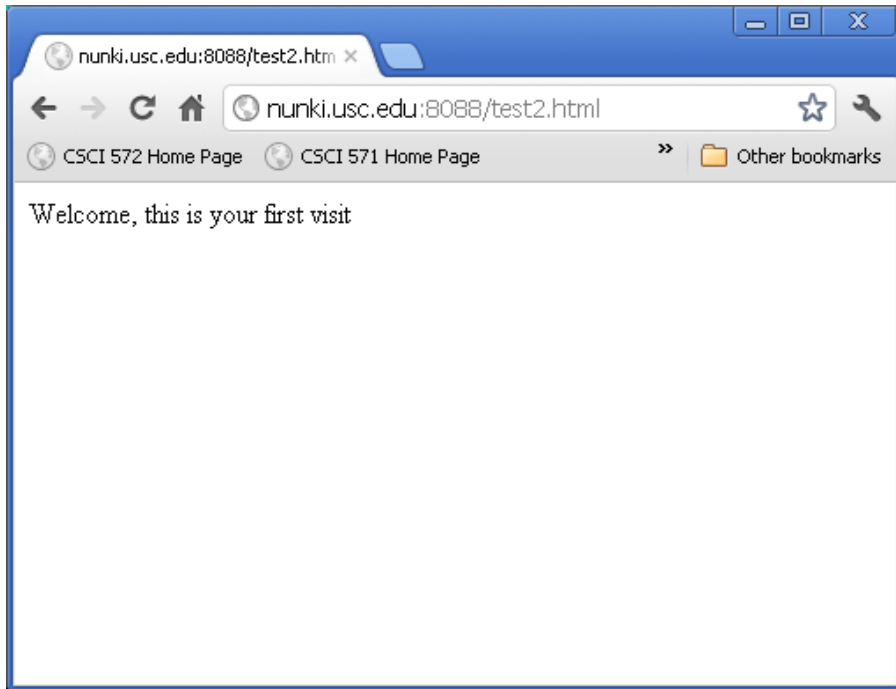
http://csci571.com/examples/js/js_21_4.html

Another Example: Site Visit

- A cookie can maintain a count of the number of times a client has visited your site

```
<HTML><HEAD><TITLE>Testing cookie counter</TITLE>
<script language=javascript src=cookies.js></script>
</head><body>
<script language=javascript>
var expire = new Date(21,12,31);
var numHits = getCookie("hits");
if (numHits) { //numHits is defined
    numHits = parseInt(numHits) + 1;
    document.write("You have visited this site " + numHits +
        " times"); }
else { numHits = 1;
    document.write("Welcome, this is your first
        visit"); }
setCookie("hits", numHits, expire); </script> </BODY>
</HTML>
```

Browser Output



http://csci571.com/examples/js/js_21_6.html

How Advertising on the Web Works

- An **online advertising network** or **ad network** is a company that connects advertisers to web sites that want to host advertisements.
 - The key function of an Ad Network is to place advertisements on the web sites of web publishers who wish to sell advertising space.
- There are four key players involved in an Ad Network's delivery of ads to users.
 - First, there are the **advertisers** that wish to place the ads.
 - Second, there are the **website owners** who wish to make money by selling ad space on their websites.
 - Third, there is the **Ad Network** that signs up advertisers and places their ads on the web pages of website owners.
 - Fourth, there are the **visitors** who view the web pages that contain the ads.
 - When a visitor requests a web page, the Ad Network is notified and it supplies one ad from its inventory to appear on the web page that was requested. The advertiser will pay the Ad Network for placing its ads and the Ad Network will return a portion of that fee to the website owner.

