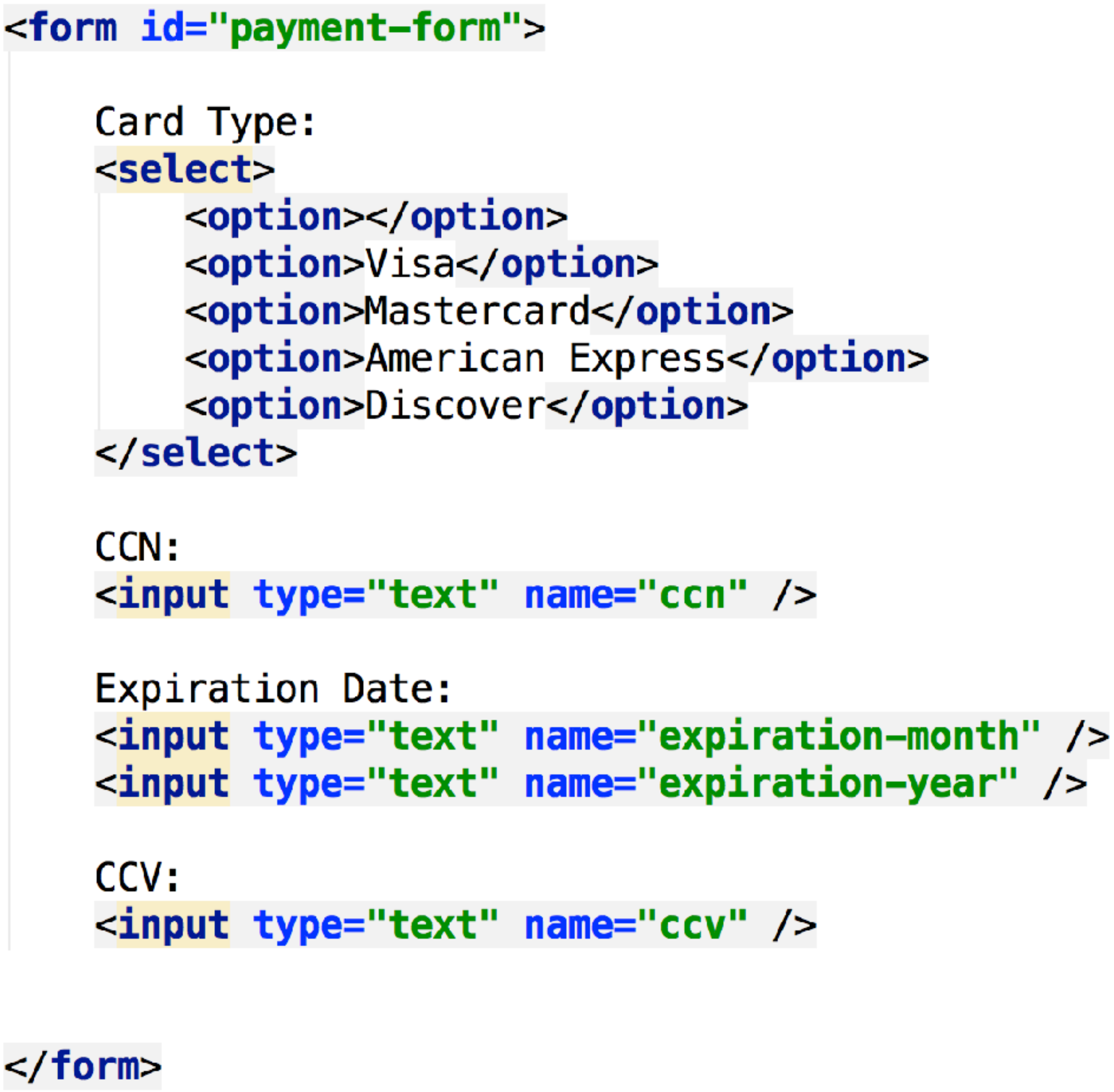
HIGHER ORDER COMPONENTS

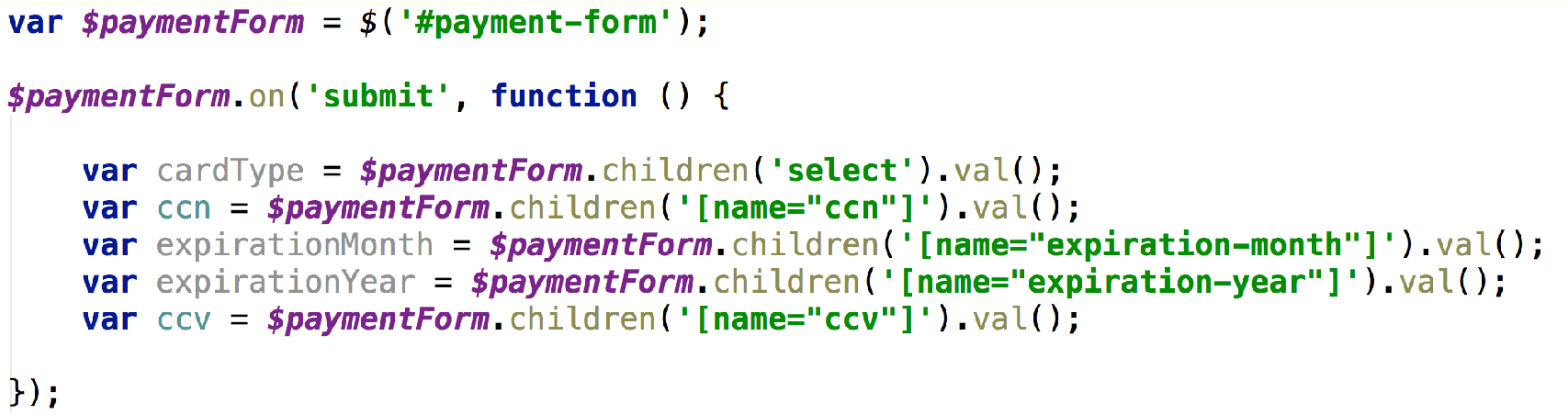
Do what you gotta do

“I’m so happy I get to program form validation for our new product! This is going to be lots of fun!”

