

The results are in! See what nearly 90,000 developers picked as their most loved, dreaded, and desired coding languages and more in the 2019 Developer Survey.

How do you disable browser Autocomplete on web form field / input tag?

[Ask Question](#)

▲ How do you disable autocomplete in the major browsers for a specific input (or form field)? ▼

2500

html

browser

autocomplete



323

edited May 23 '18 at 7:06

[molerat](#)

409 1 9 28

asked Aug 5 '08 at 16:22

[Brett Veenstra](#)

21.6k 17 60 79

- 67 One more usage includes administration forms to create or edit users; you don't want the form pre-filled with your current credentials. This also applies to password change forms, esp. those designed as *leave blank to keep current password*. – [Álvaro González](#) Sep 29 '09 at 8:17
- 77 Another reason you'd want to do this is so password, captcha, and credit card information doesn't get filled in. – [Jeff Atwood](#) Jun 6 '10 at 7:00
- 24 also note that some penetration tests require disabling autocomplete on certain fields – [Jeff Atwood](#) Jun 6 '10 at 7:00
- 21 Please think carefully about this. It makes sense to disable saving credit card information etc, but unless you are a bank, preventing password

autofill can significantly decrease how many users bother to login to your site, especially on phones where entering passwords on cramped keyboards is a lot of effort. – [John Mellor](#) Nov 22 '12 at 10:48

16 It can be a really valid thing to do in other situations though - such as in user-management forms - adding / updating a user - you almost never want passwords to be auto completed when admining the set of users in a large system. – [Dave Amphlett](#) Nov 23 '12 at 11:08

63 Answers

1 2 3 next

▲
2420



Firefox 30 ignores `autocomplete="off"` for passwords, opting to prompt the user instead whether the password should be stored on the client. Note the following [commentary](#) from May 5, 2014:

- The password manager *always* prompts if it wants to save a password. Passwords are not saved without permission from the user.
- We are the third browser to implement this change, after IE and Chrome.

According to [Mozilla developer documentation](#) the form element attribute `autocomplete` prevents form data from being cached in older browsers.

```
<input type="text" name="foo" autocomplete="off" />
```

edited Jan 20 '15 at 9:23



[Dave Jarvis](#)

21.3k 30 132 258

answered Aug 5 '08 at 16:24



[nlucaroni](#)



40.8k 4 51 80

- 38 This did not work for me in Firefox 3.0.3 I had to put the autocomplete attribute in the FORM rather than the INPUT. – [Winston Fassett](#) Nov 12 '08 at 4:11
- 13 Autocomplete is only defined in the HTML 5 standards, so it will break any validations you run against HTML 4.*... – [Jrgns](#) Jan 19 '09 at 8:04
- 88 @Winston, you should put it both on the form, AND on the input element itself. That way you cover all the nonstandardness of browsers. – [AviD](#) Dec 13 '10 at 12:11
- 81 And remember to disable your [autocomplete = on](#) extension (if you're using Chrome) before you test your webapp. Else you'll feel real silly like me. ;) – [Jo Liss](#) Feb 26 '11 at 0:57
- 8 autocomplete="<anything but on or off works>" – [swapab](#) Jan 6 '16 at 11:05

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams

Q&A for work

[Learn More](#)

I've solved the endless fight with Google Chrome with the use of random characters.

16



```
<input name="name" type="text" autocomplete="rutjfkde">
```

Hope that it will help to other people.

edited Apr 5 at 12:56

answered Aug 4 '18 at 14:22



step

621 2 9 29

- 1 Works well on my forms. – [user5957221](#) Aug 4 '18 at 14:29

-
- 1 Sadly it works better than what is called standard on chrome – [Nour Lababidi](#) Nov 4 '18 at 3:01
-
- 1 WOW, this works well! Thank you! – [Rudy](#) Mar 27 at 15:51
-

2

I wanted something that took the field management completely out of the browser's hands, so to speak. In this example, there's a single standard text input field to capture a password — no email, user name etc...

```
<input id='input_password' type='text' autocomplete='off' autofocus>
```

There's a variable named "input", set to be an empty string...

```
var input = "";
```

The field events are monitored by JQuery...

