BRISTOL JS

HTML5 CONSTRAINT VALIDATION

@drewm

FORMS ARE HARD.

YOU'RE LAZY AND YOUR CODE SUCKS.

BROWSERS ARE HERE TO SAVE US.

THE PREMISE

1. Users enter weird shiz into forms
2. So user input needs to be validated
3. It's faster and more user-friendly to do that in the browser first
4. You're lazy and your code sucks
5. HTML5 introduced built-in validation for forms

HTML5 CONSTRAINT VALIDATION

```
<form method="post" action="/cgi-bin/login.pl">

  <label>Username
    <input type="text" name="username" required>
  </label>

  <label>Password
    <input type="password" name="password" required>
  </label>

  <input type="submit" value="Log in">

</form>
```

HTML5 CONSTRAINT VALIDATION

```
<form method="post" action="/cgi-bin/login.pl">

  <label>Username
    <input type="text" name="username" required>
  </label>

  <label>Password
    <input type="password" name="password" required>
  </label>

  <input type="submit" value="Log in">

</form>
```

HTML5 CONSTRAINT VALIDATION

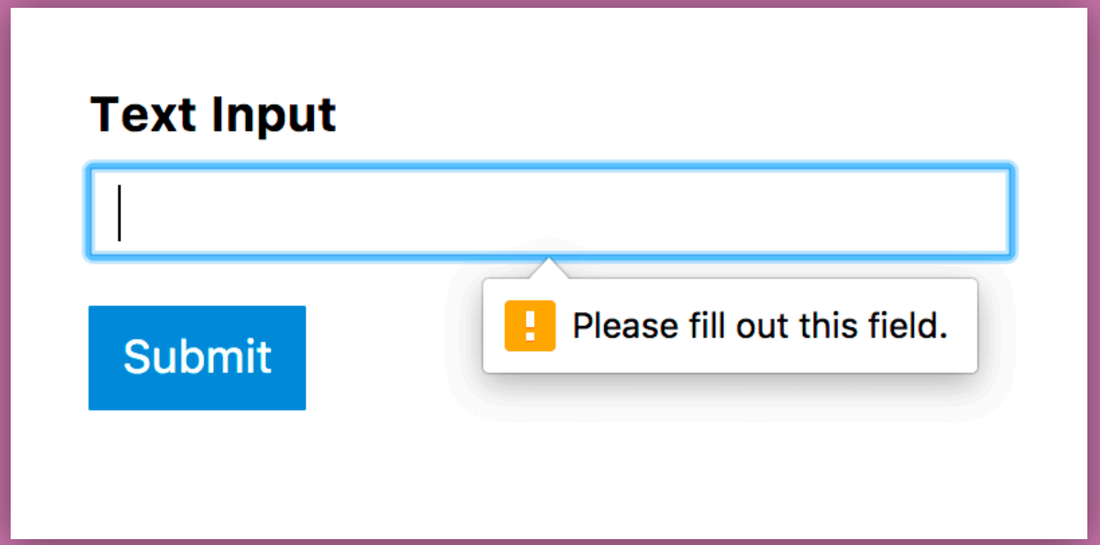


Image: css-tricks.com

HTML5 CONSTRAINT VALIDATION

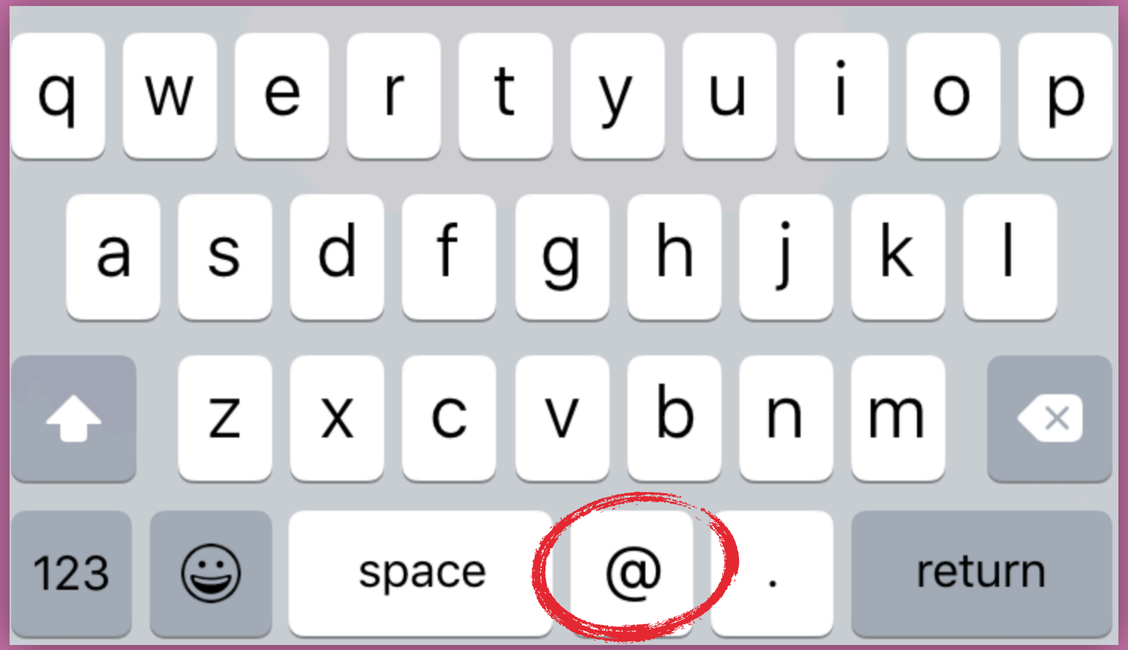
```
<form method="post" action="/cgi-bin/login.pl">

  <label>Email
    <input type="email" name="email" required>
  </label>

  <label>Password
    <input type="password" name="password" required>
  </label>

  <input type="submit" value="Log in">

</form>
```

HTML5 CONSTRAINT VALIDATION

```
<form method="post" action="/cgi-bin/login.pl">

  <label>Email
    <input type="email" name="email" required>
  </label>

  <label>Password
    <input type="password" name="password" required>
  </label>

  <input type="submit" value="Log in">

</form>
```