Cookie-based Marketing

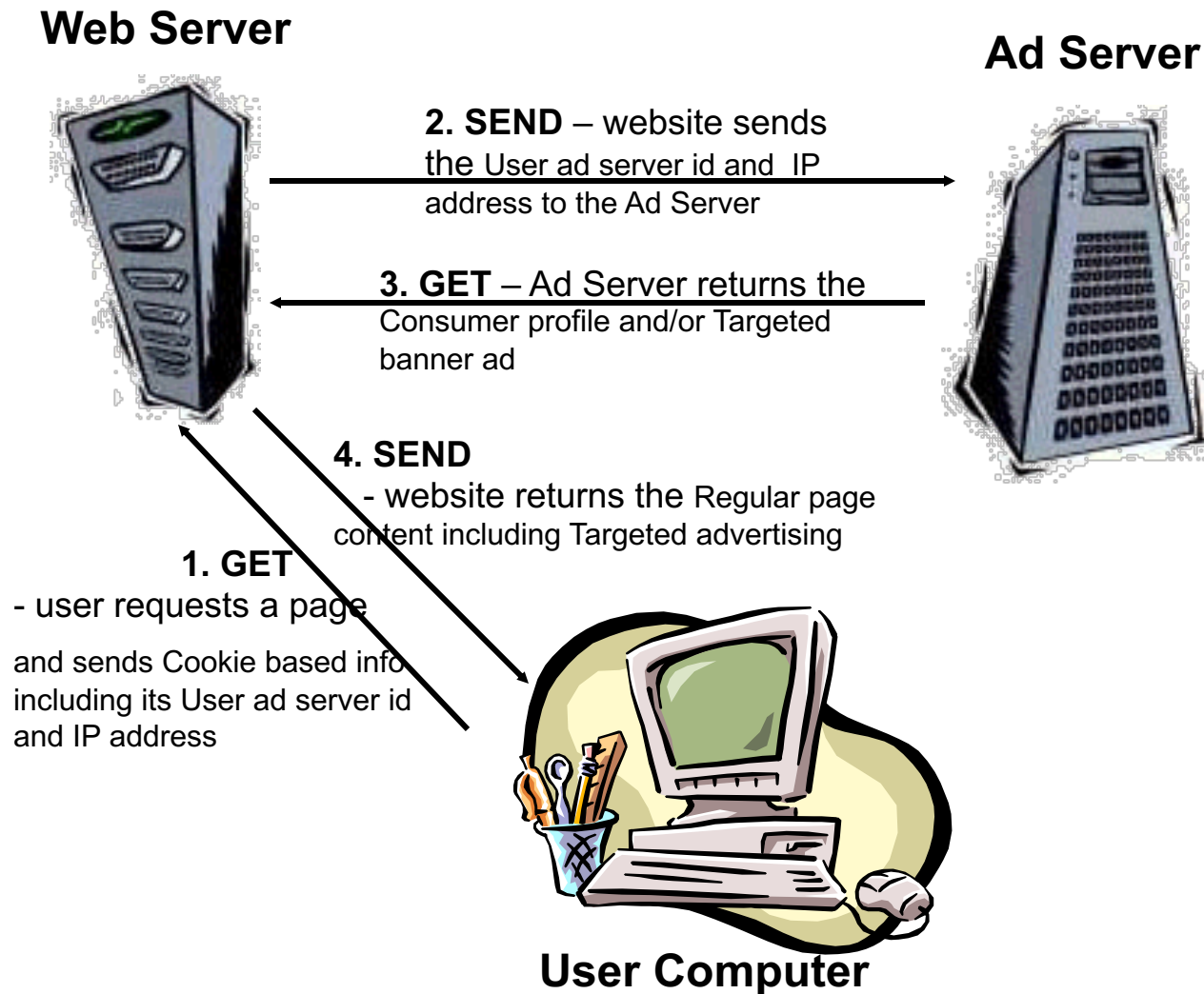
- **What is it?**

A *user customized* online advertising and marketing system that uses cookies and databases to create, maintain and utilize consumer profiles and monitor their activity

- **How does it work?**

- Ad serving companies make agreements with website owners; website owners agree to send cookies from ad serving companies to their clients
- When a user visits another such site, it sends data placed in your cookies to the Ad Serving company which retrieves marketing information about you from their database enabling them to customize the resulting ad
- **Result:** One person may see ads for sporting goods and another for baby clothes

