

# 16일 (월)

## 1. Codecademy - Building Interactive Websites - HTML Forms

### Lesson 2. HTML Forms

#### 목차

- Intro to HTML Forms
- How a Form Works
- Text Input
- Adding a Label
- Password Input
- Number Input
- Range Input
- Checkbox Input
- Radio Button Input
- Dropdown list
- Datalist Input
- Textarea element
- Submit Form
- Review

- Intro to HTML Forms



우리의 일상에서 아주 많은 폼들을 마주친다. 하지만 우리는 의식하지 않기때문에 이러한 사실을 지나치기 쉽다. 폼은 정보를 모을 수 있는 방법이며, 이번 레슨에서 우리는 HTML 의 form 태그에 대해서 배워볼 것이다.

- How a Form Works

컴퓨터는 네트워크인 인터넷으로 정보를 주고받을 수 있다. 정보를 주고 받기 위해서는 HTTP request 을 이용해야한다.

form 태그는 정보를 모으기 위한 좋은 툴이다. 그러한 정보를 어딘가로 보내야한다. 우리는 form 태그에 그러한 정보를 어디로 보내야할 지, 또 어떤 HTTP 요청을 만들지 적어야한다.

```
<form action="/example.html" method="POST">
</form>
```

예시이다.

⇒ action 속성은 어디로 정보들을 보낼 지 지정하는 속성이다.

⇒ method 속성은 HTTP 요청에 포함되는 HTTP 동사(verb) 를 지정한다. ( post 와 같이 대문자로 안써도 되지만 이는 전통과 같기 때문에 쓰는 것이 좋다. )

```
<form action="/example.html" method="POST">
  <h1>Creating a form</h1>
  <p>Looks like you want to learn how to create an HTML form.
  Well, the best way to learn is to play around with it.</p>
</form>
```

form 태그는 아래와 같이 자식태그를 가질 수 있다. 자식태그를 가짐으로써 유저에게 폼에 대한 자세한 설명을 가능토록 한다. 뒤에서 자식태그에 쓸 수 있는 것들을 모두 알아보자.

- Text Input

form 태그 안에 input 필드를 넣고 싶다면 <input> 태그를 사용해야한다. input 요소에는 type 속성을 가지고 있는데 이 속성에 어떤 종류의 인풋을 넣고 싶은지 적어야한다. 우리가 이번에 배울 type 은 type="text" 이다.

또하나 기억해야할 중요한 사실은 input 태그 안에 name 속성도 꼭 포함해야한다는 것이다. name 속성이 없다면, input 의 정보는 form 이 제출될 때 정보가 보내지지 않을 것이다. 제출과 제출버튼에 대해서는 나중에 설명할 것이다.

```
<form action="/example.html" method="POST">
  <input type="text" name="first-text-field">
</form>
```

flow goes into the `value` attribute of the `input` tag.

위 코드는 아래와 같은 텍스트 박스를 만든다.

만약 위와 같이 텍스트를 입력하고 제출하게 되면 input 태그의 value 속성은 타이핑 된 내용이 된다.

```
<form action="/example.html" method="POST">
  <input type="text" name="first-text-field" value="already
pre-filled">
</form>
```

Which renders:

Time to put this knowledge into practice!

value 값을 미리 설정해서 디폴트 값을 설정할 수도 있다.