HTML5 CONSTRAINT VALIDATION

HTML5 CONSTRAINT VALIDATION



BUILT-IN VALIDATION

1. Enforces required fields
2. Prevents mistakes by customising UI to common field types
3. Validates the format of common field types
4. Can be customised with advanced regex patterns
5. Can be styled with CSS

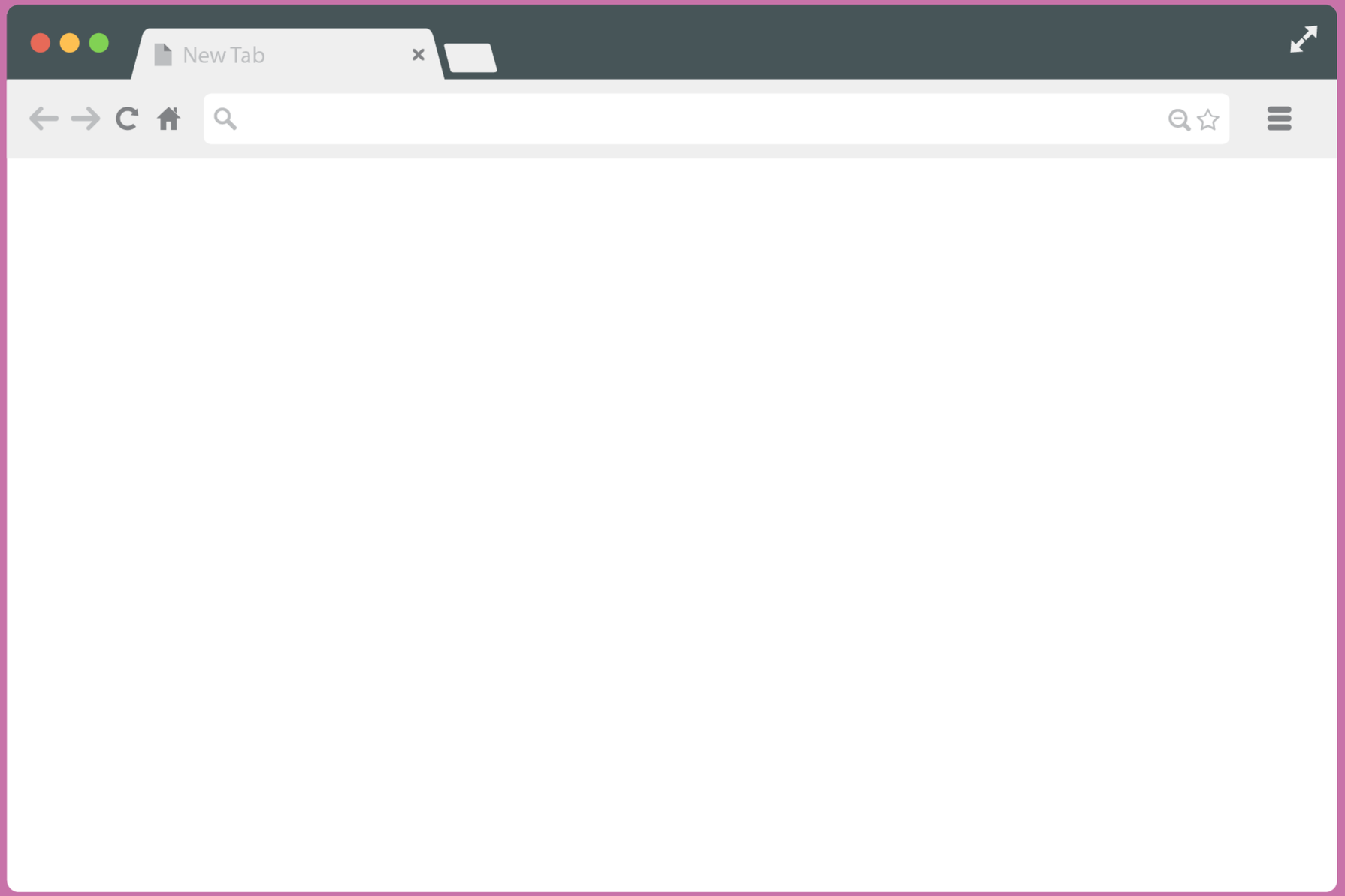
HTML5 CONSTRAINT VALIDATION

```css
input:valid {
  border: 2px solid green;
}

input:invalid {
  border: 2px solid red;
}
```