1. On focus, the field content and the associated "input" variable are always cleared.
2. On keypress, any alphanumeric character, as well as some defined symbols, are appended to the "input" variable, and the field input is replaced with a bullet character. Additionally, when the Enter key is pressed, the typed characters (stored in the "input" variable) are sent to the server via Ajax. (See "Server Details" below.)
3. On keyup, the Home, End, and Arrow keys cause the "input" variable and field values to be flushed. (I could have gotten fancy with arrow navigation and the focus event, and used .selectionStart to figure out where the user had clicked or was navigating, but it's not worth the effort for a password field.) Additionally, pressing the Backspace key truncates both the variable and field content accordingly.

```
$("#input_password").off().on("focus", function(event) {
    $(this).val("");
    input = "";

}).on("keypress", function(event) {
    event.preventDefault();

    if (event.key !== "Enter" && event.key.match(/^[\0-9a-z!@#\$%&*-_
        $(this).val( $(this).val() + "." );
        input += event.key;
    }
    else if (event.key == "Enter") {
        var params = {};
        params.password = input;

        $.post(SERVER_URL, params, function(data, status, ajax) {
            location.reload();
        });
    }
});

}).on("keyup", function(event) {
    var navigationKeys = ["Home", "End", "ArrowLeft", "ArrowRight",
"ArrowDown"];
    if ($.inArray(event.key, navigationKeys) > -1) {
        event.preventDefault();
        $(this).val("");
        input = "";
    }
    else if (event.key == "Backspace") {
        var length = $(this).val().length - 1 > 0 ? $(this).val().length : 0;
        input = input.substring(0, length);
    }
});
```

Front-End Summary

In essence, this gives the browser nothing useful to capture. Even if it overrides the autocomplete setting, and/or presents a dropdown with previously entered values, all it has is bullets stored for the field value.

Server Details (optional reading)

As shown above, Javascript executes `location.reload()` as soon as the server returns a JSON response. (This logon technique is for access to a restricted administration tool. Some of the overkill, related to the cookie content, could be skipped for a more generalized implementation.) Here are the details:

- When a user navigates to the site, the server looks for a legitimate cookie.
- If there is no cookie, the logon page is presented. When the user enters a password and it is sent via Ajax, the server confirms the password and also checks to see if the user's IP is in an Authorized IP list.
- If either the password or IP are not recognized, the server doesn't generate a cookie, so when the page reloads, the user sees the same logon page.
- If both the password and IP are recognized, the server generates a cookie that has a ten-minute life span, and it also stores two scrambled values that correspond with the time-frame and IP.
- When the page reloads, the server finds the cookie and verifies that the scrambled values are correct (i.e., that the time-frame corresponds with the cookie's date and that the IP is the same).
- The process of authenticating and updating the cookie is repeated every time the user interacts with the server, whether they are logging in, displaying data, or updating a record.
- If at all times the cookie's values are correct, the server presents the full website (if the user is logging in) or fulfills whatever display or update request was submitted.
- If at any time the cookie's values are not correct, the server removes the current cookie which then, upon reload, causes the logon page to be re-displayed.

edited Mar 15 at 23:32

answered Feb 19 at 20:16



Alan M.

707 2 11 28

3

If you want to prevent the common browser plug-in LastPass from auto-filling a field as well, you can add the attribute `data-lpignore="true"` added to the other suggestions on this thread. Note that this doesn't only apply to password fields.

```
<input type="text" autocomplete="false" data-lpignore="true" />
```

I was trying to do this same thing a while back, and was stumped because none of the suggestions I found worked for me. Turned out it was LastPass .

edited Feb 15 at 8:32



SollyM

987 1 13 29

answered Nov 21 '18 at 23:06



JoerT

57 4

1

This worked for me like a charm.