Cookie-based Marketing - Schema



How Doubleclick Works

- **DoubleClick** is an Ad Network; it was purchased by Google for \$3.1 billion in 2007
- How Doubleclick works:
 - When a user invokes Web page, a tag on the page signals Doubleclick's server to delve into its inventory of advertisements to find one that matches the marketer's needs with the user's profile.
 - Here is an example of a Doubleclick tag
`http://ad.doubleclick.net/ABC/publisher/zone;topic=abc;sbtpc=def;cat=ghi;kw=xyz;tile=1;slot=728x90.1;sz=728x90;ord=7268140825331981?`
 - How to interpret the fields of a Doubleclick tag can be found here <http://www.adopsinsider.com/ad-ops-basics/how-to-read-doubleclick-ad-tags-and-ad-tag-variables/>
- Doubleclick will read multiple criteria including:
 - User location, embedded in a user's Internet address,
 - time of day,
 - cookies previously placed on the user's disk, which can further refine the target by telling Doubleclick whether someone is a repeat visitor , or has already seen a specific ad.

DoubleClick Ad Tag (Expanded)

- <http://ad.doubleclick.net/ADJ/publisher/zone;topic=abc;sbtpc=def;cat=ghi;kw=xyz;tile=1;slot=728x90.1;sz=728x90;ord=7268140825331981?>
- <http://ad.doubleclick.net/> - **host address for the ad server**
- **ADJ/** - **defines the Ad type which can be {images, XML, scripts}**
- **publisher/** - **identifies the website publisher, e.g. www.nytimes.com**
- **zone;** - **identifies the landing page at the publisher's site**
- **topic=abc;** - **identifies whatever topic is being talked about**
- **sbtpc=def;** - **subtopic level**
- **kw=xyz;** - **keyword level**
- **tile=1;** - **there may be multiple ads on the same page, each has a tile number**
- **slot=728x90.1;** - **defines the size of the ad (728 x 90) for tile 1**
- **ord=7268140825331981?** - **a random number that prevents the page from being cached**

How does Google use the DoubleClick cookie to serve ads?

- Google uses the DoubleClick cookies to collect information that includes:

time: 06/Aug/2011 12:01:32

ad_placement_id: 105

ad_id: 1003

userid: 000000000000000001

client_ip: 123.45.67.89

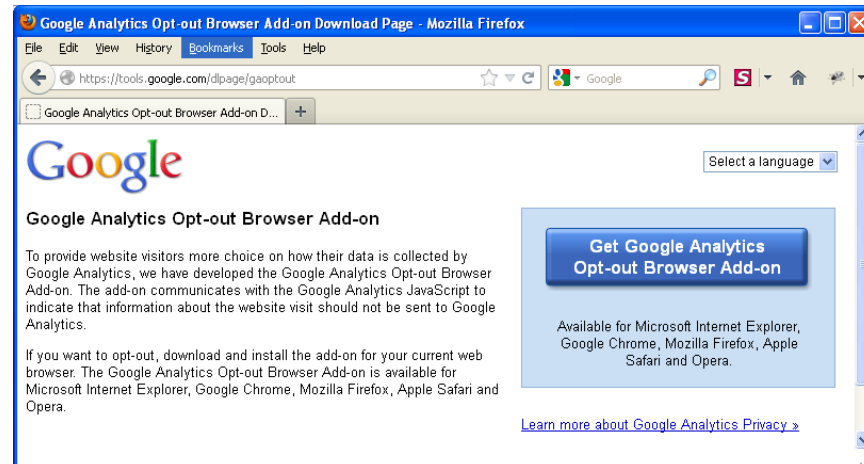
referral_url: "http://youtube.com/categories"

- The “time” field reflects the time the ad was displayed.
- The “ad placement id” and “ad id” identify the advertising campaign and the specific ad served.
- The “userid” is the display ad cookie that identifies the browser.
- The “client IP” reflects the user’s Internet Protocol (IP) address.
- A “referral URL” indicates the URL of the page where the ad was served. the logs also record whether a user’s browser clicks or interacts with an ad.
- **Opting Out**
 - Anyone who prefers not to see ads with this level of relevance can opt out. This opt-out will be specific only to the browser that you are using when you click the “Opt out” button.
 - For more details see <http://www.google.com/policies/privacy/ads/> and <http://www.google.com/ads/preferences/plugin/>
 - Use browser add-ons, such as AdBlock (getadblock.com)

How does Google use Cookies for Google Analytics?

