

Form validation

Peter-Paul Koch

<http://quirksmode.org>

<http://twitter.com/ppk>

Browser API Special, 15 June 2017

Form validation

Literally the oldest trick in the JavaScript book. Netscape 2 could do form validation, and little else.

Modern input types

- `type="date"`
- `type="email"`
- `required`
- `pattern`
- and so on

Modern input types

- Automatically validated
- Automatically get native error messages
- Your own validation script only kicks in afterwards
- (Turn this off by setting the novalidate attribute on the form)

Your name

This field is required



Please fill in this field.

Postal code

This field should contain a Dutch postal

Modern techniques

- CSS :valid and :invalid
- Constraint Validation API

Great!

That solves all our problems, right?

Well,

no

And I'll show
you why not

Form validation

Oh, and one thing before we start....

Always
validate your
forms
server-side

Examples

I prepared a few example pages. Find them at

<https://quirksmode.org/forms/>

Sponsored by



Samsung Internet



What do we
want?

Examples

Please view the What We Want example.

This, roughly, is what we would like - details aside.

What, exactly, are we doing?

The ideal form

- Gives visual clue when field is valid or invalid
- Places error messages directly next to the form field
- Validates an individual field onblur.

Validation timing

- onsubmit, i.e. when the user submits the form. Disadvantage: far too late.
- onkeypress, i.e. whenever the user changes part of the value. Disadvantage: too soon, and too many times
- onblur, i.e. when the user leaves a form field. This is the sweet spot.

Validation timing

- ~~onsubmit, i.e. when the user submits the form. Disadvantage: far too late.~~
- ~~onkeypress, i.e. whenever the user changes part of the value. Disadvantage: too soon, and too many times~~
- onblur, i.e. when the user leaves a form field. This is the sweet spot.

Validation timing

- So native CSS/API form validation uses onblur timing, right?
- Funny you should ask. No, it doesn't.
- The API uses onsubmit timing, which is wrong.
- CSS uses onkeypress timing, which is even more wrong.

2

CSS

Examples

<https://quirksmode.org/forms/>

Please view the CSS Validation test page and do the first test.

:valid and :invalid

- :valid
- :invalid

There are a few more, but they're specific cases of :valid and :invalid.

As you see they fire onkeypress; i.e. validity is re-evaluated every time the user changes something.

Also, a form field is valid or invalid. There is no indeterminate state.

:valid and :invalid

Moving to onblur:

```
input:invalid:not(:focus)
```

See the second example.

This postpones visual validation until the user is done. Also, it gives the field an indeterminate style while the user is typing.

The ideal form

- ✔ Gives visual clue when field is valid or invalid
- Places error messages directly next to the form field
- ✔ Validates an individual field onblur.

CSS error messages

How do we do an error message?

```
<input required name="name"  
  data-error="This field is required">  
input:invalid:after {  
  content: attr(data-error);  
  // styling  
}
```

CSS error messages

Alas alas.

input:before and input:after are not supported.

Or rather, they're *supposed* to be not supported.

- Chrome, Safari: works on checkboxes, radios, and ranges
- Chrome, Safari/iOS: also works on date-related types

Why? Beats me.

CSS error messages

How do we do an error message?

```
<input name="name" required>
```

```
<span>This field is required</span>
```

```
input:invalid + span {
```

```
    display: block;
```

```
}
```

Required

<https://quirksmode.org/forms/>

Please view the third example on the CSS Validation test page.

It's invalid before you even do anything.

Required

A required field without a value is invalid. Duh.

Initially, the field has no value. Thus it is invalid.

Before the user has even done anything.

Not good.

Required

Proposed solution:

`input:user-invalid` (W3C)

`input:user-error` (WHATWG)

Means “if the field is invalid after the user has interacted with it.”

Good idea! Let’s hope for a speedy implementation.

:valid and :invalid

- on input
- on form
- on fieldset (not Edge)
- ...
- on label?

:valid and :invalid

- on input
- on form
- on fieldset (not Edge)
- ...
- ~~on label?~~
- Of course not. Why would that work?

Label blocks

Label blocks are so useful!

```
<label>
```

```
  Your name
```

```
  <input name="name" required>
```

```
  <span>This field is required</span>
```

```
</label>
```

Label blocks

Label blocks are so useful!

```
label:invalid span {  
    display: block;  
}
```


CSS form validation

- Kind-of OK
- need to be triggered onblur, and not onkeypress
- need :user-invalid
- need label:invalid and label:valid
- take a bloody decision on input:before/after
- indeterminate state

3 Constraint Validation API

The ideal API

- Figure out if a field is valid or invalid, and why
- Rewrite a native error message
- Show a native error message

The ideal API

- Figure out if a field is valid or invalid, and why
- Rewrite a native error message
- Show a native error message

Checking validity

- validity properties, which tell you exactly what is wrong
- `checkValidity()`, which does not tell you what's wrong

