Quick FAQs on input[type=date] in Google Chrome

By Eiji Kitamura
Developer Advocate in Tokyo

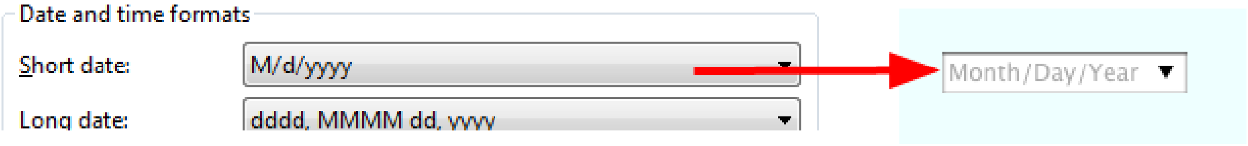
- This article is written by a Chrome software engineer Kent Tamura.

As you might have already noticed, Google Chrome supports a datepicker since Chrome 20. Just by setting the `type` attribute of the `input` element to `date`, the user can click the arrow button and Chrome will pop up a nice calendar widget.

As we have received a lot of feedback from developers, we'd like to clarify a few things about how to get the best out of using the date picker in this article.

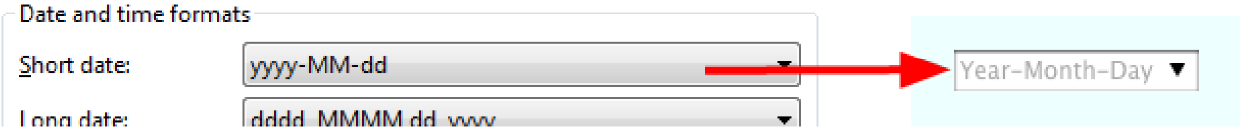
How does locale affect the date format of the input value?

Users can type a date value into the text field of an `input[type=date]` with the date format shown in the box as gray text. This format is obtained from the operating system's setting.



Date and time formats
Short date: M/d/yyyy
Long date: dddd, MMMM dd, yyyy

Month/Day/Year ▼



Date and time formats
Short date: yyyy-MM-dd
Long date: dddd, MMMM dd, yyyy

Year–Month–Day ▼

Web authors have no way to change the date format because there currently is no standards to specify the format.

How is the input value represented when sending to a server?

The date value interpreted from `input[type=date]` in the HTTP Request such as GET / POST will be formatted as `yyyy-mm-dd`.

What kind of format does the input value return?

The `input.value` always returns as `yyyy-mm-dd` regardless of the presentation format.

What kind of format does the input value accept?

When setting the `input.value` programmatically, the value accepts only `yyyy-mm-dd` style regardless of the presentation format for both the initial value and the JavaScript injected value.

How do I change the appearance of the date picker?

You cannot currently style the appearance of the date picker. In WebKit, we have previously provided ways to style form controls with the `-webkit-appearance` CSS property or the `::-webkit-foo` pseudo class selector. However the calendar popup does not provide such ways in WebKit because it is separate from the document, like a popup menu for `<select>`, and there is not currently a standard for how to control styling on its sub-elements.

How do I avoid conflicts between the jQuery Datepicker and the native date picker?

If you have tried jQuery Datepicker on `input[type=date]` in Google Chrome, you might have noticed overlapping calendars of both the jQuery UI and the native calendar popup. If you'd like to apply jQuery Datepicker on all platforms, use `input[type=text]` instead of `input[type=date]`. Not only Google Chrome but also iOS Safari, the BlackBerry browser, and Opera have their own UI for `input[type=date]`, and there is currently no way to achieve a unified UI on all platforms using `input[type=date]`. If you'd like to apply jQuery Datepicker only on platforms without `input[type=date]` support, you may use Modernizr, or the following code:

```
var isDateInputSupported = function(){
var elem = document.createElement('input');
elem.setAttribute('type','date');
elem.value = 'foo';
return (elem.type == 'date' && elem.value != 'foo');
}

if (!isDateInputSupported())  // or.. !Modernizr.inputtypes.date
  $('input[type="date"]').datepicker();
```

Last updated July 2, 2018.