

15일 (일)

1. Codecademy - Building Interactive Websites - HTML Forms

Lesson 1. Introduction to Form Validation

목차

- Intro
- Why Validate Forms?
- Regular Expressions
- Client-side Validation : HTML
- Client-side Validation : Javascript
- Back-end Validation
- Review

- **Intro**

유저에 대한 데이터들을 모으고 그에 대한 유효성을 체크하는 공간을 HTML form 이라고 한다. 제출된 유저에 대한 정보를 체크하는 과정을 form validation checking(폼 유효성 검증) 이라고 한다.

이번 레슨에서는 input 폼에 입력된 데이터를 검증하는 법을 배울 것이다.

- **Why Validate Forms?**

입력되어 제출된 대부분의 데이터는 웹사이트 또는 애플리케이션에 의해 저장된다. 이는 데이터베이스 내부에 저장되는데 이러한 데이터가 정확하게 저장되는지 잘 확인해야하는 이유들이 몇가지 있다.

만약 다른 사람들의 아이디가 같도록 설정하게 둔다면, 수많은 에러들을 초래할 것이며, 유효하지 않는 신상으로 가입을 하게 된다면 많은 문제를 야기할 것이다. ㄷ

또한 우리는 이용자들의 개인정보를 빼가는 해커들로부터 보호해야할 의무를 가지고 있으며, 유효하지 않은 이용자가 우리 서버에 침투하여 어플리케이션을 망치는 일을 일어나지 않도록 해야한다. 그러기 위해서는 유효성 검사가 중요하다.

- **Regular Expressions (정규표현식)**

- ABCDEF2GHIJKLMNOPQRSTUVWXYZ
- My zip code is 9021
- The ct meowed
- `<h1> Hello, World! </h2>`

인간은 직감적으로 위 문자열에 대한 규칙 및 에러를 바로 발견할 수 있지만, 컴퓨터는 그러한 능력을 갖고 있지않다. 그렇기 때문에 우리는 컴퓨터가 해당 문자열에 대해 규칙과 패턴을 파악할 수 있도록 정규표현식을 사용해야한다.

Enter A Pattern And See What Matches:

- ❌ hello
- ❌ hello world
- ❌ Hello, world!
- ❌ Hello, Friend!
- ❌ coolEmail@blorglax.com
- ❌ P@55w0rd!
- ❌ (617) 222-1247
- ❌ +1-341-554-3910
- ❌ 1630 Revello Drive, Sunnydale, California
- ❌ 3170 West 53 Road, #35, Annapolis, Maryland

Enter A Pattern And See What Matches:

- ❌ **hello**
- ❌ **hello** world
- ❌ **Hello**, world!
- ❌ **Hello**, Friend!
- ❌ coolEmail@blorglax.com
- ❌ P@55w0rd!
- ❌ (617) 222-1247
- ❌ +1-341-554-3910
- ❌ 1630 Rev**ello** Drive, Sunnydale, California
- ❌ 3170 West 53 Road, #35, Annapolis, Maryland

Enter A Pattern And See What Matches:

- ✓ hello
- ✓ hello world
- ✓ Hello, world!
- ✓ Hello, Friend!
- ✓ coolEmail@blorglax.com
- ✓ P@55w0rd!
- ✓ (617) 222-1247
- ✓ +1-341-554-3910
- ✓ 1630 Revello Drive, Sunnydale, California
- ✓ 3170 West 53 Road, #35, Annapolis, Maryland

정규표현식중 하나의 명령어이다.

Enter A Pattern And See What Matches:

- ✗ hello
- ✗ hello world
- ✗ Hello, world!
- ✗ Hello, Friend!
- ✗ coolEmail@blorglax.com
- ✗ P@55w0rd!
- ✓ (617) 222-1247
- ✓ +1-341-554-3910
- ✗ 1630 Revello Drive, Sunnydale, California
- ✗ 3170 West 53 Road, #35, Annapolis, Maryland

