

Student: Jorge Alvarez Grana (#249179033)
Class: WDD330
Term: Fall 2021
Subject: W10 Readings

MDN: Validating Forms

This section was a rather interesting one because I got to see how we can help users fill forms correctly. As a user myself, I don't like to fill forms and I usually try to do as little as I can in order to move on. But those forms might need to be filled correctly because they include user's information that might be relevant in other steps. Forms are one of the main ways for companies to get information and for users to interact with those businesses. If any attribute is wrong (card name, email, card number, phone...) the operation might well be unable to move forward.

It was interesting to see the two alternatives for doing this type of job: built-in validation rules and JavaScript. On the second I had done some previously work on an example shown on week 4. But this week I tried to use some of the built-in options using HTML & CSS. On this case I had a form where the field was in red using CSS until I typed the write data, turning then into green. And the type of data was inserted into the HTML:

```
<label class="top">First Name* <input type="text" name="fname" pattern="[A-Za-z].{0,}"  
required></label>
```

User Information:

First Name*

Last Name*

E-mail*

```
form label.top input:required {  
  border-left: 2px solid red;  
  border-right: 2px solid red;  
}  
  
form label.top input:required:valid {  
  border-left: 2px solid green;  
  border-right: 2px solid green;  
}
```

MDN: Using Fetch

Here it was explained that these option was used before using XMLHttpRequest, although the text didn't explain much why now *fetch()* is a better option. I read about it in an article¹ and it was said that provides more consistency fetching data, there are no callbacks and foremost is that it is well supported by modern browsers. However, as most of the cases thus far seen, between the two the choice is ours, because although "Fetch is the future [...] the API is relatively new, it does not provide all XHR functionality, and some options are cumbersome."

1 <https://www.sitepoint.com/xmlhttprequest-vs-the-fetch-api-whats-best-for-ajax-in-2019/>

On my own I tried to make a service where I could retrieve a Pokemon from the site <https://pokeapi.co> . In the end a function like this would add a Pikachu icon to the website:

```
function bringPokemon (){  
  fetch("https://pokeapi.co/api/v2/pokemon/pikachu/").  
  then((res) => res.json()).then((data) => {  
    createPokemon(data);  
  });  
}
```

Search Pikachu!



pikachu