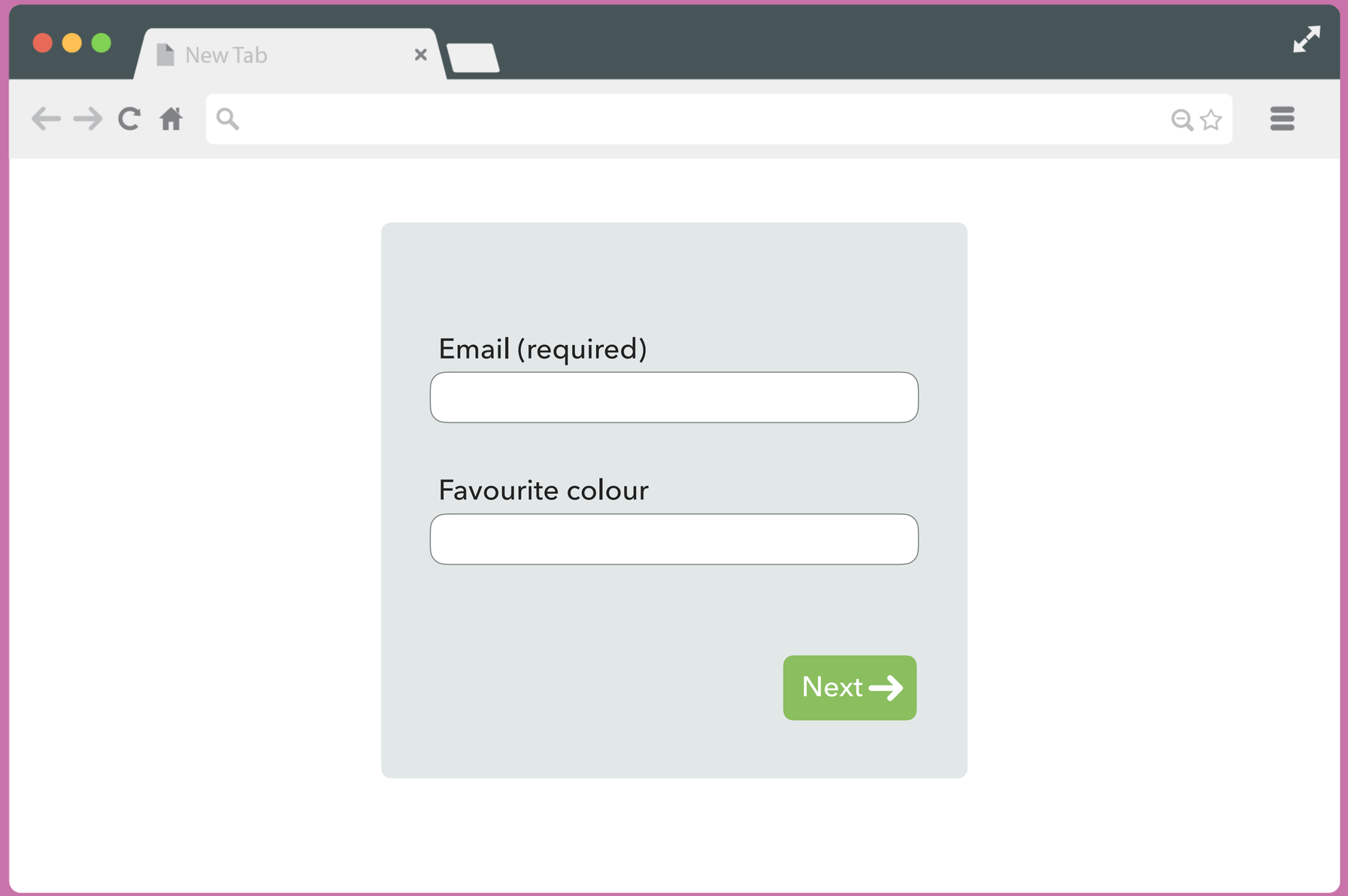
CSS PSEUDO CLASSES

```
:valid

:invalid

:in-range

:out-of-range

:required

:optional
```