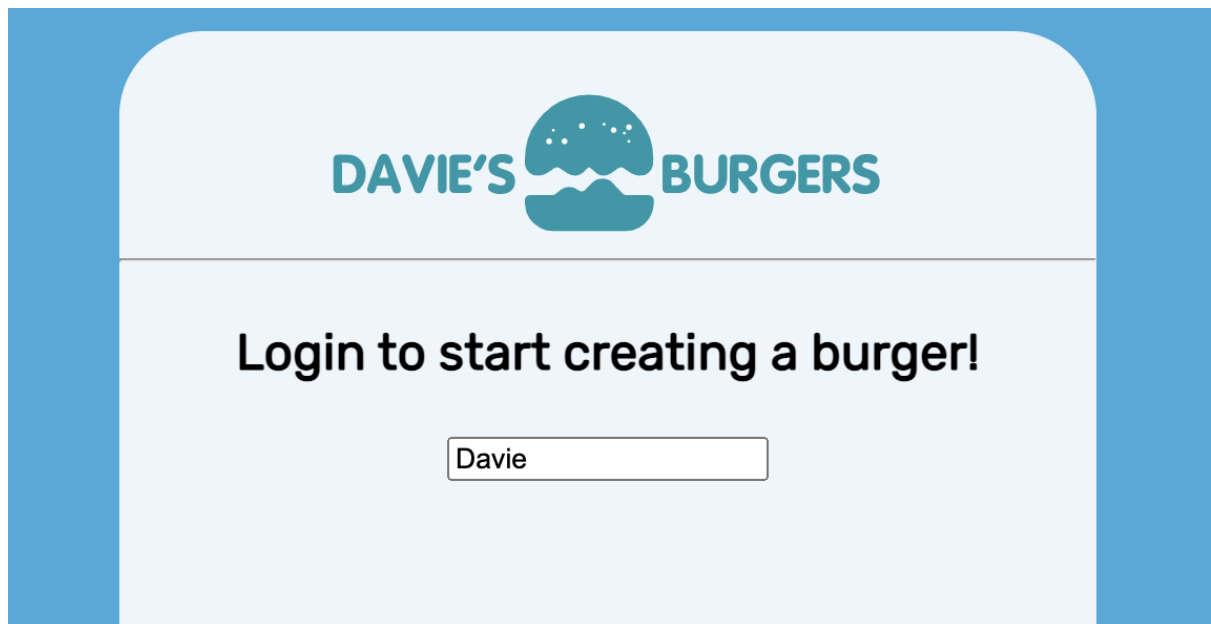
예시 실습

```

<hr>
<form>
  <h1>Login to start creating a burger!</h1>
  <!--Add your code below-->
  <input type="text" name="username" value="Davie">

</form>

```



- Adding a Label

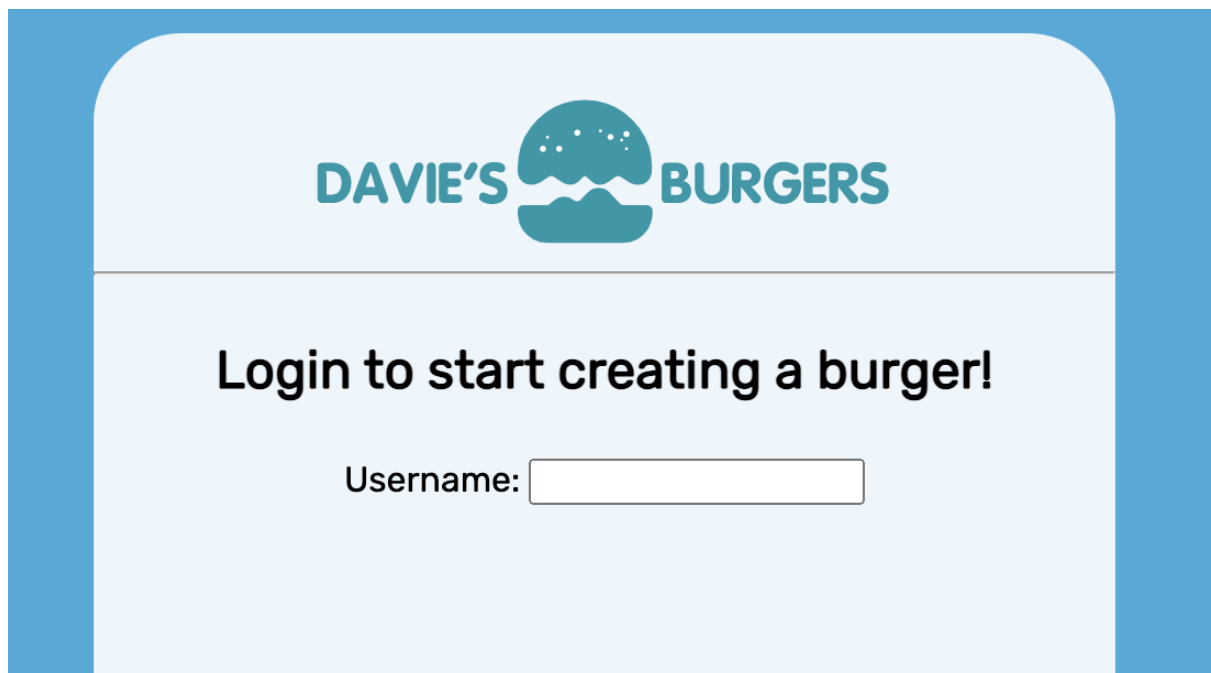
이전에 만든 input 태그에 대한 설명을 붙여보자. <label> 태그를 쓰는 것이다. label 을 쓰기 위해 input 태그에 id 속성을 적어야한다. 그다음 label 태그 내에서 for 속성을 사용하여 input 의 id 값을 적어주면 두 값이 상호성을 갖게 된다.

```
<form action="/example.html" method="POST">
  <label for="meal">What do you want to eat?</label>
  <br>
  <input type="text" name="food" id="meal">
</form>
```

The code above renders:

What do you want to eat?