1. Set the autocomplete attribute of the form to off
2. Add a dummy input field and set its attribute also to off.

```
<form autocomplete="off">
  <input type="text" autocomplete="off" style="display:none">
</form>
```

answered Dec 17 '18 at 6:39



zapping

2,901 4 34 50

Try to add

11

```
readonly onfocus="this.removeAttribute('readonly');"
```

in addition to

```
autocomplete="off"
```

to the input(s) that you do not want to remember form data
(username , password , etc.) as shown below:

```
<input type="text" name="UserName" autocomplete="off" readonly
onfocus="this.removeAttribute('readonly');" >
```

```
<input type="password" name="Password" autocomplete="off" readonly
onfocus="this.removeAttribute('readonly');" >
```

Update: Here is full example below based on this approach that prevents from *drag & drop*, *copy*, *paste*, etc.

```
<input type="text" name="UserName" style="text-transform:lowercase;"
placeholder="Username"
autocomplete="off" readonly onfocus="this.removeAttribute('readonly');"
oncopy="return false" ondrag="return false" ondrop="return false"
```



```
onpaste="return false" oncontextmenu="return false" >
```

```
<input type="password" name="Password" placeholder="Password" autocor
readonly
onfocus="this.removeAttribute('readonly');" oncopy="return false
false"
ondrop="return false" onpaste="return false" oncontextmenu="retu
```

Tested on the latest versions of the major browsers i.e. Google Chrome , Mozilla Firefox , Microsoft Edge , etc. and working without any problem. Hope this helps...

edited Nov 8 '18 at 18:20

answered Nov 1 '17 at 13:26



[Murat Yıldız](#)

7,135 5 38 46

1 For me in IE11 I can't type into the text box even after the onfocus removes the readonly attribute. However, if I click a second time on the text box then I can type. – [mcallahan](#) Nov 10 '17 at 19:16

1 good idea, combine it with others perhaps – [Andrew](#) Nov 8 '18 at 16:35

3

To prevent browser auto fill with the user's saved site login credentials, place a text and password input field at the top of the form with non empty values and style "position: absolute; top: -999px; left: -999px" set to hide the fields.

```
<form>
  <input type="text" name="username_X" value="-" tabindex="-1" aria-l
style="position: absolute; top: -999px; left: -999px" />
  <input type="password" name="password_X" value="-" tabindex="-1" a
style="position: absolute; top: -999px; left: -999px" />
```

```
<!-- Place the form elements below here. -->
</form>
```

It is important that a text field precede the password field. Otherwise the auto fill may not be prevented in some cases.

It is important that the value of both the text and password fields not be empty, to prevent default values from being overwritten in some cases.

It is important that these two fields are before the "real" password type field(s) in the form.

For newer browsers that are html 5.3 compliant the autocomplete attribute value "new-password" should work.

```
<form>
  <input type="text" name="username" value="" />
  <input type="password" name="password" value="" autocomplete="new-|
</form>
```

A combination of the two methods can be used to support both older and newer browsers.

```
<form>
  <div style="display:none">
    <input type="text" readonly tabindex="-1" />
    <input type="password" readonly tabindex="-1" />
  </div>
  <!-- Place the form elements below here. -->
  <input type="text" name="username" value="" />
  <input type="password" name="password" value="" autocomplete="new-|
</form>
```

edited Jul 10 '18 at 18:25

answered Jul 7 '18 at 18:20



NOYB

172 1 9

No fake inputs, no javascript!

6

There is no way to disable autofill consistently across browsers. I have tried all the different suggestions and none of them work in all browsers. The only way is not using password input at all. Here's what I came up with:

```
<style type="text/css">
  @font-face {
    font-family: 'PasswordDots';
    src: url('text-security-disc.woff') format('woff');
    font-weight: normal;
    font-style: normal;
  }

  input.password {
    font-family: 'PasswordDots' !important;
    font-size: 8px !important;
  }
</style>

<input class="password" type="text" spellcheck="false" />
```

Download: text-security-disc.woff

Here's how my final result looks like:

A screenshot of a web form. At the top, there's a label 'PASSWORD' with a small diamond-shaped toggle icon to its left. Below the label is a text input field. Inside the input field, there are 12 dots representing a password. To the right of the input field is a blue eye icon, which typically represents a toggle for password visibility. The input field has a light blue border and is set against a light gray background.