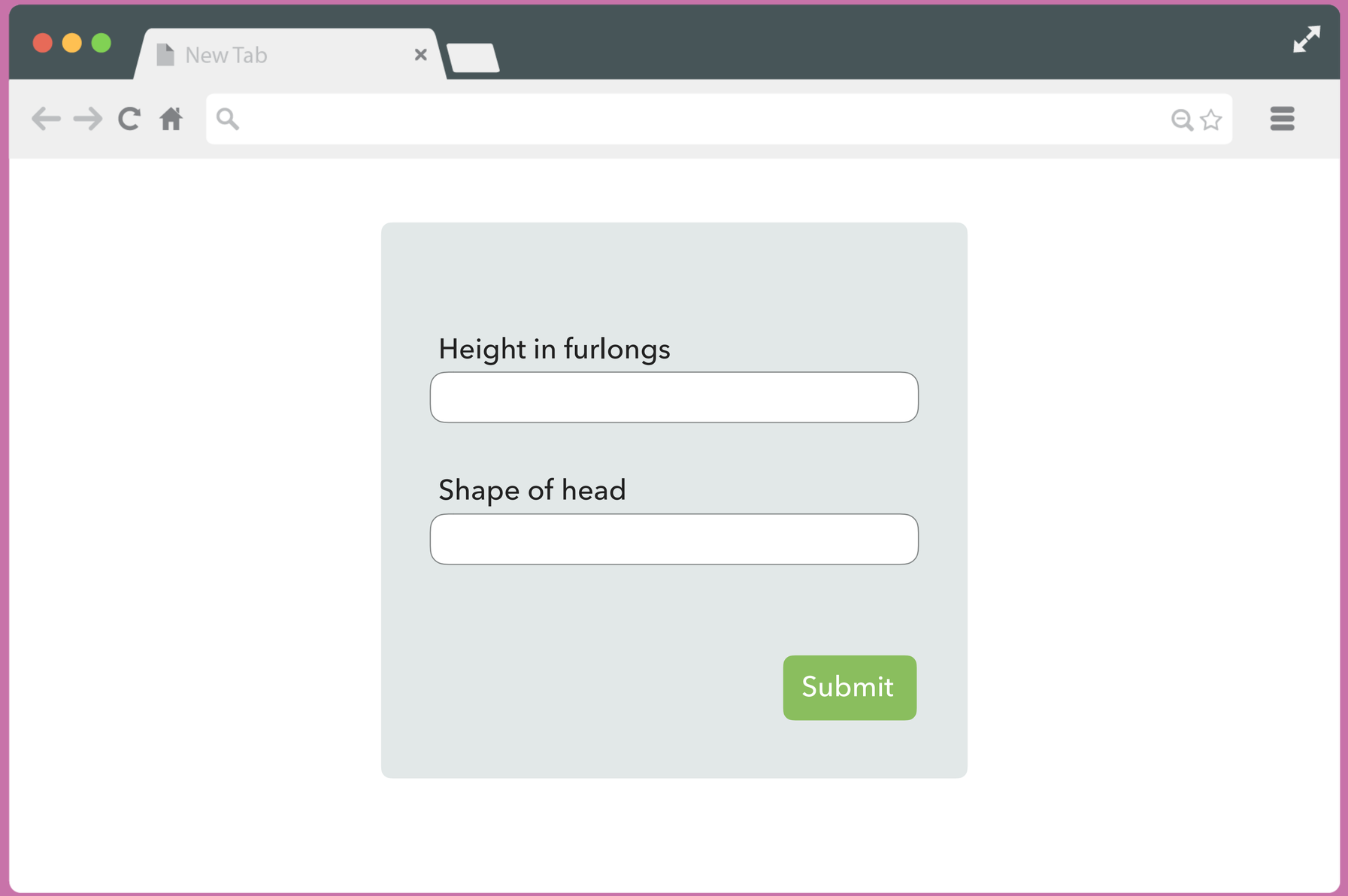
HTML5 CONSTRAINT VALIDATION



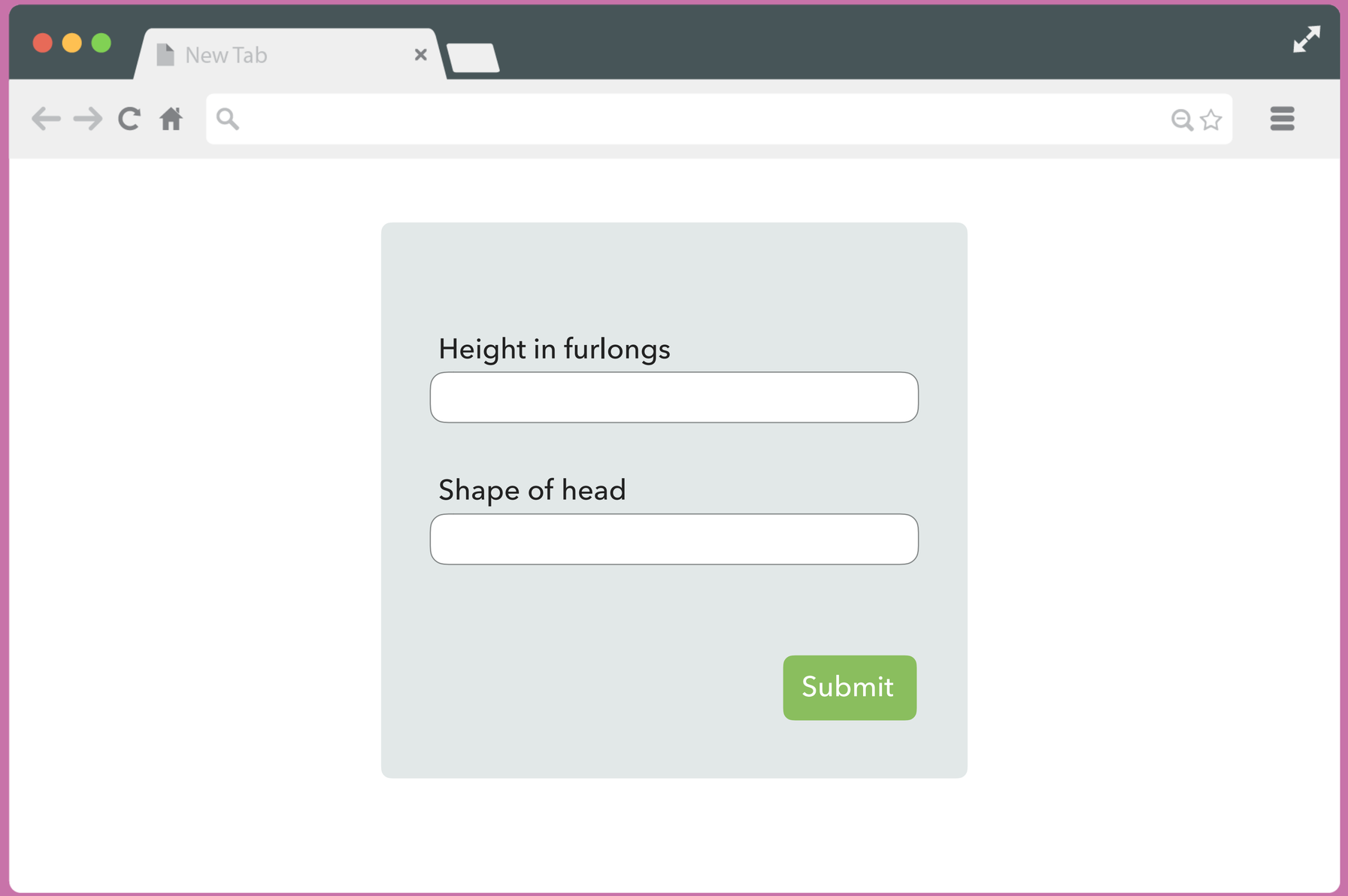
HTML5 CONSTRAINT VALIDATION



HTML5 CONSTRAINT VALIDATION



HTML5 CONSTRAINT VALIDATION



novalidate

HTML5 CONSTRAINT VALIDATION

```
<form method="post" action="/whatevs" novalidate>

  <label>Important information
    <input name="important" type="text" required>
  </label>

</form>
```

HTML5 CONSTRAINT VALIDATION

```
<form method="post" action="/whatevs" novalidate>

  <label>Important information
    <input name="important" type="text" required>
  </label>

</form>
```

DISABLING VALIDATION

1. It's very tempting to turn off validation
2. In doing so you lose a lot of functionality
3. You then need to implement that functionality yourself
4. You're lazy and your code sucks
5. Better to work with it

JAVASCRIPT API

VALIDATION API

- Enables us to ascertain the validity state of a field
- We get to find out not only if it's valid/invalid, but also why
- We can then update our interface accordingly
- Write the bespoke parts and use the browser for the boring stuff
- Because you're lazy and your code sucks

HTML5 CONSTRAINT VALIDATION

```
var myField = document.querySelector('input[type="text"]');
var validityState = myField.validity;

console.log(validityState); // ValidityState object
```

VALIDITY STATE OBJECT

- `valid` - passes validation
- `valueMissing` - required field is empty
- `typeMismatch` - email or url field value is not the correct type
- `tooShort` - minLength attribute not satisfied
- `tooLong` - maxLength attribute not satisfied

VALIDITY STATE OBJECT

- `patternMismatch` - **pattern attribute not satisfied**
- `badInput` - **non-numeric value in a number field**
- `stepMismatch` - **value does not match the step attribute**
- `rangeOverflow` - **value is heigher than the max attribute**
- `rangeUnderflow` - **value is lower than the min attribute**