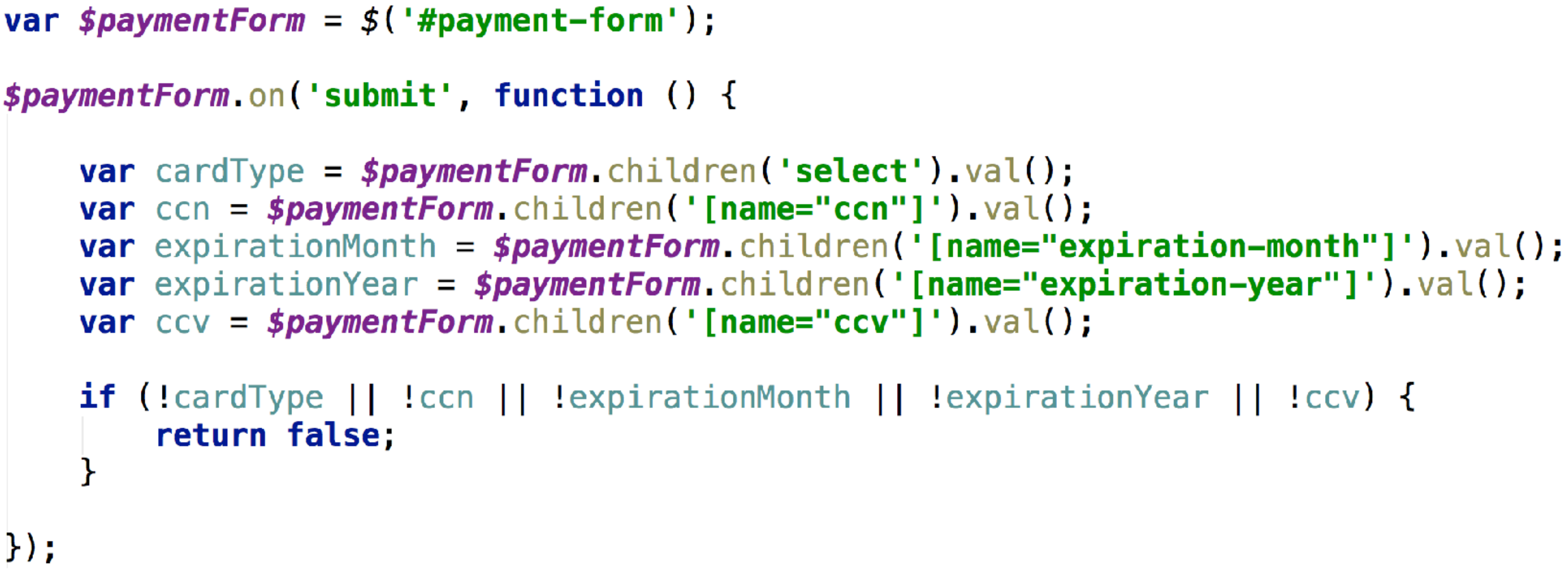
– NO DEVELOPER EVER



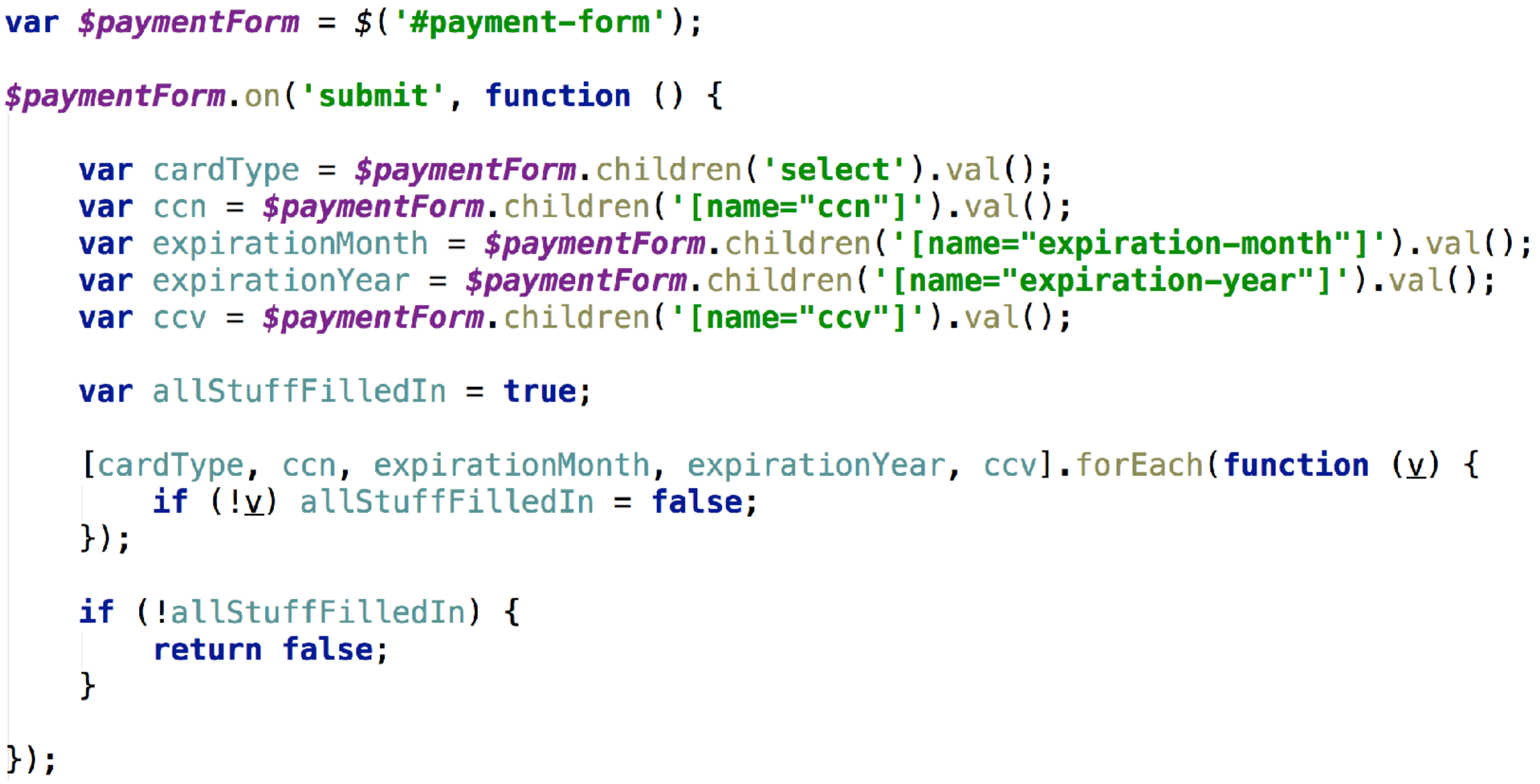
```
<form id="payment-form">

    Card Type:
    <select>
        <option></option>
        <option>Visa</option>
        <option>Mastercard</option>
        <option>American Express</option>
        <option>Discover</option>
    </select>

    CCN:
    <input type="text" name="ccn" />

    Expiration Date:
    <input type="text" name="expiration-month" />
    <input type="text" name="expiration-year" />

    CCV:
    <input type="text" name="ccv" />


</form>
```