The negative side effect is that it's possible to copy plain text from the input, though it should be possible to prevent that with some JS.

edited Jul 9 '18 at 14:55

answered Apr 13 '18 at 6:35



Julius Žaromskis

584 4 12



You can use `autocomplete = off` in input controls to avoid auto completion

2

For example:



```
<input type=text name="test" autocomplete="off" />
```

if the above code doesn't work then try to add those attributes also

```
autocapitalize="off" autocomplete="off"
```

or

Change input type attribute to `type="search"` . Google doesn't apply auto-fill to inputs with a type of search.

edited Jul 9 '18 at 2:16

community wiki
2 revs, 2 users 92%
Codemaker



4

Google Chrome ignores the `autocomplete="off"` attribute for certain inputs, including password inputs and common inputs detected by name.



For example, if you have an input with name `address` , then Chrome will provide autofill suggestions from addresses entered on other sites, *even if you tell it not to*:

```
<input type="string" name="address" autocomplete="off">
```

If you don't want Chrome to do that, then you can rename or namespace the form field's name:

```
<input type="string" name="mysite_addr" autocomplete="off">
```

If you don't mind auto-completing values which were previously entered on your site, then you can leave autocomplete enabled. Namespacing the field name should be enough to prevent values remembered from other sites from appearing.

```
<input type="string" name="mysite_addr" autocomplete="on">
```

answered Jun 22 '18 at 6:10



joeytwiddle

16.7k 8 75 79



To solve this problem, I have used some CSS tricks and the following works for me.

4



```
input {  
  text-security:disc;  
  -webkit-text-security:disc;  
  -moz-text-security:disc;  
}
```

Please read [this](#) article for further detail.

edited May 11 '18 at 15:37



RobC

6,817 9 27 41

answered May 11 '18 at 13:19



gem007bd

659 7 8



I was able to stop Chrome 66 from autofilling by adding two fake inputs and giving them position absolute:

3



```
<form style="position: relative">  
  <div style="position: absolute; top: -999px; left: -999px;">  
    <input name="username" type="text" />  
    <input name="password" type="password" />  
  </div>
```

```
<input name="username" type="text" />
<input name="password" type="password" />
```

At first, I tried adding `display:none;` to the inputs but Chrome ignored them and autofilled the visible ones.

answered May 8 '18 at 22:07



[user3014373](#)

66 3



276

In addition to `autocomplete=off`, you could also have your form fields names be randomized by the code that generates the page, perhaps by adding some session-specific string to the end of the names.



When the form is submitted, you can strip that part off before processing them on the server side. This would prevent the web browser from finding context for your field and also might help prevent XSRF attacks because an attacker wouldn't be able to guess the field names for a form submission.

edited May 7 '18 at 7:56



[CodeCaster](#)

110k 17 148 200

answered Oct 20 '08 at 13:36






[Ben Combee](#)

12k 6 34 38

7 This is a much better solution compared to using `autocomplete="off"`. All you have to do is generate a new name on every page load and save that name to a `$_SESSION` for future use: `$_SESSION['codefield_name'] = md5(uniqid('auth', true));` – [enchance](#) Nov 13 '11 at 9:03

67 No, this is not a better solution, because the origin of preference for this setting is user agent also known as the web browser. There is a

difference between supporting certain behaviour (which HTML 5 attempts to do) and forcing it by deciding on behalf of the user, which you suggest is a "much better solution". – [amn](#) May 27 '13 at 17:09 