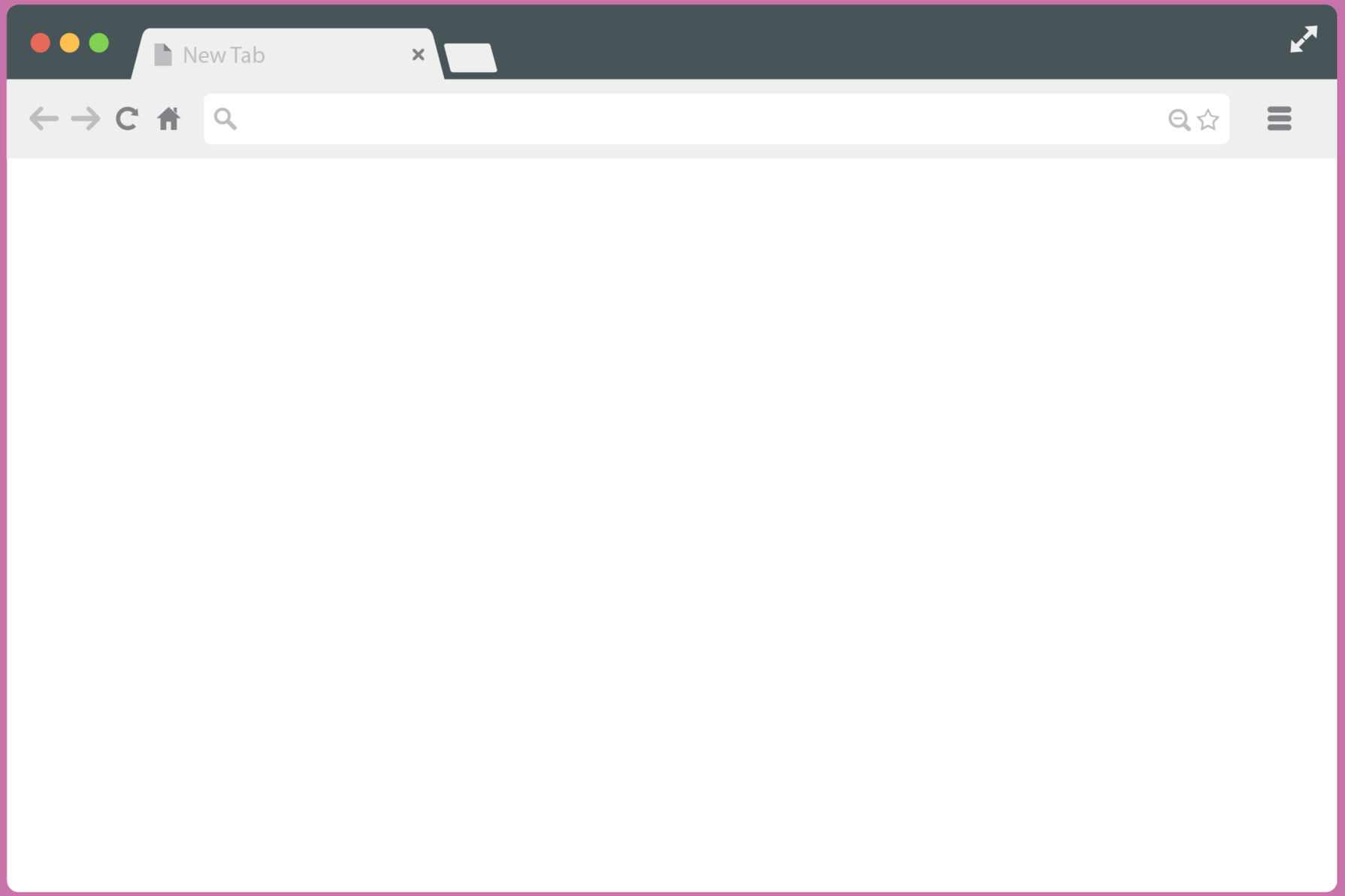
HTML5 CONSTRAINT VALIDATION

```
var myField = document.querySelector('input');

myField.onchange = function(){
  var state = myField.validity;

  if (state.valid) {
    console.log('Valid!');
  }

  if (state.valueMissing) {
      console.log('Required field empty!');
  }

  if (state.typeMismatch) {
      console.log('Incorrect value in %s field', myField.type);
  }
};
```