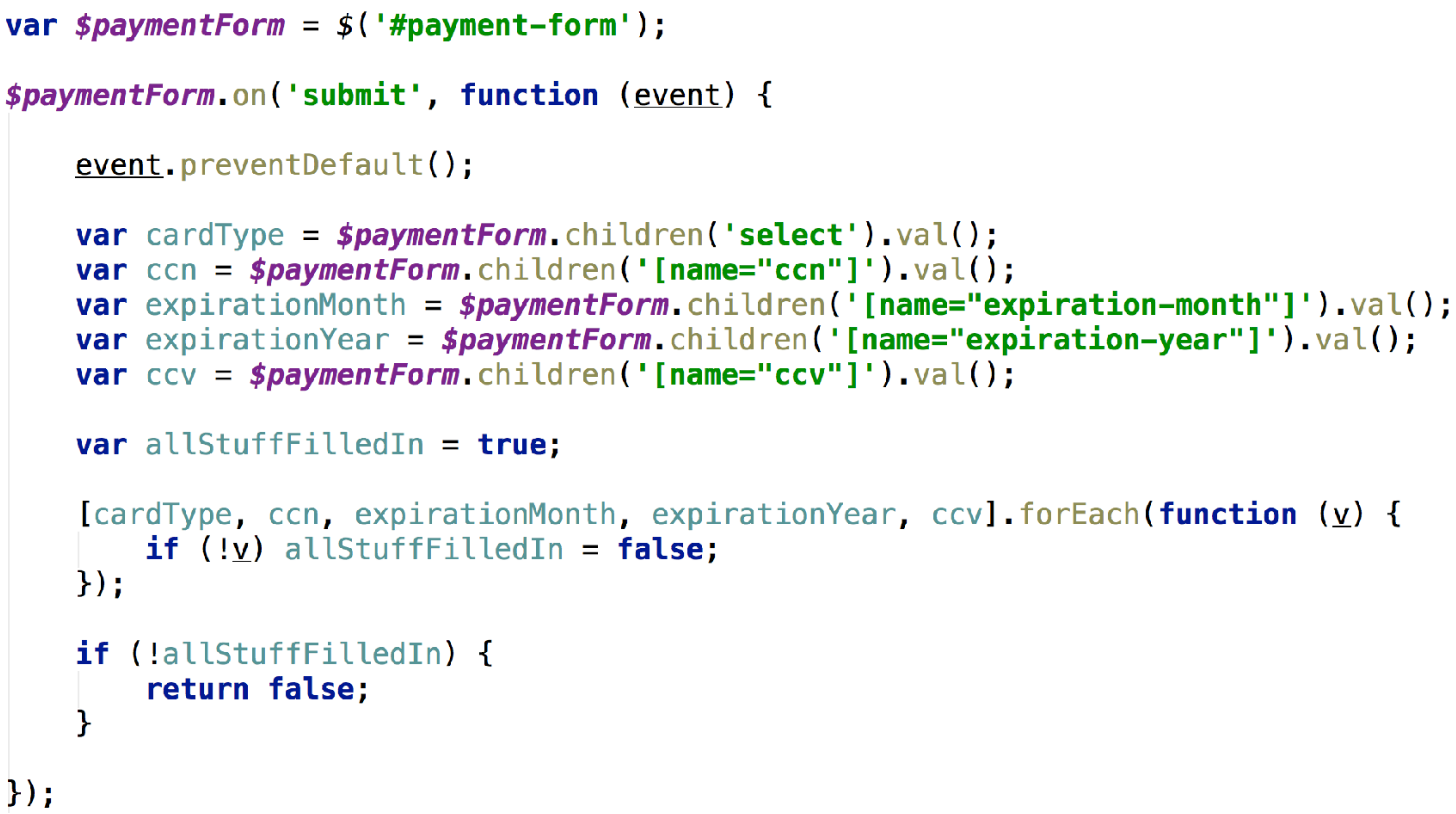
```
var $paymentForm = $('#payment-form');

$paymentForm.on('submit', function () {

    var cardType = $paymentForm.children('select').val();
    var ccn = $paymentForm.children('[name="ccn"]').val();
    var expirationMonth = $paymentForm.children('[name="expiration-month"]').val();
    var expirationYear = $paymentForm.children('[name="expiration-year"]').val();
    var ccv = $paymentForm.children('[name="ccv"]').val();

});
```