- Client-side Validation : HTML

폼 데이터 유효성 검사를 위한 첫번째 기술은 클라이언트 입력 폼에서 문제의 여지가 있는 입력에 대한 유효성을 서버에 제출하기 전에 검사하는 것이다. 이것을 client-side validation 이라고 한다. HTML 은 유효성 검증을 위한 코드를 제공하며 이를 사용하여 서버와 상호작용 없이 효율적으로 미리 유효하지않은 입력폼, 데이터를 방지할 수 있다.

```
1  <!DOCTYPE html>
2  ▼ <html lang="en" dir="ltr">
3  ▼   <body>
4      <h1>Basic HTML Validation</h1>
5  ▼   <form action="" method="POST">
6      <label for="text">Enter your name here:</label>
7      <input id="name" name="name" type="text" required
        minlength="3" maxlength="100">
8      <br><br>
9      <label for="number">Enter your age here:</label>
10     <input type="number" name="age" id="age" required
        min="1" max="123">
11  ▼     <br><br>
12     <label for="code">Best place to learn to code: (hint:
        starts with a "C")</label>
13     <input id="code" name="code" type="text" required pattern="
        ▼ [cC]odecademy">
14     <br><br>
15     <input type="submit" value="Submit">
16   </form>
17 </body>
18 </html>
```

이처럼 html 파일내에서 유효성에 대한 코드를 작성할 수 있다.

- required : 필수적으로 입력해야하는 폼을 설정할 수 있다.
- name input : 3부터 100 사이의 길이로 제한한다.
- age input : 1 부터 123 사이의 숫자로 제한한다.

- code input : 정규표현식을 이용하여 Codecademy 또는 codecademy 만 입력 가능하도록 제한한다.

Basic HTML Validation

Enter your name here:

Enter your age here:



이 입력란을 작성하세요.

Best place to learn to code: (hint: starts with a "C")

Submit

• Client-side Validation : Javascript

클라이언트 유효성 검증은 두가지의 이점을 가지고 있다. 첫번째 이점은 유저에게 즉시 오류에 대한 문제점을 제공해줌으로써 기다리지 않고 시간을 아낄 수 있다는 이점이다. 두 번째 이점은 서버와의 상호작용 없이 자원을 아껴서 더 빠른 애플리케이션을 구성할 수 있다는 것이다. 하지만 모든 이슈들은 HTML 의 빌트인 프로퍼티로 해결될 수 없다.

더욱 더 복잡한 룰들은 자바스크립트를 이용하여 설정해야한다. 보통 js 를 이용한 유효성 검증은 바닐라 js 로 구성이 가능하지만 비효율적이며 이미 만들어져있는 좋은 js 기본 유효성 라이브러리를 이용한다. 하지만 DOM 에 직접적으로 접근하고 조작하기 때문에 가상 DOM 을 이용하는 프레임워크에는 적합하지 않다. 그러한 프레임워크와 어울리는 라이브러리는 React 프레임워크와 같이 쓰이는 formik library 가 있다.

Rules:

Please enter a password. It must:

- Be between 5 and 20 characters in length.
- Use at least one letter.
- Use at least one number.
- Use at least one kind of bracket.
- Not contain the word "password".
- Definitely contain the word "validation".

Let's Go!

Enter your preferred password:

- The password MUST contain the word "validation".
- The password MUST contain contain brackets.
- The password MUST contain a letter.

Re-enter the password:

- The two passwords should be the same.

Problem!

This password submission seems to be invalid :(

• Back-end Validation

프론트엔드에서 유효성을 통과하였더라도 해커들은 이러한 방어벽을 충분히 뚫을 수 있으므로, 백엔드에서도 유효성 검증은 필수적이다. 백엔드에서의 검증에는 여러가지 이점이 있다.

⇒ 기계의 훨씬 더 컴퓨팅한 힘을 가진 유효성 코드를 사용할 수 있다.