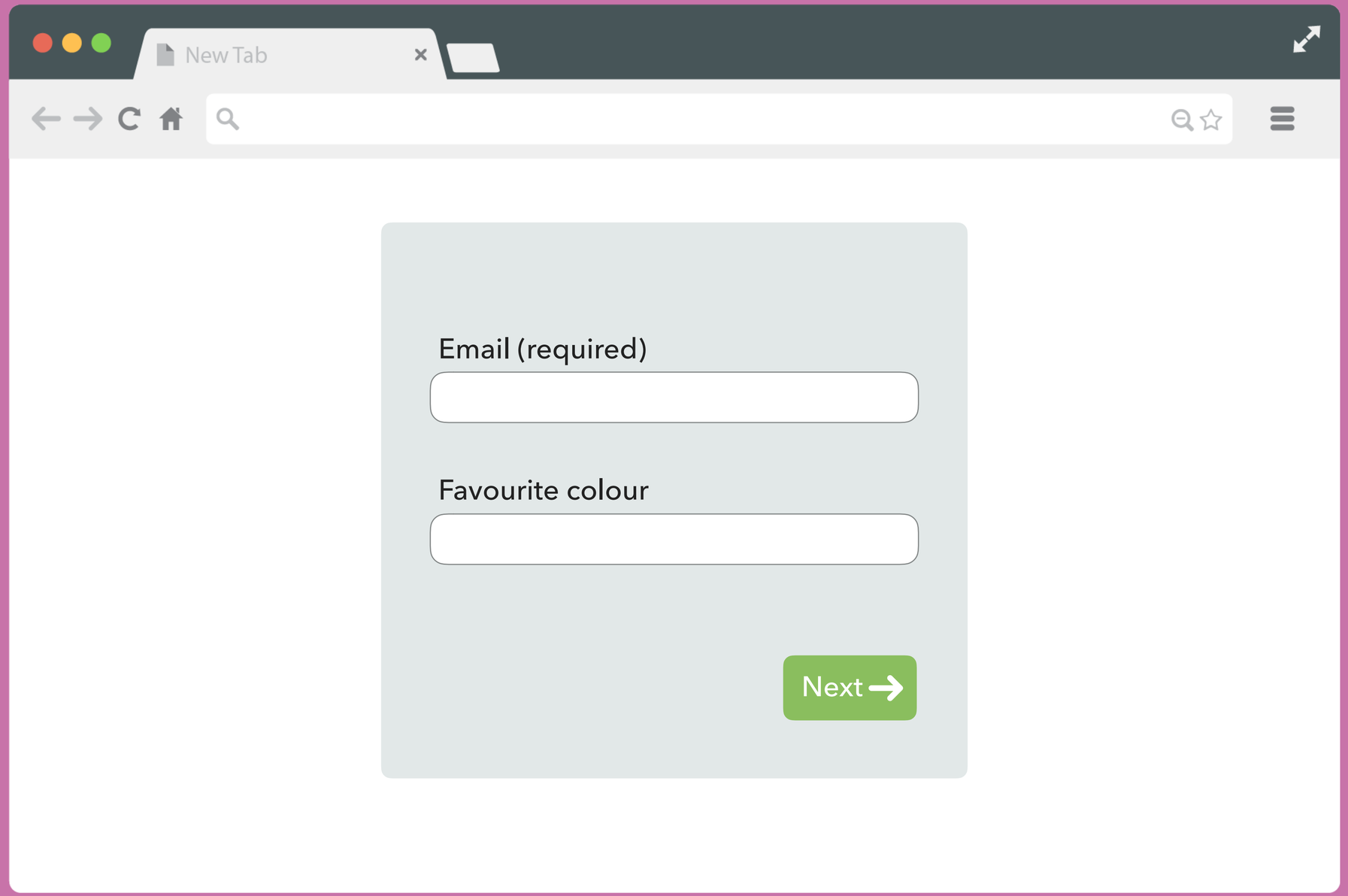
VALIDATION API

- What you do with the information is up to you
- Offers the opportunity to customise the validation UI
- That's great because the browser UI is ugly