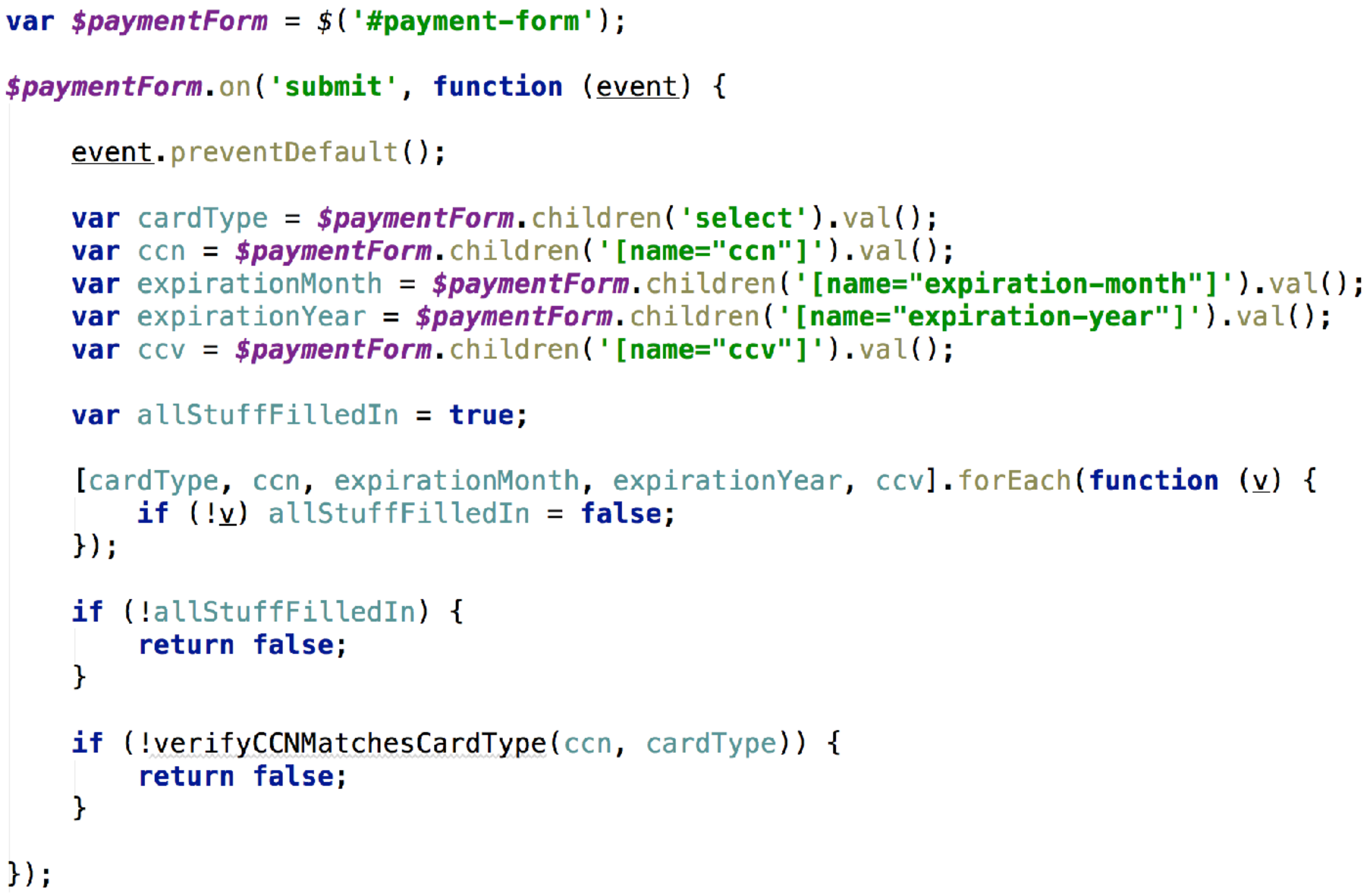
```
var $paymentForm = $('#payment-form');

$paymentForm.on('submit', function () {

    var cardType = $paymentForm.children('select').val();
    var ccn = $paymentForm.children('[name="ccn"]').val();
    var expirationMonth = $paymentForm.children('[name="expiration-month"]').val();
    var expirationYear = $paymentForm.children('[name="expiration-year"]').val();
    var ccv = $paymentForm.children('[name="ccv"]').val();

    if (!cardType || !ccn || !expirationMonth || !expirationYear || !ccv) {
        return false;
    }

});
```



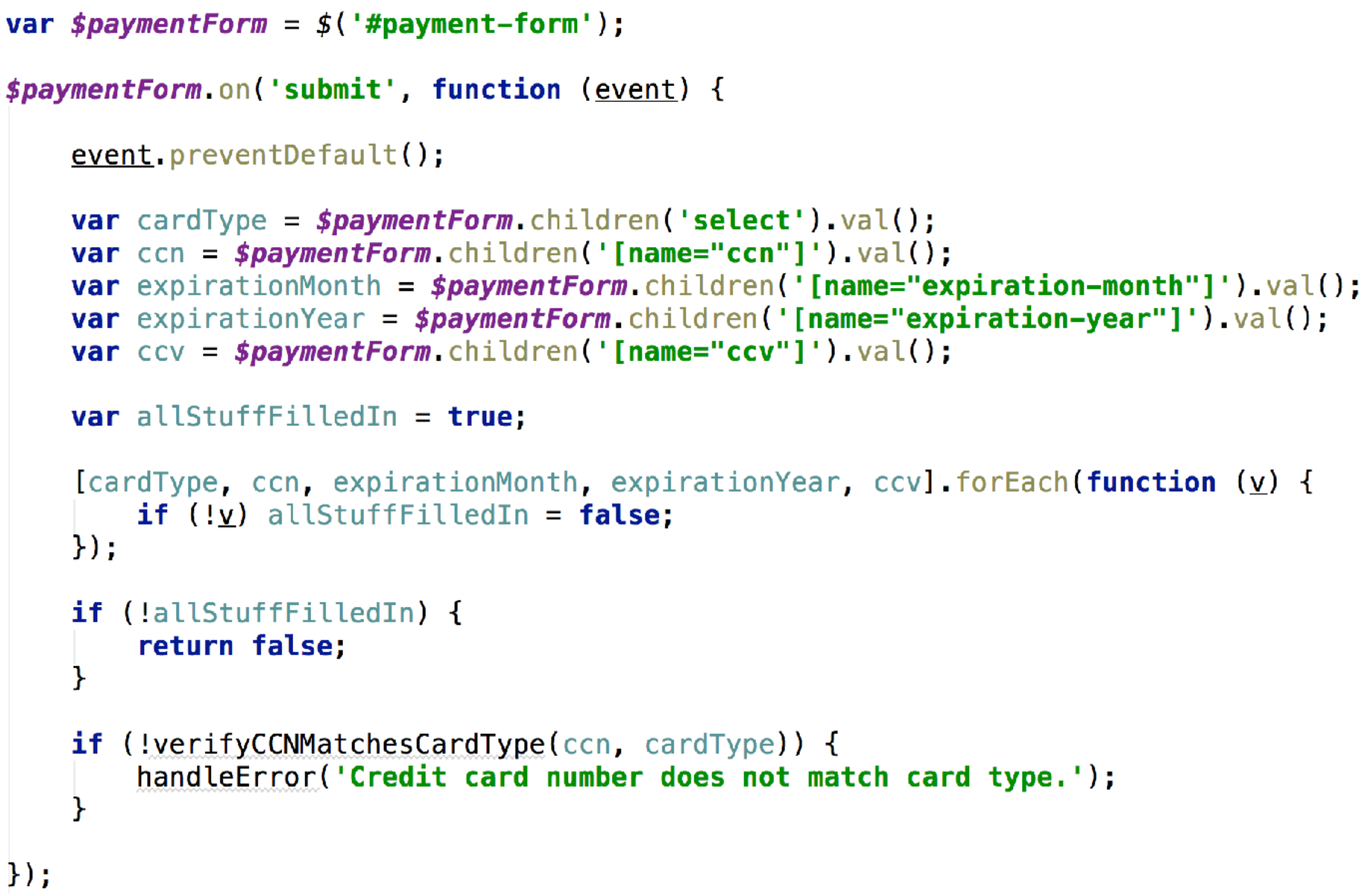
```
var $paymentForm = $('#payment-form');

$paymentForm.on('submit', function () {

    var cardType = $paymentForm.children('select').val();
    var ccn = $paymentForm.children('[name="ccn"]').val();
    var expirationMonth = $paymentForm.children('[name="expiration-month"]').val();
    var expirationYear = $paymentForm.children('[name="expiration-year"]').val();
    var ccv = $paymentForm.children('[name="ccv"]').val();

    var allStuffFilledIn = true;

    [cardType, ccn, expirationMonth, expirationYear, ccv].forEach(function (v) {
        if (!v) allStuffFilledIn = false;
    });

    if (!allStuffFilledIn) {
        return false;
    }

});
```