- Google Analytics is Google's free web analytics tool that helps website owners understand how their visitors engage with their website.
- Google Analytics collects information anonymously, and reports website trends without identifying individual visitors.
- Analytics uses its own set of cookies to track visitor interactions. These cookies are used to store information, such as time of current visit, previous visits, and referred site.
- A different set of cookies is used for each website, and visitors are not tracked across sites.
- Available for IE11, Chrome, Firefox, Safari and Opera.
- To disable this cookie, you can install the **Google Analytics Opt-out Add-on** in your browser, which prevents Google Analytics from collecting information about your website visits. See:

<https://tools.google.com/dlpage/gaoptout>



Google Uses Cookies for Conversion Tracking

- Google uses cookies to help businesses that buy ads from Google determine how many people who click their ads end up purchasing their products.
- The **conversion tracking cookie** is set on your browser only when you click an ad delivered by Google where the advertiser has opted in to conversion tracking.
- These cookies expire within 30 days and do not contain information that can identify you personally. If this cookie has not yet expired when you visit certain pages of the advertiser's website, Google and the advertiser will be able to tell that you clicked the ad and proceeded to that page.
- Each advertiser gets a different cookie, so no cookie can be tracked across advertiser websites.
- If you want to **disable conversion tracking cookies**, you can set your browser to **block cookies from the googleadservices.com domain**.

Other Types of Cookies

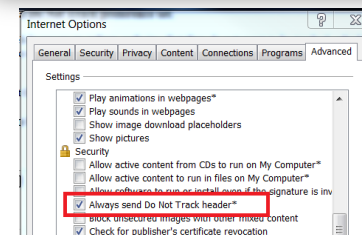
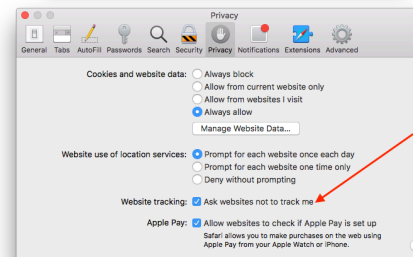
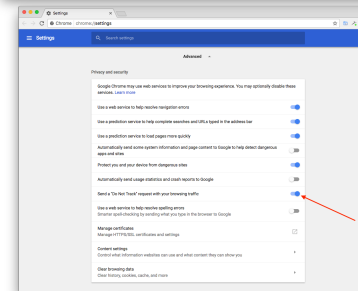
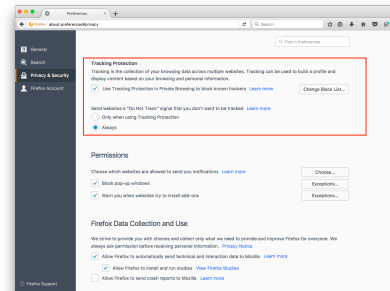
1. *Evercookie* is a JavaScript API that produces extremely persistent cookies in a browser
 2. When creating a new cookie, it uses the following storage mechanisms when available:
 - Standard [HTTP Cookies](#)
 - [Local Shared Objects](#) (Flash Cookies)
 - Silverlight [Isolated Storage](#)
 - Storing cookies in RGB values of auto-generated, force-cached PNGs using HTML5 Canvas tag to read pixels (cookies) back out
 - Storing cookies in [Web History](#)
 - Storing cookies in HTTP [ETags](#)
 - Storing cookies in [Web cache](#)
 - [window.name](#) caching
 - Internet Explorer [userData](#) storage
 - HTML5 [Session Storage](#), HTML5 [Local Storage](#), HTML5 [Global Storage](#), HTML5 [Database Storage](#) via SQLite
- See <http://samy.pl/evercookie/>
 - Developed by Samy Kamkar
 - Source code available on Github See: <https://github.com/samyk/evercookie>