- 12 This solution can work with all browsers, so in that respect it is "better". Still, amn is correct, deciding to disable autocomplete on behalf of your users is not a good idea. This means I would only disable autocomplete in very specific situations, such as when you plan to build your own autocomplete functionality and don't want conflicts or strange behavior. – [macguru2000](#) May 15 '14 at 21:34 
- 6 Regarding XSRF attacks, I'm not sure what type of attack you were picturing, but couldn't the attacker just strip off the end part the same way you do server-side to identify the fields? Or if the attacker is posting the fields, couldn't they append their own random string since it'll be stripped off by the server? – [xr280xr](#) Feb 11 '15 at 20:10 
- 5 @macguru2000 building your own autocomplete is a completely legit and common use-case. Really the browser should make it easier for developers to turn off autocomplete when they need to instead of forcing us to use hacks like this one – [whoadave](#) May 6 '15 at 2:01

at last i got a solution first add 2 hidden text fields

1

and just add a angular directive like this

```
(function () {
    'use strict';

    appname.directive('changePasswordType', directive);

    directive.$inject = ['$timeout', '$rootScope', '$cookies'];

    function directive($timeout, $rootScope, $cookies) {
        var directive = {
            link: link,
            restrict: 'A'
        };

        return directive;
    }
})
```



```

function link(scope,element) {
  var process = function () {
    var elem =element[0];
    elem.value.length > 0 ? element[0].setAttribute("type", "password") :
    element[0].setAttribute("type", "text");
  }

  element.bind('input', function () {
    process();
  });

  element.bind('keyup', function () {
    process();
  });
}
}
})();

```

then use it in your text field where you need to prevent auto complete

```

<input type="text" style="display:none">\\can avoid this 2 lines
<input type="password" style="display:none">
<input type="text" autocomplete="new-password" change-password-

```

NB: dont forget to include jquery, and set type ="text" initially

answered Mar 26 '18 at 12:07



Edison

737 7 24



The best solution:

13

Prevent autocomplete username (or email) and password:



```

<input type="email" name="email"><!-- Can be type="text" -->
<input type="password" name="password" autocomplete="new-password">

```

Prevent autocomplete a field:

```
<input type="text" name="field" autocomplete="nope">
```

Explanation: autocomplete continues work in <input> ,
autocomplete="off" does not work, but you can change off to a
random string, like nope .

Works in:

- Chrome: 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63 and 64
- Firefox: 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57 and 58

answered Mar 20 '18 at 4:31



Cava

2,425

1 11 23

-
- 2 I found this to work in my preliminary testing. So odd that "off" doesn't work. – [Craig Jacobs](#) Mar 26 '18 at 14:36
-



You can also use a instead of an **input** to skip the autocomplete part.

1



answered Mar 19 '18 at 12:18



Sofiene Djebali

3,150

1 12 25



This works for me.

16

```
<input name="pass" type="password" autocomplete="new-password" />
```

We can also use this strategy in other controls like text, select etc

edited Mar 3 '18 at 7:34



DisplayName

11.2k 2 7 20

answered Dec 30 '16 at 15:05



Muhammad Awais

2,037 1 25 22

As others have said, the answer is `autocomplete="off"`

57

However, I think it's worth stating **why** it's a good idea to use this in certain cases as some answers to this and [duplicate](#) questions have suggested it's better not to turn it off.

Stopping browsers storing credit card numbers shouldn't be left to users. Too many users won't even realize it's a problem.

It's particularly important to turn it off on fields for credit card security codes. As [this page](#) states:

"Never store the security code ... its value depends on the presumption that the only way to supply it is to read it from the physical credit card, proving that the person supplying it actually holds the card."

The problem is, if it's a public computer (cyber cafe, library etc) it's then easy for other users to steal your card details, and even on your own machine a malicious website could [steal autocomplete data](#).

edited Jan 25 '18 at 8:59



Zub

553 7 20

answered Jan 23 '09 at 21:21



Sam Hasler

11.2k 8 59 98

-
- 7 if i went to a site and it remembered my card in the dropdown i'd be very unhappy. id start to wonder how they could be so careless. – [Simon_Weaver](#) Nov 22 '09 at 5:13
-
- 6 Much simpler / more critical case. When I visit a user's page on the admin portion of my site, it tries to set their username and password to be my admin username and password, not being able to tell that this isn't a login form. I want to have my admin password remembered, but it is a critical error that it tries to apply that remembered username / password to any users that I then edit. – [rjmunro](#) Dec 17 '15 at 13:35
-



Sometimes **even autocomplete=off** would **not prevent to fill** in credentials into wrong fields, but not user or nickname field.