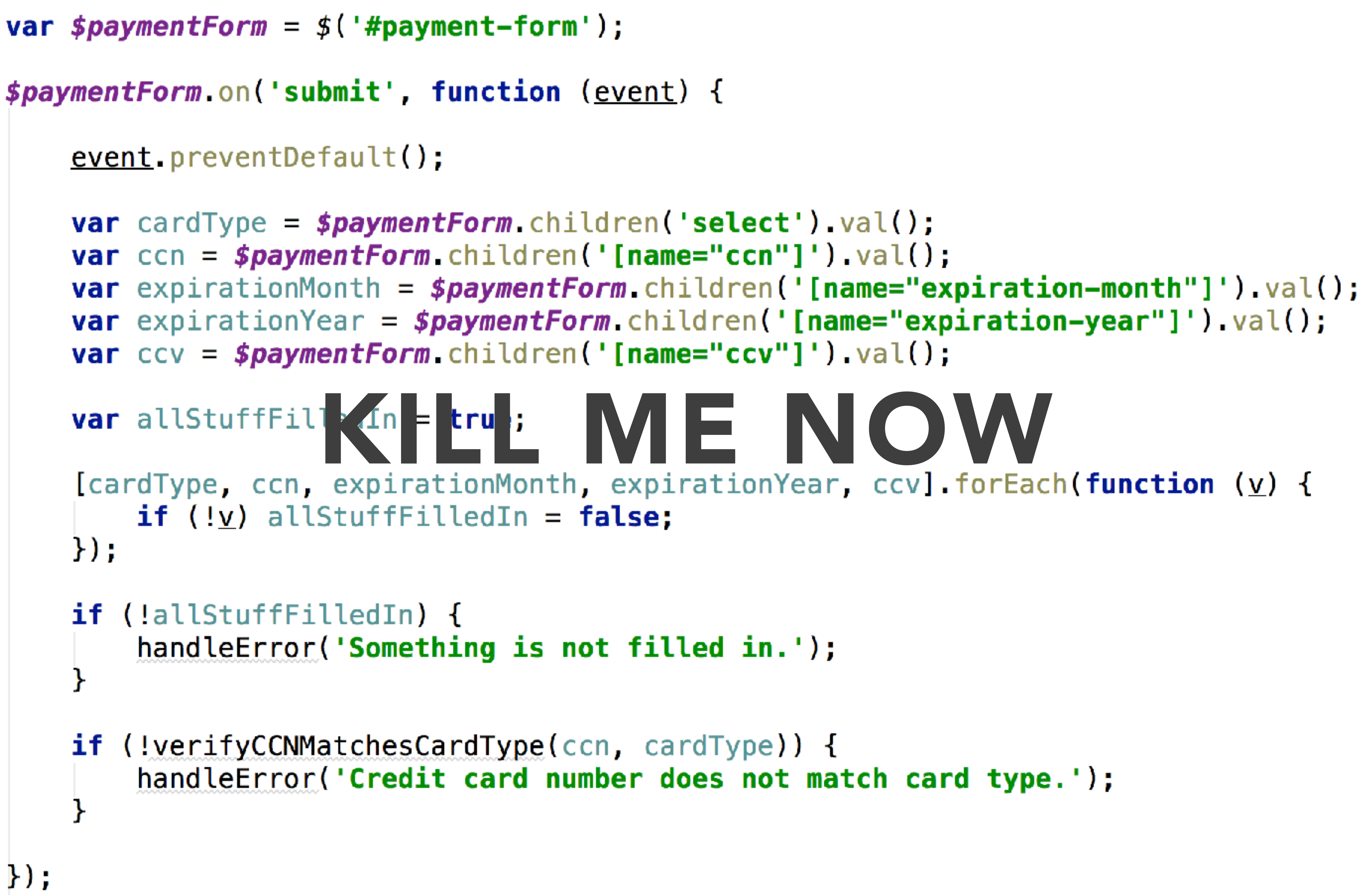
```
var $paymentForm = $('#payment-form');

$paymentForm.on('submit', function (event) {

    event.preventDefault();

    var cardType = $paymentForm.children('select').val();
    var ccn = $paymentForm.children('[name="ccn"]').val();
    var expirationMonth = $paymentForm.children('[name="expiration-month"]').val();
    var expirationYear = $paymentForm.children('[name="expiration-year"]').val();
    var ccv = $paymentForm.children('[name="ccv"]').val();

    var allStuffFilledIn = true;

    [cardType, ccn, expirationMonth, expirationYear, ccv].forEach(function (v) {
        if (!v) allStuffFilledIn = false;
    });

    if (!allStuffFilledIn) {
        return false;
    }

});
```



```
var $paymentForm = $('#payment-form');

$paymentForm.on('submit', function (event) {

    event.preventDefault();

    var cardType = $paymentForm.children('select').val();
    var ccn = $paymentForm.children('[name="ccn"]').val();
    var expirationMonth = $paymentForm.children('[name="expiration-month"]').val();
    var expirationYear = $paymentForm.children('[name="expiration-year"]').val();
    var ccv = $paymentForm.children('[name="ccv"]').val();

    var allStuffFilledIn = true;

    [cardType, ccn, expirationMonth, expirationYear, ccv].forEach(function (v) {
        if (!v) allStuffFilledIn = false;
    });

    if (!allStuffFilledIn) {
        return false;
    }

    if (!verifyCCNMatchesCardType(ccn, cardType)) {
        return false;
    }

});
```