Six Ways to Opt Out of cookies

1. **Select “do not track” in your browser Settings.** This setting is available Firefox 9+, Chrome, Safari 5.1+, Internet Explorer 9/10. See “Web Tracking Protection” submission:
<http://www.w3.org/Submission/2011/SUBM-web-tracking-protection-20110224/#dnt-uas>
2. **Download opt-out cookies.** This is a process that usually involves clicking on a button to download the opt-out cookie.
 - you go to the marketer's web site, find the privacy policy, then find the "opt out" information. The cookie your computer will get tells the company not to track you anymore.
3. **Use the cookie management tools in your web browser.** In most web browsers, you can set your browser to accept only session cookies, or to turn all cookies into session cookies. Session cookies are generally harmless.
 - For Macintosh Safari users, you can tell the browser to only accept cookies from "the site you are navigating to." This means that you will not accept third party cookies.
4. **View current cookies and delete what you don't need.** Most web browsers allow you to see what cookies you already have stored.
 - Some cookies, such as registration cookies for web sites you visit frequently, are useful to keep around. But other cookies, like tracking cookies from atdmt.com, doubleclick.net, 2o7.net, atwola.com, and other advertisers aren't necessarily helpful to you.
5. **Check your account preferences on registration sites.** Some sites, such as eBay, require registration and the use of cookies. On eBay, for example, if you do not opt-out of advertising tracking, information about your eBay activities can be used by other sites and advertisers outside of eBay.
6. **Use browser Add-ons.** Free browser extensions are available for most browsers to control tracking, such as Ghostery (www.ghostery.com)

Opt-Out's Do Not Track

- In Firefox: Preferences -> Privacy & Security -> check “ Use tracking protection in Private Browsing...”, -> select “Always” in Send Websites a “do not track” signal
- In Chrome: Preferences -> Settings -> Advanced -> Privacy and security -> check “Send a ‘Do Not Track’ request with your browsing traffic”
- In Safari: Preferences -> Privacy -> check “Ask websites not to track me”
- In IE 10: Internet options -> Advanced -> Settings -> Security -> Always send Do Not Track header [does not work in IE 11]



Canvas Fingerprinting

- **Canvas fingerprinting** is one of a number of browser fingerprinting techniques for tracking online users that allows websites to uniquely identify and track visitors **without** the use of browser cookies
- Primarily it makes use of the Canvas API of HTML5;
 - it relies on the fact that the drawing of the text will contain subtle differences that arise from font rasterization, anti-aliasing, pixel smoothing, related to the browser, etc.
 - See <https://developer.mozilla.org/en-US/docs/Web/API/HTMLCanvasElement>
- Here is the 4 step process that is followed:
 1. user visits a page
 2. fingerprinting script draws text with specific font and size and adds background colors
 3. the script calls Canvas API's ToDataURL method to get the canvas pixel data in dataURL format (Base64 encoded representation of binary pixel data)
 4. the script takes the hash of the text-encoded pixel data and uses that as the fingerprint
- While not sufficient to uniquely identify users by itself, this fingerprint is usually combined with other sources of information, e.g. browser plugins, to provide a unique identifier
- see the paper *The Web never forgets*, at https://securehomes.esat.kuleuven.be/~gacar/persistent/the_web_never_forgets.pdf

Some References

- Articles on Cookies
 - www.cookiecentral.com
 - www.echoecho.com
 - www.wmlpulse.com
 - www.epic.org
 - www.ciac.org
 - www.howstuffworks.com
 - www.webmonkey.com
 - www.ozemail.com.au
- Articles on Online Advertising
 - <http://en.wikipedia.org/wiki/DoubleClick>
 - <http://computer.howstuffworks.com/web-advertising.htm>
 - http://en.wikipedia.org/wiki/Online_advertising