131



This workaround is in addition to apinstein's post about browser behavior.

fix browser autofill in read-only and set writable on focus (click and tab)

```
<input type="password" readonly
onfocus="this.removeAttribute('readonly');"/>
```

Update: Mobile Safari sets cursor in the field, but does not show virtual keyboard. New Fix works like before but handles virtual keyboard:

```
<input id="email" readonly type="email" onfocus="if (this.hasAttribu
this.removeAttribute('readonly');
```

```
// fix for mobile safari to show virtual keyboard  
this.blur();    this.focus(); }" />
```

Live Demo <https://jsfiddle.net/danielsuess/n0scguv6/>

// UpdateEnd

Because Browser auto fills credentials to wrong text field!?

I notice this strange behavior on Chrome and Safari, when there are password fields in *the same form*. I guess, the browser looks for a password field to insert your saved credentials. Then it auto fills (just guessing due to observation) the nearest textlike-input field, that appears prior the password field in DOM. As the browser is the last instance and you can not control it,

This readonly-fix above worked for me.

edited Jan 24 '18 at 22:58



Zub

553 7 20

answered Jun 16 '14 at 16:04



dsuess

3,865 1 15 21

-
- 7 An if there is no javascript then the whole form fails. -1 – [Jimmy Kane](#) Jul 10 '14 at 10:56
-
- 6 @JimmyKane the key would be to also add the attribute using javascript in the first place (which dsuess hasn't done here, but just adding for completeness sake). – [trnelson](#) Mar 27 '15 at 17:58
-
- 3 This doesn't work right in IE8, the readonly password field is not editable the first time you focus it, only after you unfocus and focus again. Nice idea, but unfortunately it's a bit too hacky and not safe to use. – [Sam Watkins](#) May 19 '15 at 4:57 ✎
-

202

Most of the major browsers and password managers (correctly, IMHO) now ignore `autocomplete=off`.

Why? Many banks and other "high security" websites added `autocomplete=off` to their login pages "for security purposes" but this actually decreases security since it causes people to change the passwords on these high-security sites to be easy to remember (and thus crack) since autocomplete was broken.

Long ago most password managers started ignoring `autocomplete=off`, and now the browsers are starting to do the same for username/password inputs only.

Unfortunately, bugs in the autocomplete implementations insert username and/or password info into inappropriate form fields, causing form validation errors, or worse yet, accidentally inserting usernames into fields that were intentionally left blank by the user.

What's a web developer to do?

- If you can keep all password fields on a page by themselves, that's a great start as it seems that the presence of a password field is the main trigger for user/pass autocomplete to kick in. Otherwise, read the tips below.
- **Safari** notices that there are 2 password fields and disables autocomplete in this case, assuming it must be a change password form, not a login form. So just be sure to use 2 password fields (new and confirm new) for any forms where you allow
- **Chrome** 34, unfortunately, will try to autofill fields with user/pass whenever it sees a password field. This is quite a bad bug that hopefully, they will change the Safari behavior. However, adding this to the top of your form seems to disable the password autofill:

```
<input type="text" style="display:none">
<input type="password" style="display:none">
```

I haven't yet investigated IE or Firefox thoroughly but will be happy to update the answer if others have info in the comments.

edited Jan 24 '18 at 22:14



Zub

553 7 20

answered Apr 23 '14 at 4:00



apinstein

3,947 1 19 21

-
- 4 what do you mean with "adding this on your page seems to disable autofill for the page:" – [wutzebaer](#) May 7 '14 at 10:31
-
- 4 @wutzebaer, Chrome notices the hidden password field and halts auto-complete. Reportedly this is to prevent the site stealing password info without the user noticing. – [David W](#) Dec 1 '14 at 23:05
-
- 5 Your snippet of code prevent autocompletes for login fields on Chrome, Firefox, IE 8 and IE 10. Did not test IE 11. Good stuff! Only simple answer that still works. – [Sam Watkins](#) May 19 '15 at 4:44
-
- 2 Your safari note seems to work on Chrome too, at least as of Dec 2015. I had a username and password field on a registration form that was autocompleting with data from the login form. Creating two `type='password'` fields on the one page caused the browser's "save password" autocomplete to be ignored, which made a whole load of sense since registration forms tend to ask for the password twice when login forms only ask for it once. – [Matt Fletcher](#) Dec 20 '15 at 20:27
-
- 2 Does not seem to work in Chrome 55 anymore, unless the extra password field is not hidden, which defeats the purpose. – [jokkedk](#) Oct 10 '16 at 9:06
-