Look, now users know what the `id` attribute element is for! Another  
위와 같이 label 의 for 속성을 이용하게 되면 라벨 텍스트를 눌러도 input 박스가 활성화된다.

A login form for 'DAVIE'S BURGERS'. The form has a light blue background with rounded corners, set against a darker blue border. At the top, the logo 'DAVIE'S BURGERS' is displayed in teal, with a burger icon between the words. Below the logo, the text 'Login to start creating a burger!' is centered in bold black font. Underneath, the label 'Username:' is followed by a text input field.

- Password Input

`<input type="password">` 를 사용하면 비밀번호 인풋이 렌더링된다.

```
<form>
  <label for="user-password">Password: </label>
  <input type="password" id="user-password" name="user-
password">
</form>
```

After a user types into the field, it would look like:

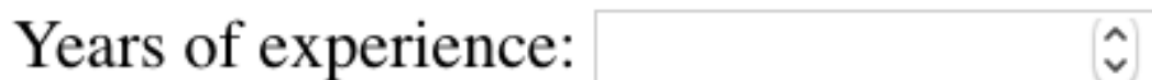


- **Number Input**

숫자만 칠 수 있도록 하는 type 이다. `type="number"`

```
<form>
  <label for="years"> Years of experience: </label>
  <input id="years" name="years" type="number" step="1">
</form>
```

Which renders:



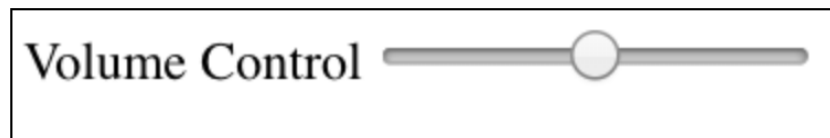
`step` 속성으로 회살표를 누르면 증가되는 숫자를 변경할 수 있다.

- Range Input

다음은 슬라이더를 만들 수 있는 타입이다.

```
<form>
  <label for="volume"> Volume Control</label>
  <input id="volume" name="volume" type="range" min="0"
max="100" step="1">
</form>
```

The code above renders:



최소값과 최대값을 기입해야하고 step 속성도 넣어야한다.

- Checkbox Input

다음은 여러 옵션을 선택할 수 있는 checkbox type이다.



```

<form>
  <p>Choose your pizza toppings:</p>
  <label for="cheese">Extra cheese</label>
  <input id="cheese" name="topping" type="checkbox"
value="cheese">
  <br>
  <label for="pepperoni">Pepperoni</label>
  <input id="pepperoni" name="topping" type="checkbox"
value="pepperoni">
  <br>
  <label for="anchovy">Anchovy</label>
  <input id="anchovy" name="topping" type="checkbox"
value="anchovy">
</form>

```

Which renders:

Choose your pizza toppings:

Extra cheese ☐

Pepperoni ☐

Anchovy ☐

각 input 의 name 은 같게 하고 id 와 value 값은 각 label for 속성과 같게 한다. value 값은 지정해줘야한다!!!

input 을 감싸는 section 태그와 구분 짓는 hr 태그

```

28         <span>Rare</span>
29         <input type="range" name="doneness" id="doneness"
value="3" min="1" max="5">
30         <span>Well-Done</span>
31     </section>
32     <hr>
33
34     <section class="toppings">
35         <span>What toppings would you like?</span>
36         <br>
37         <!--Add your code below for the first checkbox-->
38         <input type="checkbox" name="topping"
value="lettuce" id="lettuce">
39         <label for="lettuce">Lettuce</label>
40         <!--Add your code below for the second checkbox-->
41         <input type="checkbox" name="topping" id="tomato"
value="tomato">
42         <label for="tomato">Tomato</label>
43         <!--Add your code below for the third checkbox-->
44         <input type="checkbox" name="topping" id="onion"
value="onion">
45         <label for="onion">Onion</label>
46
47     </section>
48
49 </form>

```

각 항목을 div 로 묶지 않고 독립성을 유지하는 section 태그로 묶자. section 에 class 를 넣어주고 , 각 항목 사이에는 <hr> 태그를 넣어주자.