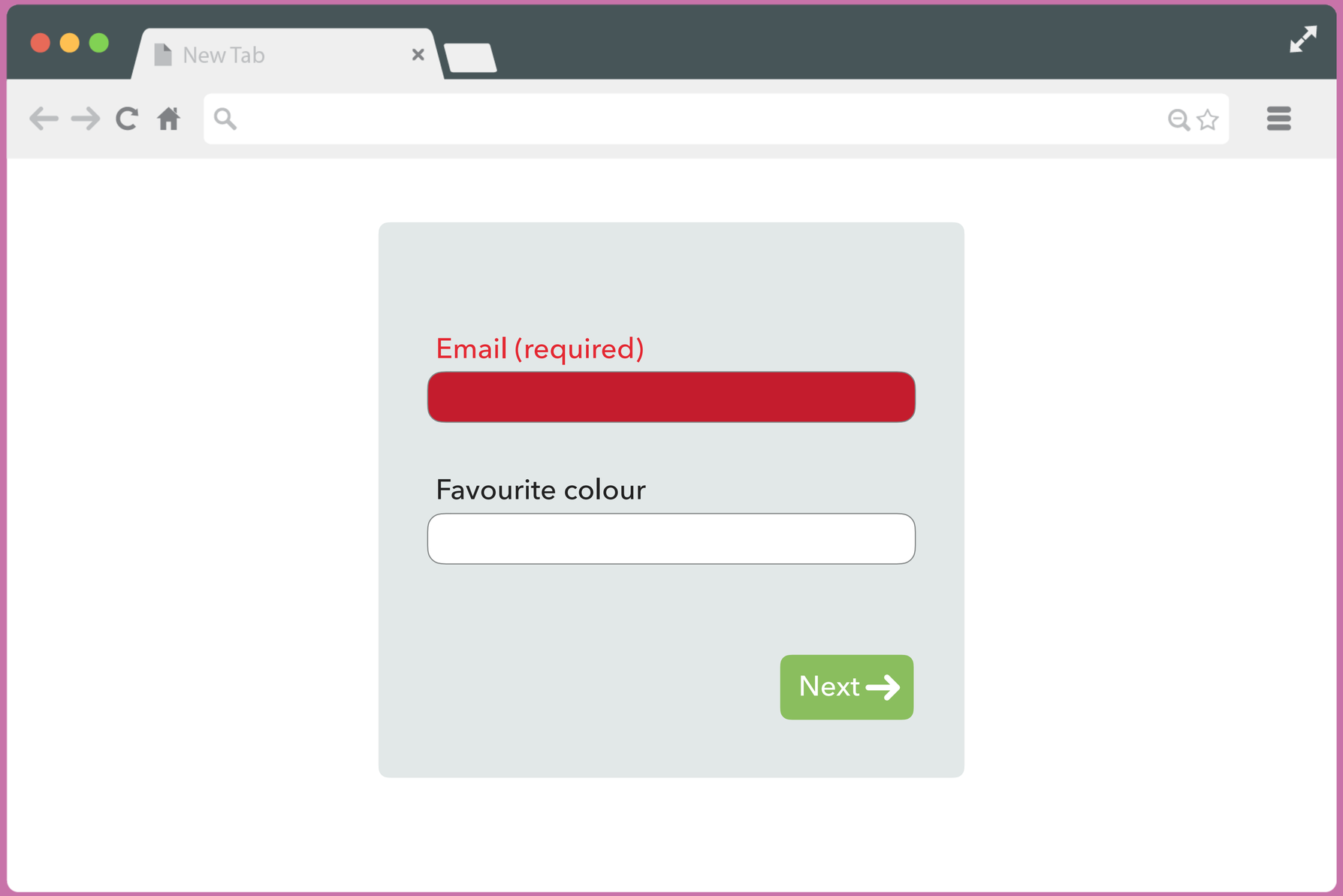
HTML5 CONSTRAINT VALIDATION



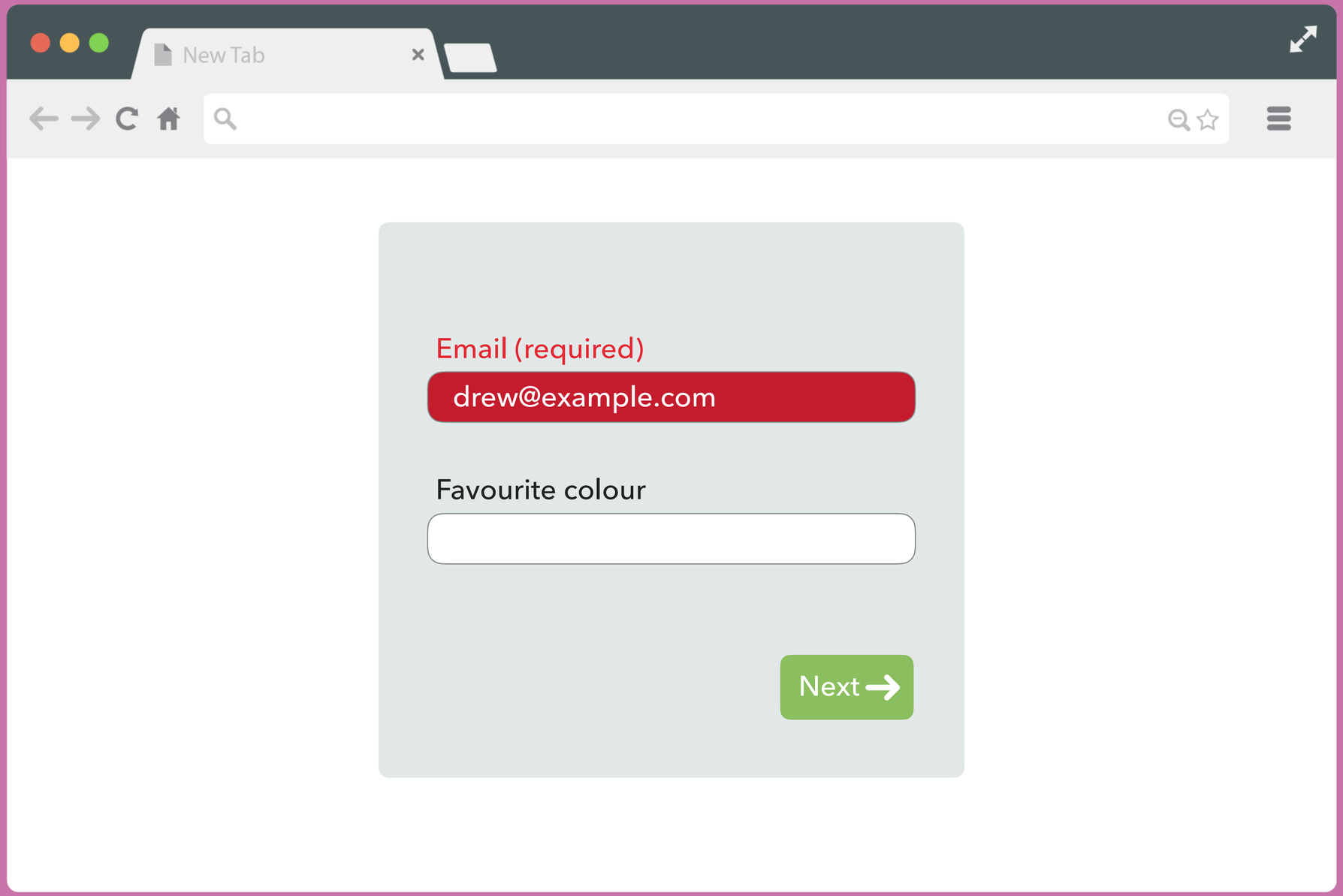
HTML5 CONSTRAINT VALIDATION



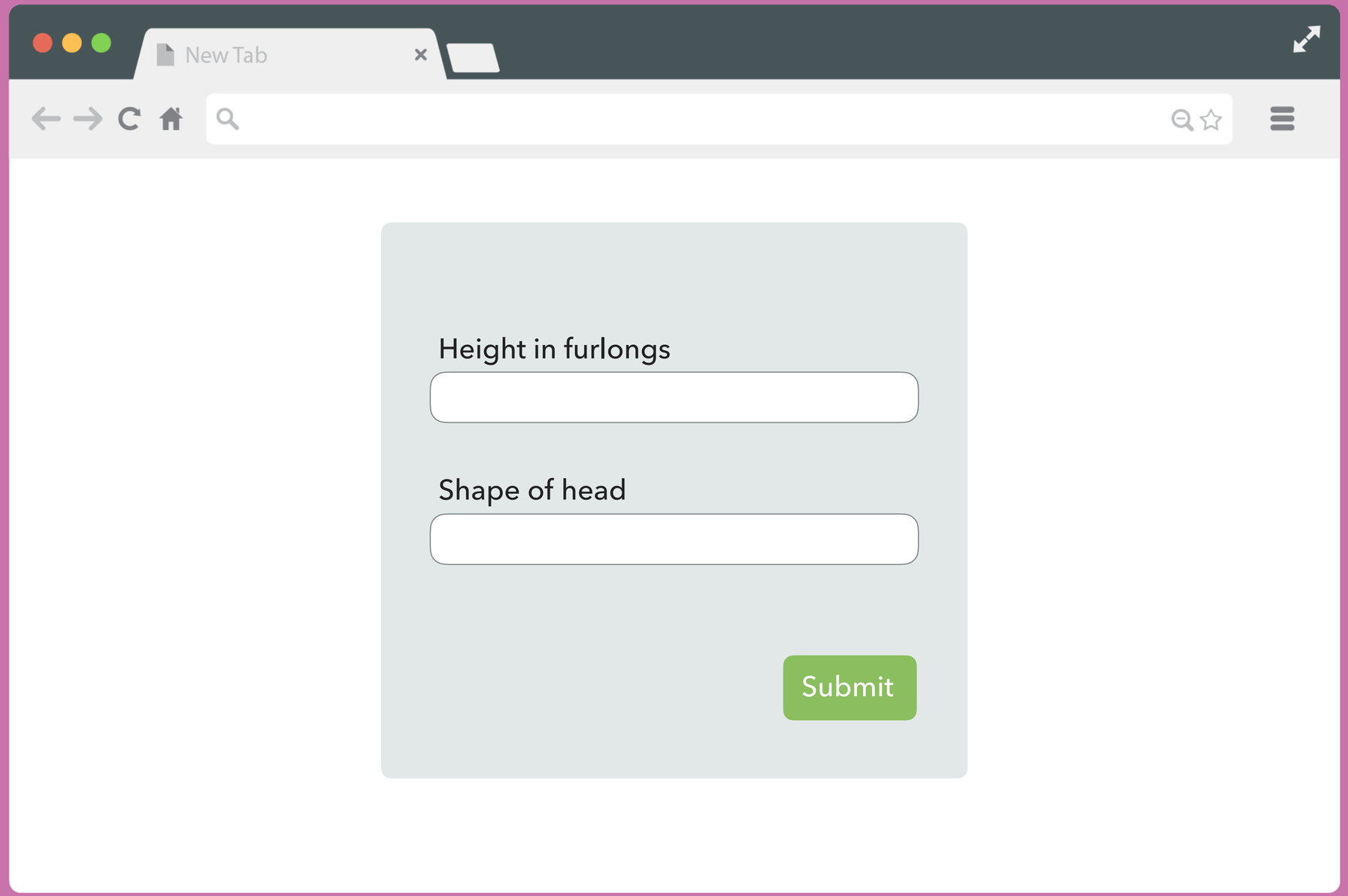
HTML5 CONSTRAINT VALIDATION