This is a security issue that browsers ignore now. Browsers identify and stores content using input names, even if developers consider

9

the information is sensitive and should not be stored. Making an input name different between 2 requests will solve the problem (but will still be saved in browser's cache and will also increase browser's cache). Ask the user to activate or deactivate options in its browser's settings is not a good solution. The issue can be fixed in the backend.

Here's my fix. An approach that I have implemented in my framework. All autocomplete elements are generated with an hidden input like this :

```
<? $r = rmd5(rand()).microtime(TRUE)); ?>
<form method="POST" action="." />
  <input type="text" name="<? echo $r; ?>" />
  <input type="hidden" name="__autocomplete_fix_<? echo $r; ?>" value="" />
  <input type="submit" name="submit" value="submit" />
</form>
```

Server then process post variables like this :

```
foreach ($_POST as $key => $val)
{
    if(preg_match('#__autocomplete_fix_#', $key) === 1){
        $n = substr($key, 19);
        if(isset($_POST[$n]))$_POST[$val] = $_POST[$n];
    }
}
```

The value can be accessed as usual

```
var_dump($_POST['username']);
```

And the browser won't be able to suggest information from the previous request or from previous users.

All works like a charm, even if browsers updates, want to ignore autocomplete or not. That has been the best way to fix the issue for me.

edited Jan 24 '18 at 19:33

community wiki

5 revs, 3 users 93%

Simmoniz

▲ Fixed. Just need to add above real input field

1

▼ https://developer.mozilla.org/en-US/docs/Web/Security/Securing_your_site/Turning_off_form_autocompletion - MDN <https://medium.com/paul-jaworski/turning-off-autocomplete-in-chrome-ee3ff8ef0908> - medium tested on EDGE, Chrome(latest v63), Firefox Quantum (57.0.4 64-бит), Firefox(52.2.0) fake fields are a workaround for chrome/opera autofill getting the wrong fields

```
const fakeInputStyle = {opacity: 0, float: 'left', border: 'none', |  
'0'}
```

```
<input type="password" name='fake-password' autoComplete='new-passw  
style={fakeInputStyle} />
```

```
<TextField  
  name='userName'  
  autoComplete='nope'  
  ...  
>
```

```
<TextField  
  name='password'  
  autoComplete='new-password'  
  ...  
>
```

answered Jan 19 '18 at 12:57



Palaniichuk Dmytro

630 1 12 30

16

Was a non-standard way to do this (I think Mozilla and Internet Explorer still support it) but messing with the users expectations is a bad idea.

If the user enters their credit card details in a form and then let's someone else use that browser it's not your concern. :)

edited Jan 6 '18 at 10:54



Raghav Dinesh

406 4 12

answered Aug 5 '08 at 16:26



sparkes

14.5k 5 34 44

1 As far as I remember, most browsers don't use autocomplete even if it is on when doing something over HTTPS. If your users enter their credit card details into a form that will be sent over HTTP, they might have bigger problems than autocomplete. – [pilsetnieks](#) Oct 29 '08 at 12:04

21 and it IS your problem if someone gets frauded and blames you because you were the only merchant they gave the card to – [Simon_Weaver](#) Nov 22 '09 at 5:11

1 @pilsetnieks: That may have once been true; it's certainly not anymore, at least with Firefox 27. – [Lawrence Dol](#) Feb 15 '14 at 0:59



On a related or actually, on the completely opposite note -

20

"If you're the user of the aforementioned form and want to re-enable the autocomplete functionality, use the 'remember password' bookmarklet from this [bookmarklets page](#). It removes all `autocomplete="off"` attributes from all forms on the page. Keep fighting the good fight!"