Checking validity

- validity properties, which tell you exactly what is wrong
- ~~checkValidity(), which does not tell you what's wrong~~

validity properties

`field.validity.patternMismatch`

- `patternMismatch`
- `rangeOverflow`, `rangeUnderflow`
- `stepMismatch`
- `tooLong`, `tooShort`
- `typeMismatch`
- `valid`
- `valueMissing`

The ideal API

- ✔ Figure out if a field is valid or invalid, and why
- Rewrite a native error message
- Show a native error message

Rewriting error messages

- `setCustomValidity('New message')` rewrites the error message
- BUT it also sets the field to invalid, regardless of its value.
- `setCustomValidity("")` resets the error message and removes the invalid lock
- `setCustomValidity()`, without argument, gives an error. You clearly need that empty string here ...

Rewriting error messages

- title=“Please fill in a valid Dutch postal code”
- works only on fields with a pattern attribute
- and then it **ADDS** the title text to the error message instead of replacing it

The ideal API

- ✔ Figure out if a field is valid or invalid, and why
- ~~Rewrite a native error message~~
- Show a native error message

4 Native error messages

Your name

This field is required



Please fill in this field.

Postal code

This field should contain a Dutch postal

Examples

<https://quirksmode.org/forms/>

Please view the Native Error Messages test page for an example.

Showing error messages

- When the user submits the form, but only the first error
- When `reportValidity()` is called on a form field
- When the field has been found invalid before and the user focuses (Edge only)
- The field automatically gains the focus.
- Onsubmit in Edge and Firefox, all invalid fields get a red outline.

Triggering errors

- We want to call it onblur, and not onsubmit.
- So we have to use reportValidity() (which is not supported by Edge, but let's ignore that for now)
- So far so not too bad.

onblur and reportValidity()

So this is what we need:

```
form.addEventListener('blur',  
  function (e) {  
    if (e.target.nodeName === 'INPUT') {  
      e.target.reportValidity();  
    }  
  }, true);
```

onblur and reportValidity()

So this is what we need:

```
form.addEventListener('blur',  
  function (e) {  
    if (e.target.nodeName === 'INPUT') {  
      e.target.reportValidity();  
    }  
  }, true);
```

Alas alas.

Massive
EPIC Safari/
Chrome fail

onblur and reportValidity()

<https://quirksmode.org/forms/>

Please view the Blur and reportValidity test page in Safari or Chrome.

Focus on one field. Don't change it.

Then focus on another field.

You can't.

Whut?

Hiding error messages

- when the user scrolls (Edge, Firefox, Safari)
- after 5 seconds (Chrome)
- when the field receives the focus (all)

But not

- when the field becomes valid

In fact, why would we want the error message to disappear at all before the user corrects the mistake?

Because browsers are weird.

onblur and reportValidity()

- onblur, the script is triggered
- reportValidity() fires and finds the field is invalid
- A native error message is generated
- and the field receives the focus that it just lost
- and the error message disappears

Native error
messages are
useless

Not showing error messages

- `<form novalidate>`
- `field.oninvalid = function () {return false}`

Invalid event

- The invalid event?
- Yup. The invalid event.
- Fires whenever a field is found to be invalid during the submission process
- or with `checkValidity()` or `reportValidity()`
- And its default action is showing the native error message.

Invalid event

- Actually, the invalid event is a pretty good idea.
- And you know what would also be a good idea?
- A valid event that fires whenever a form field is found to be valid
- Who here thinks browsers support it?

5

What are
we to do?

What can we use?

- `:valid` and `:invalid`, provided we use the `:not(:focus)` trick.
- `validity.valid` and friends
- `novalidate` and/or the `invalid` event, so we can turn off native error messages

HTML

```
<form novalidate>  
  <label>  
    Your name  
    <input name="name" required>  
    <span>This field is required</span>  
  </label>  
</form>
```

CSS

```
label.valid {}
```

```
label.valid input {}
```

```
label.valid span {}
```

```
label.invalid {}
```

```
label.invalid input {}
```

```
label.invalid span {}
```

JavaScript

Suppress native error messages

```
form.addEventListener('invalid',  
    function (e) {  
        e.preventDefault()  
    }, true);
```

JavaScript

Trigger validation onblur

```
form.addEventListener('blur',  
    function (e) {  
        var tgt = e.target;  
        if (tgt.nodeName === 'INPUT') {  
            validateField(tgt);  
        }  
    }, true);  
form.onsubmit = validateAll;
```


JavaScript

Actual validation

```
if (!field.validity.valid) {  
    // set class to invalid  
} else if (field.validity.valid  
    && field.value) {  
    // set class to valid  
} else {  
    // remove both classes  
}
```

To be done

- My third article (see the example page) contains a long list of features that should be added or improved

Thank you

I'll put these slides online

Questions?

Peter-Paul Koch

<http://quirksmode.org>

<http://twitter.com/ppk>

Browser API Special, 15 June 2017