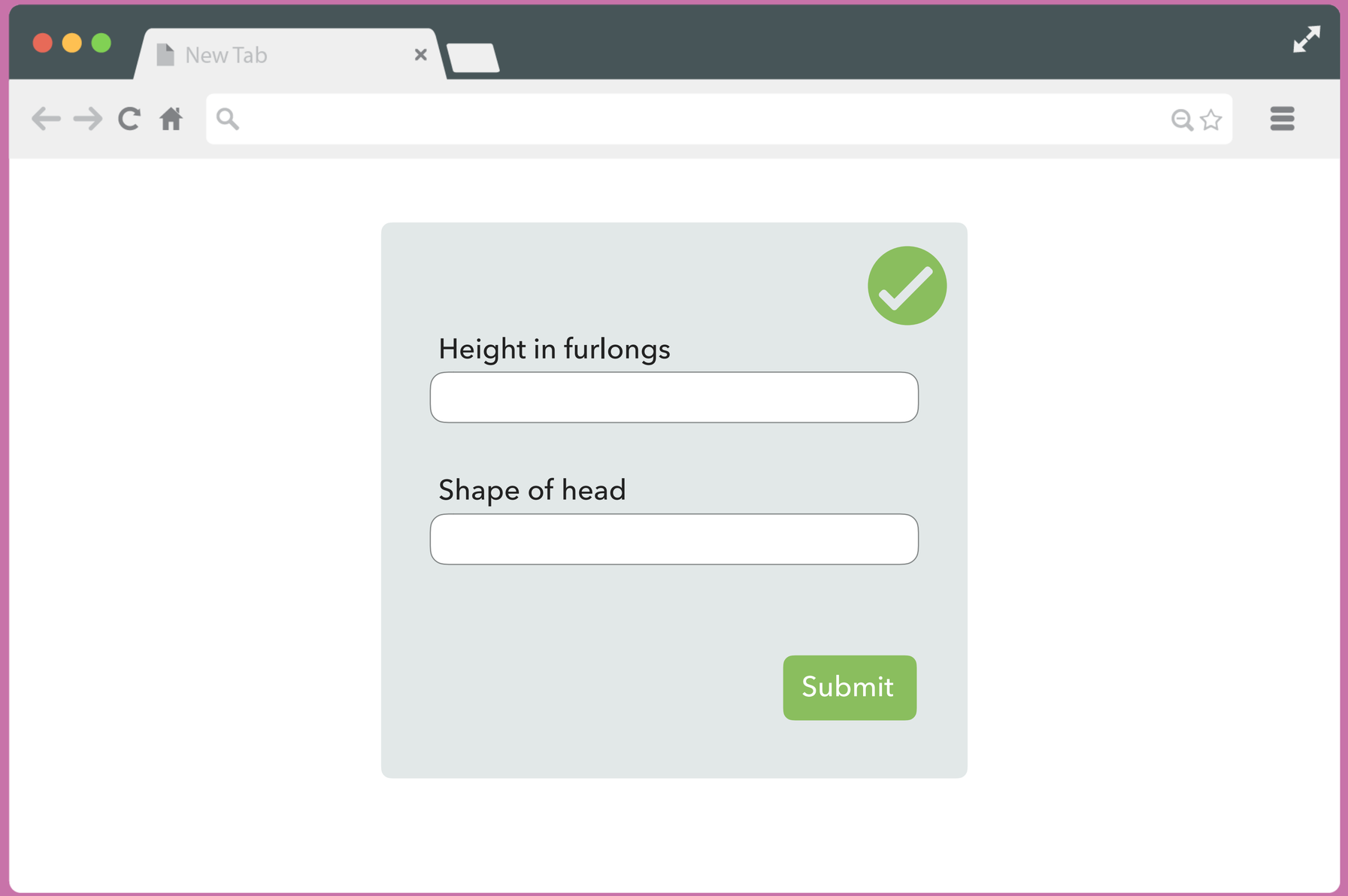
HTML5 CONSTRAINT VALIDATION



HTML5 CONSTRAINT VALIDATION



HTML5 CONSTRAINT VALIDATION



IT'S BRILLIANT.

IT'S BRILLIANT.

Search developer.mozilla.org for “constraint validation”

THANKS.

@drewm