edited Jan 6 '18 at 7:34



[Raghav Dinesh](#)

406 4 12

answered Sep 8 '08 at 19:29



[Antti Kissaniemi](#)

14.9k 13 47 47



96



```
<form name="form1" id="form1" method="post"
      autocomplete="off" action="http://www.example.com/form.cgi">
```

This will work in Internet Explorer and Mozilla FireFox, the downside is that it is not XHTML standard.

edited Jan 6 '18 at 6:36



[Raghav Dinesh](#)

406 4 12

answered Aug 5 '08 at 16:27



[brendan](#)

22.2k 16 61 103

14 Actually @sholsinger, it's best to put it both on the form, AND on the input element itself. That way you cover all the nonstandardness of browsers. – [Avid](#) Dec 13 '10 at 12:11

1 Sadly, as of IE 11, Microsoft no longer respects this for `input type="password"`. Hopefully no other browsers choose to remove this functionality. – [SamHuckaby](#) Mar 21 '14 at 21:18

5

Many modern browsers do not support autocomplete="off" for login fields anymore. autocomplete="new-password" is working instead, more information [MDN docs](#)

answered Dec 21 '17 at 4:33



hien

664 9 9

1

None of the provided answers worked on all the browsers I tested. Building on already provided answers, this is what I ended up with, (tested) on **Chrome 61**, **Microsoft Edge 40** (EdgeHTML 15), **IE 11**, **Firefox 57**, **Opera 49** and **Safari 5.1**. It is wacky as a result of many trials; however it does work for me.

```
<form autocomplete="off">
  ...
  <input type="password" readonly autocomplete="off" id="Password"
onblur="this.setAttribute('readonly');" onfocus="this.removeAttribute('readonly');"
onfocusin="this.removeAttribute('readonly');"
onfocusout="this.setAttribute('readonly');" />
  ...
</form>

<script type="text/javascript">
  $(function () {
    $('#input#Password').val('');
    $('#input#Password').on('focus', function () {
      if (!$('this').val() || $('this').val().length < 2) {
        $(this).attr('type', 'text');
      }
      else {
        $(this).attr('type', 'password');
      }
    });
    $('#input#Password').on('keyup', function () {
      if (!$('this').val() || $('this').val().length < 2) {
```

```
$(this).attr('type', 'text');
}
else {
  $(this).attr('type', 'password');
}
});
$('input#Password').on('keydown', function () {
  if (!$('this').val() || $('this').val().length < 2) {
    $(this).attr('type', 'text');
  }
  else {
    $(this).attr('type', 'password');
  }
});
</script>
```

edited Nov 22 '17 at 15:13

answered Nov 22 '17 at 8:28



zinczinc

480 5 8

So here is it:

8

```
function turnOnPasswordStyle() {
  $('#inputpassword').attr('type', "password");
}
```

```
<input oninput="turnOnPasswordStyle()" id="inputpassword" type="te">
```

Run code snippet

[Expand snippet](#)

edited Jul 20 '17 at 17:07



garfbradaz

1,828 6 30 59

answered May 29 '16 at 19:57



Stav Bodik

567 7 16

-
- 1 Looks like good idea if style text boxes to prevent flash of visible password. – [Andrew](#) Nov 8 '18 at 16:33
-
- 1 Thanks so much, this did the job +1, my variation to your solution was to make it by single line: `<input oninput="this.type='password'" id="inputpassword" type="text">` – [Hasnaa Ibraheem](#) Feb 21 at 10:43
-

Try this :

3

`<input type='text' autocomplete='off' />`

answered May 27 '17 at 8:13



Cool

139 1 8

I use this `TextMode="password" autocomplete="new-password"` and in in page load in aspx `txtPassword.Attributes.Add("value", '');`

4

answered Mar 14 '17 at 9:35

[Mohaimin Moin](#)

617 4 18

1

2

3

[next](#)

protected by [Richard J. Ross III](#) Apr 15 '13 at 20:35

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?