## Create a burger!

What type of protein would you like?

How many patties would you like?

How do you want your patty cooked

Rare  Well-Done

What toppings would you like?

☐ Lettuce ☐ Tomato ☐ Onion

- **Radio Button Input**

다음은 하나의 선택지만 선택 가능한 radio type 이다. 체크박스과 같이 라디오 또한 모두 같은 name 속성을 가져야한다.

```
44
45 ▼      <section class="cheesy">
46         <span>Would you like to add cheese?</span>
47         <br>
48         <!--Add your first radio button below-->
49         <input type="radio" name="cheese" id="yes"
value="yes">
50         <label for="yes">Yes</label>
51         <!--Add your second radio button below-->
52         <input type="radio" name="cheese" id="no"
value="no">
53         <label for="no">No</label>
54     </section>
55
```



## Create a burger!

What type of protein would you like?

How many patties would you like?

How do you want your patty cooked

Rare  Well-Done

What toppings would you like?

☐ Lettuce ☐ Tomato ☐ Onion

Would you like to add cheese?

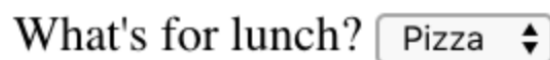
☒ Yes ☐ No

- **Dropdown list**

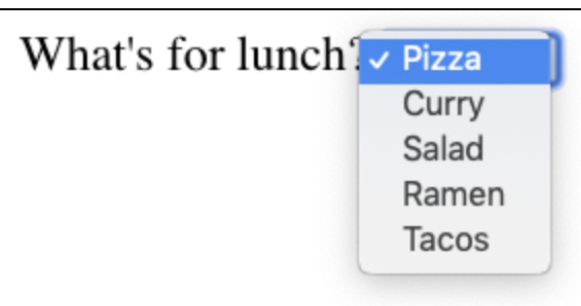
라디오 버튼은 하나의 선택지를 고르는데에 있어서 가장 좋은 방법이다. 하지만 10개의 항목중 하나의 선택지를 고른다면 너무 많은 선택지때문에 눈이 아플 것이며 가독성이 떨어질 것이다. 이럴 때 dropdown list 를 사용하자.

```
<form>
  <label for="lunch">What's for lunch?</label>
  <select id="lunch" name="lunch">
    <option value="pizza">Pizza</option>
    <option value="curry">Curry</option>
    <option value="salad">Salad</option>
    <option value="ramen">Ramen</option>
    <option value="tacos">Tacos</option>
  </select>
</form>
```

Which renders:



And if we click on the field containing the first option, the list is revealed:



다른 종류와 다르게 <select> 태그를 사용해야한다. 또한 다른 옵션들에 대해서는 <option> 태그를 사용해야한다. select 는 오프닝태그만 갖고, option은 오프닝, 엔딩 태그 모두 갖는다.

- Datalist Input ( 데이터리스트 )

dropdown list 와 같은 기능을 하지만, 훨씬 더 간편한 기능을 가지고 있는 타입이다. 이는 검색기능이 추가된 것이며, 해당하는 값이 없다면 직접 입력해서 제출이 가능하다.

이는 input 태그와 같이 쓰여야하는 차이점 또한 가지고 있다. text 타입의 input 태그 내에 list 속성을 추가하여 이 값을 datalist 태그의 id 값과 동일시 하여 연결해준다.

```
<form>
  <label for="city">Ideal city to visit?</label>
  <input type="text" list="cities" id="city" name="city">

  <datalist id="cities">
    <option value="New York City"></option>
    <option value="Tokyo"></option>
    <option value="Barcelona"></option>
    <option value="Mexico City"></option>
    <option value="Melbourne"></option>
    <option value="Other"></option>
  </datalist>
</form>
```

Notice, in the code above, we have an `<input>` that has a `list` attribute. The `<input>` is associated to the `<datalist>` via the `<input>` 's `list` attribute and the `id` of the `<datalist>`.

From the code provided, the following form is rendered:

Ideal city to visit?

예시에서 볼 수 있는 바와 같이, option 사이에 text 칸에 아무것도 안 써도 되며, value 값이 대체한다.

Ideal city to visit?

