

Autocapitalize for mobile



By Paul Kinlan

Paul is a Developer Advocate

This may look like the most unimpressive feature in existence, but I think it is important because everyone dislikes typing on mobile: You hate it, I loathe it. In Chrome for Android (prior to Chrome 43 - Beta as of April 2015) a developer has little control over how the browser can help the user enter text. If you are typing on a device today, it might look like:

Notice everything is in lowercase apart from some values that Android recognised was a name.

Apple introduced an attribute on `HTMLInputElement` and `HTMLTextAreaElement` called autocapitalize in iOS 5 and it allows the page author to hint at how the browser should present the virtual keyboard for a user to optimize text entry for the user. In its simplest form, you could indicate that a text box should automatically capitalize the first letter of every new sentence.

From Chrome 43, Chrome will support the *autocapitalize* attribute on both `HTMLInputElement` and `HTMLTextAreaElement`, which will allow you to control the autocapitalization behaviour of the virtual keyboard and bring it inline with Safari on iOS.

autocapitalize will only apply to `HTMLInputElement`s that have the *type* attribute set to: `type="text"`, `type="search"`, `type="url"`, `type="tel"`, `type="email"` or `type="password"`. The default is to **not** autocapitalize.

Here's a simple example letting you autocapitalize sentences in a text area:

```
<textarea autocapitalize="sentences">
```

What values can autocapitalize take?

The following table shows the different states that an input element can be in:

State	Keywords

	State	Keywords
<code><input></code> <code><input autocapitalize=off></code>	No Capitalization	none [default]
		off
<code><input autocapitalize=characters></code>	Characters Capitalization	characters
<code><input autocapitalize=words></code>	Words Capitalization	words
<code><input autocapitalize=sentences></code>	Sentences Capitalization	sentences

For `HTMLInputElement`, the invalid value default is *Sentences Capitalization* if the type of the element is `type=text` or `type=search`. Otherwise, it is *No Capitalization*.

- `<input autocapitalize="simon">` would be a text field with *Sentences Capitalization*
- `<input type="email" autocapitalize="simon">` would be a text field with *No Capitalization*.
- `<input>` would be a text field with *No Capitalization*.

For `HTMLTextAreaElement`, the invalid value default is *Sentences Capitalization*. This is a change from the default behaviour.

- `<textarea autocapitalize="terry"></textarea>` would be a text area with *Sentences Capitalization*
- `<textarea></textarea>` would be a text area with *Sentence Capitalization*.
- `<textarea autocapitalize="none"></textarea>` would be a text area with *No Capitalization*.

For `HTMLFormElement` we have decided not to implement the attribute, because we've found that it is rarely used on pages today, and when it is used, it is mostly used to disable autocapitalization on the form entirely:

```
<form autocapitalize=off><input></form>
```

The above is odd, as the default state for `HTMLInputElement` is *No Capitalization*.

Why are you using this over `inputmode`?

`inputmode` is meant to solve the same type of problem, among other things. However, it has been lacking browser implementations — to the best of our knowledge, only Firefox OS has

an implementation and it is prefixed (x-inputmode) — but it also has very little usage on the web. On the other hand, `autocapitalize` is used across millions of pages on hundred of thousands of websites already.

When should I use this?

This isn't an exhaustive list of when you should use `autocapitalize`; however there are a number of places where helping the user enter text provides great value:

- Use `autocapitalization=words` if you are
 - Expecting people's names (note: not all names follow this rule, but the majority of western names will capitalize automatically as expected)
 - Company names
 - Addresses
- Use `autocapitalization=characters` if you are expecting:
 - US states
 - UK postal codes
- Use `sentences` for input elements if you are expecting content that is entered in normal paragraph form - for example, a blog post.
- Use `none` on TextAreas if you are expecting content that should not be affected - for example, entering code.
- If you don't want hinting, don't add `autocapitalize`.

Other interesting links

- [Original implmentation proposal by Mounir Lamouri](#)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.