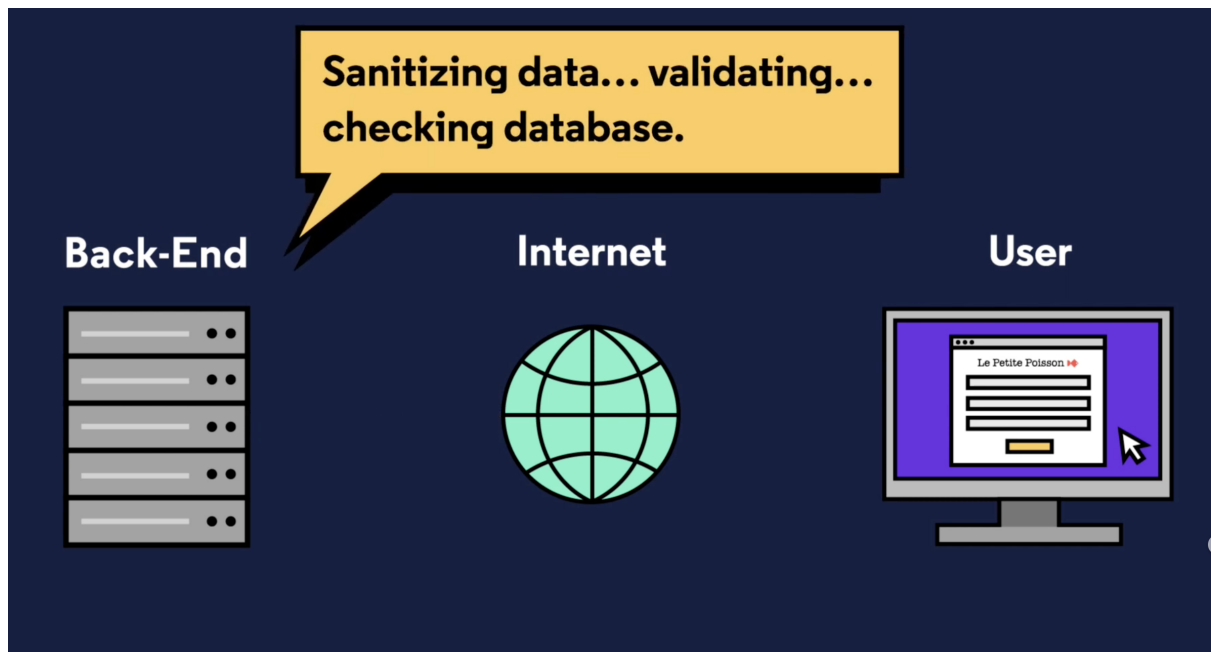
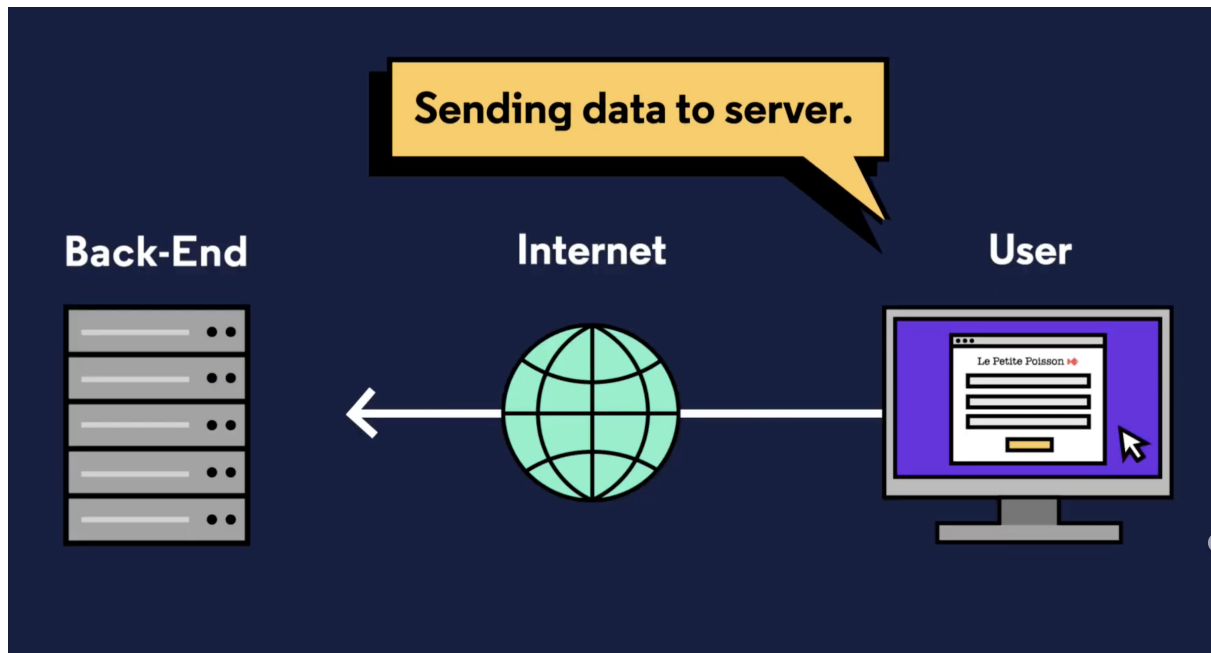
⇒ 유저가 볼 수 없는 유효성 코드를 사용할 수 있다. 즉, 해커는 이러한 코드를 볼 수 없기에 접근하기 힘들다.