- New York City
- Tokyo
- Barcelona
- Mexico City
- Melbourne
- Other

실습 예시

```
62
63 ▼      <section class="sauce-selection">
64          <label for="sauce">What type of sauce would you
like?</label>
65          <input list="sauces" id="sauce" name="sauce">
66          <!--Add your code below-->
67 ▼      <datalist id="sauces">
68          <option value="ketchup"></option>
69          <option value="mayo"></option>
70          <option value="mustard"></option>
71
72      </section>
73
74  </form>
```



☐ Yes    ☐ No

---

What type of bun would you like? Sesame ▼

---

What type of sauce would you like? m| ▼

- mayo
- mustard

- **Textarea element**

많은 글을 써야할 때는 textarea 태그를 사용한다. 행과 열의 수를 제한할 수 있다. 엔딩태그를 써줘야한다.

```
<form>
  <label for="blog">New Blog Post: </label>
  <br>
  <textarea id="blog" name="blog" rows="5" cols="30">
  </textarea>
</form>
```

In the code above, an empty `<textarea>` that is 5 rows by 30 columns is rendered to the page like so:



디폴트 값을 아래와 같이 쓸 수 있다.

```
<textarea>Adding default text</textarea>
```

실습예시

```

<section class="extra-info">
  <label for="extra">Anything else you want to add?</label>
  <br>
  <!--Add your code below-->
  <textarea id="extra" name="extra" rows="3" cols="40">
    There are not Comments.
  </textarea>
</section>

```

Anything else you want to add?

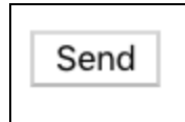
There are not Comments.

- **Submit Form**

우리는 폼을 이용해서 정보를 모으는 것이 목적이었음을 기억해라. 이러한 폼이 모두 입력된 뒤 서버에 제출을 해야 데이터를 모을 수 있게 된다. input 태그의 submit type 을 이용하여 제출을 해보자. value 값이 제출 버튼 안에 표시되는 텍스트 내용이다.

```
<form>
  <input type="submit" value="Send">
</form>
```

Which renders:



실습 예제

```
    </section>
    <hr>
    ▼    <section class="submission">
          <!--Add your code below-->
          <input type="submit" value="제출하기">
        </section>

    </form>
  </section>
</body>
</html>
```



## Create a burger!

What type of protein would you like?

How many patties would you like?

How do you want your patty cooked

Rare  Well-Done

What toppings would you like?

☐ Lettuce ☐ Tomato ☐ Onion

Would you like to add cheese?

☐ Yes ☐ No

What type of bun would you like?

What type of sauce would you like?

Anything else you want to add?

제출하기

- Review

⇒ form 태그에 action 과 method 속성을 넣어줘야한다.

⇒ input type 에 속한 값들: text, password, number, range, checkbox, radio, list(datalist 와 쓰일 때) , submit

⇒ select 태그는 droplist 에서 쓰이며, option 태그를 자식태그로 갖는다.

⇒ datalist 태그는 droplist 와 같이 option 태그를 자식태그로 가지며, input 태그와 같이 쓰여야한다.

⇒ textarea 태그

⇒ 데이터가 제출될 때, name=value 의 쌍으로 제출이 되므로 name 속성은 무조건 들어야한다!!

⇒ placeholder 을 추가할 수 있다.

---


## Lesson 3. Form Validation

### 목차

- Intro to HTML Form Validation
- Requiring an Input
- Set a Minimum and Maximum
- Checking Text Length
- Matching a Pattern
- Review

- Intro to HTML Form Validation

Please enter your first name: \*

Validation 은 요청된 데이터에 대해 제공된 유저 데이터를 체크하는 것이다.

검증에는 두가지 방식이 있는데 하나는 서버, 나머지는 클라이언트에서의 검증이다. 대표적인 예로 로그인 페이지를 들 수 있다. 서버에서의 검증은 id 와 password 폼에 입력된 정보를 서버로 보내 서버의 내용과 일치하는지 검증하는 것이며, 우리가 사용할 클라이언트 검증은 데이터가 서버에 보내지기 전에 미리 검증하는 것이다.

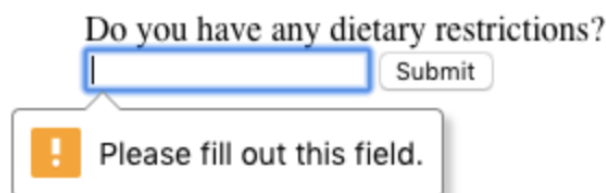
HTML의 빌트인 클라이언트 사이드 검증의 사용하는 이점은 다른 브라우저들 사이에서 함께 공유되어지고 있다. 이것을 시간을 아끼고 불편함을 해소할 수 있으며, 우리의 서버가 해커로부터 어느정도 보호가 가능하다.