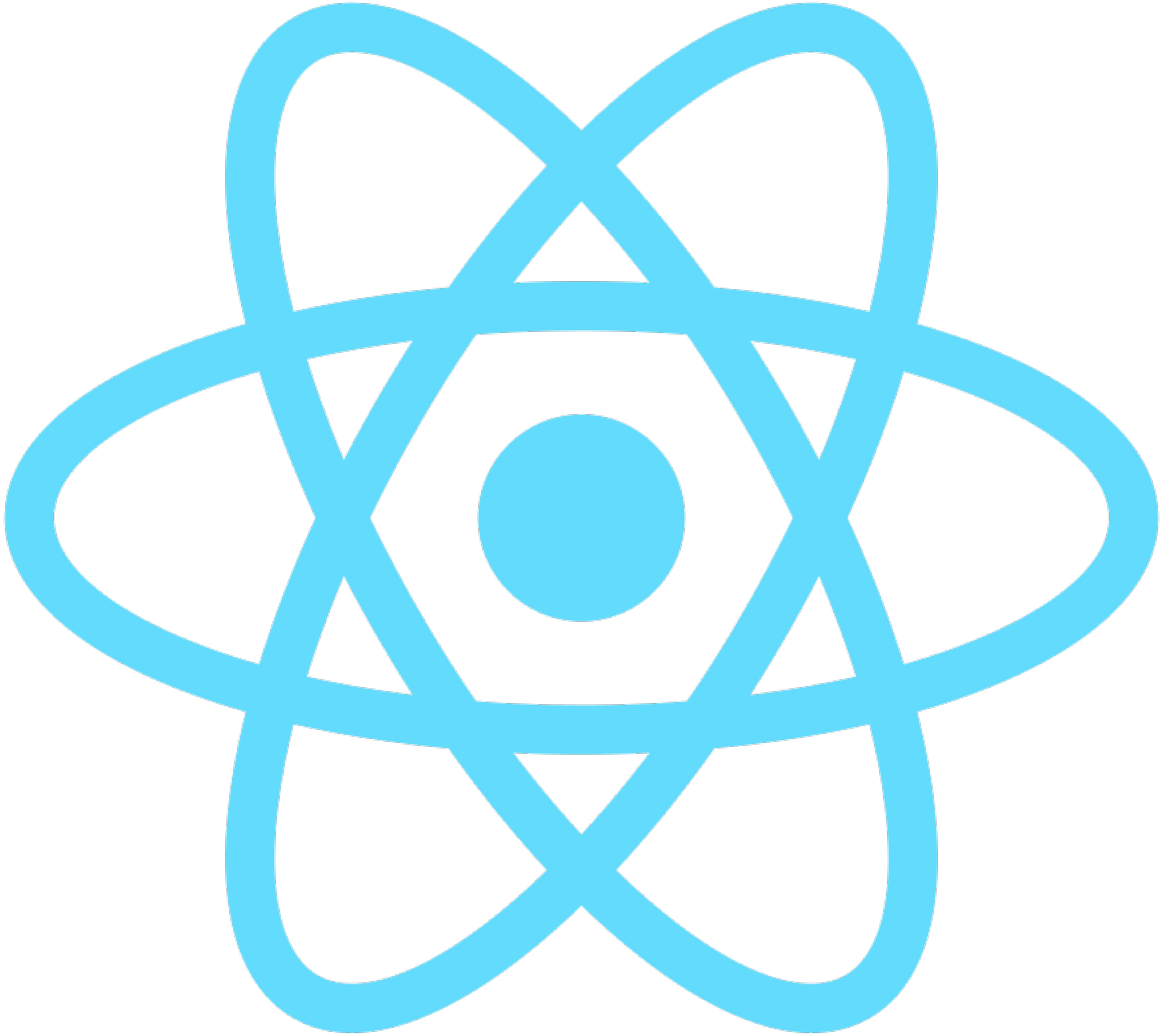
```javascript
var $paymentForm = $('#payment-form');

$paymentForm.on('submit', function (event) {

    event.preventDefault();

    var cardType = $paymentForm.children('select').val();
    var ccn = $paymentForm.children('[name="ccn"]').val();
    var expirationMonth = $paymentForm.children('[name="expiration-month"]').val();
    var expirationYear = $paymentForm.children('[name="expiration-year"]').val();
    var ccv = $paymentForm.children('[name="ccv"]').val();

    var allStuffFilledIn = true;

    [cardType, ccn, expirationMonth, expirationYear, ccv].forEach(function (v) {
        if (!v) allStuffFilledIn = false;
    });

    if (!allStuffFilledIn) {
        return false;
    }

    if (!verifyCCNMatchesCardType(ccn, cardType)) {
        handleError('Credit card number does not match card type.');
    }

});
```



```
var $paymentForm = $('#payment-form');

$paymentForm.on('submit', function (event) {

    event.preventDefault();

    var cardType = $paymentForm.children('select').val();
    var ccn = $paymentForm.children('[name="ccn"]').val();
    var expirationMonth = $paymentForm.children('[name="expiration-month"]').val();
    var expirationYear = $paymentForm.children('[name="expiration-year"]').val();
    var ccv = $paymentForm.children('[name="ccv"]').val();

    var allStuffFilledIn = true;

    [cardType, ccn, expirationMonth, expirationYear, ccv].forEach(function (v) {
        if (!v) allStuffFilledIn = false;
    });

    if (!allStuffFilledIn) {
        handleError('Something is not filled in.');
    }

    if (!verifyCCNMatchesCardType(ccn, cardType)) {
        handleError('Credit card number does not match card type.');
    }

});
```