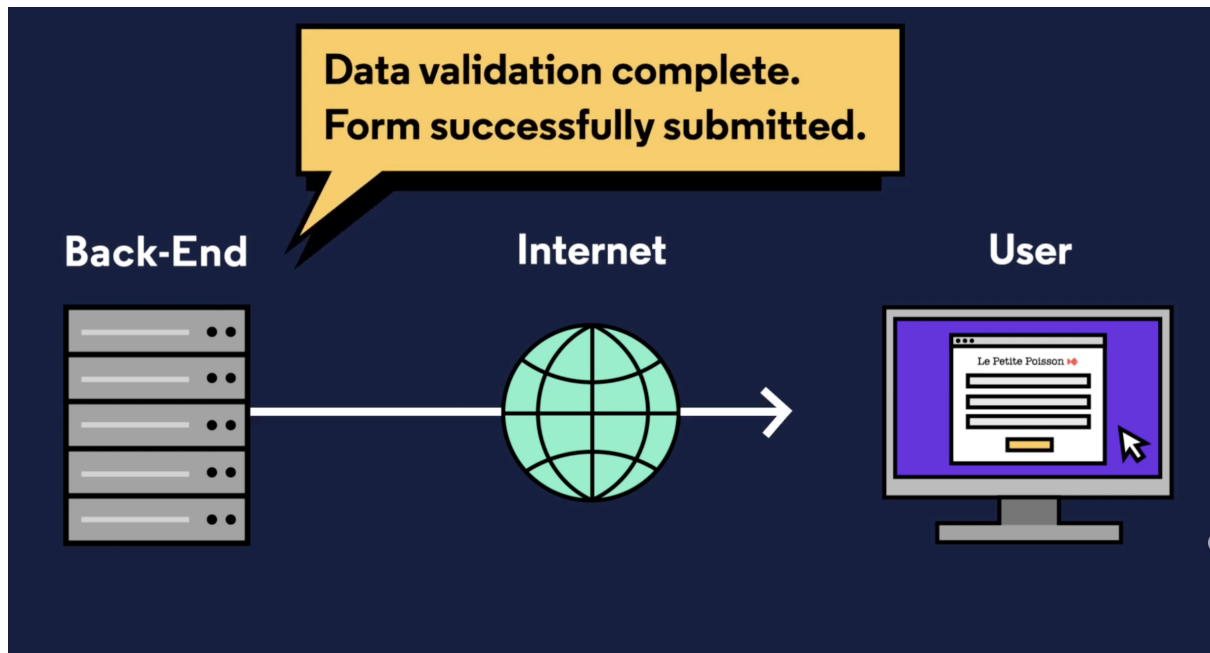
⇒ 데이터베이스와 같은 프론트엔드에서 접근하지 못하는 데이터 정보를 검사할 수 있다.

백엔드에서 제출된 데이터에 대해 소독을 할 수 있다. 데이터들은 일관적인 안전한 형식으로 변형할 수 있다.



The image shows a web browser window with a purple background. The browser's address bar displays the URL `www.lepetitepoisson.com`. The page content includes the title "Le Petite Poisson" followed by a red fish icon. Below the title, there are three form fields: "Name:" with the value "Jacque Cousteau", "Reservation Size:" with the value "4", and "Reservation Time:" with the value "9:30pm". At the bottom of the form is a yellow button labeled "Reserve". A white mouse cursor is visible on the right side of the page.





오늘의 단어

- malicious : 악의적인
- problematic : 문제적인 , 문제가 많은
- relatively : 비교적