- **Requiring an Input**

무조건 입력해야하는 input에 required 키워드를 입력하자.

```
<form action="/example.html" method="POST">
  <label for="allergies">Do you have any dietary
restrictions?</label>
  <br>
  <input id="allergies" name="allergies" type="text"
required>
  <br>
  <input type="submit" value="Submit">
</form>
```

This renders a text box, and if we try to submit the `<form>` without filling it out we get this message:



Do you have any dietary restrictions?

! Please fill out this field.



- Set a Minimum and Maximum

Input type="number" 와 Input type="range" 에서 min , max 값을 설정할 수 있다.

```
<form action="/example.html" method="POST">
  <label for="guests">Enter # of guests:</label>
  <input id="guests" name="guests" type="number" min="1"
max="4">
  <input type="submit" value="Submit">
</form>
```

If a user tries to submit an input that is less than 1 a warning will appear:

- Checking Text Length

텍스트에도 길이 제한을 위한 min max 값을 설정할 수 있다. input 태그 내에 minlength, maxlength 속성을 입력하자.

```
<form action="/example.html" method="POST">
  <label for="summary">Summarize your feelings in less than 250
characters</label>
  <input id="summary" name="summary" type="text" minlength="5"
maxlength="250" required>
  <input type="submit" value="Submit">
</form>
```

If a user tries to submit the `<form>` with less than the set minimum, this message appears:

Send us your thoughts and suggestions!

No

! Please lengthen this text to 5 characters or more (you are currently using 2 characters).

아이디 길이나 비밀번호 길이 같은 것도 제한할 수 있다.

## 실습예제

```

10 ▼ <section class="overlay">
11     <h1>Sign Up</h1>
12     <p>Create an account:</p>
13 ▼   <form action="submission.html" method="GET">
14       <!--Add the minlength and maxlength attributes to the input
15       fields-->
16       <label for="username">Username:</label>
17       <br>
18       <input id="username" name="username" type="text" minlength="3"
19       maxlength="15" required>
20       <br>
21       <label for="pw">Password:</label>
22       <br>
23       <input id="pw" name="pw" type="password" minlength="8"
24       maxlength="15" required>
25       <br>
26       <input type="submit" value="Submit">
27   </form>
28 </section>

```

**Sign Up**

Create an account:

Username:

Password:

이 텍스트를 8자 이상으로 늘리세요(현재 4자 사용 중).

Sign Up

Create an account:

Username:

123456789012345

Password:

.....

Submit

이렇게 max 를 넘어가면 더이상 입력이 안된다!!

- Matching a Pattern

텍스트의 길이를 제한하는 것 뿐만 아니라 input 태그 내에 pattern 속성을 사용하여 정규표현식으로 텍스트에 대한 조건을 걸 수 있다.

```
<form action="/example.html" method="POST">
  <label for="payment">Credit Card Number (no spaces):
</label>
  <br>
  <input id="payment" name="payment" type="text" required
pattern="[0-9]{14,16}">
  <input type="submit" value="Submit">
</form>
```

With the `pattern` in place, users can't submit the `<form>` with a number that doesn't follow the regex. When they try, they'll see a validation message like so:

Credit Card Number (no spaces):

123456789|

! Please match the requested format.

다음은 0~9까지의 숫자를 적어도 14자리부터 16자리의 숫자를 입력하라는 정규표현식이다.

```
<!--Add the pattern attribute to the input below-->
<input id="username" name="username" type="text"
required minlength="3" maxlength="15" pattern="[a-zA-Z0-9]+">
```

id 폼에 특수문자가 들어가지 못하게 하는 정규표현식이다.

The image shows a 'Sign Up' form with a light green background and rounded corners. The title 'Sign Up' is in large blue font. Below it, the text 'Create an account:' is in smaller blue font. The 'Username:' label is in blue. The input field contains '@asdfasdfs123'. A red error message box with a white exclamation mark icon is positioned below the input field, containing the text '요청한 형식과 일치시키세요.' (Please match the requested format). A 'Submit' button is located below the error message.

# Sign Up

Create an account:

Username:

! 요청한 형식과 일치시키세요.

Submit

특수문자를 넣으면 위와 같은 경고 메시지가 뜬다.

## 오늘의 단어

- obscure : 보이지않게 하다. ; 불명확한, 애매한
- On the flip side : 다른 한편으로는