KILL ME NOW

INPUT VALUING AND FORM VALIDATION IS WROUGHT WITH BUSINESS LOGIC AND EXTREMELY VERBOSE MUCH LIKE THIS SLIDE HAHA GET IT THERE ARE MANY WORDS HERE



FORMS IN REACT

- The React-Redux ecosystem has plenty of tools to help you deal with forms
- Today, we’ll do it ourselves by writing higher-order components - that way, we’ll know how any library we use down the line works

HIGHER ORDER COMPONENTS

- Remember: classes === functions
- Higher order function: a function that returns another function
- Decorator function: a function that takes a function as an argument, and returns a new function that “decorates” the behavior of the passed in function
- Higher order component: same idea - a function that takes a component as an argument, and returns a React class that renders the inner function with new props (possibly from state that the HOC sets)

HOW TO GET DATA

- Refs: an “escape hatch” to the underlying DOM
- Change events - creates a React Synthetic Event to get the value of the form as it comes in
- React Synthetic Event - wraps the native event
    - “Pooled” - cannot be used asynchronously

WHERE DOES FORM DATA BELONG?

DOES IT HAVE TO GO ON MY REDUX STORE STATE? :(

YES OR NO

- It’s up to you!
- You may choose distinguish between application state and local UI state
  - You could write a regular react component that uses setState instead of the Redux store
- You can also put it on the store so that you can use Redux’s dev tools with it