

# FORMS

---

*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;
    }

});

```



```

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.');
```

# KILL ME NOW

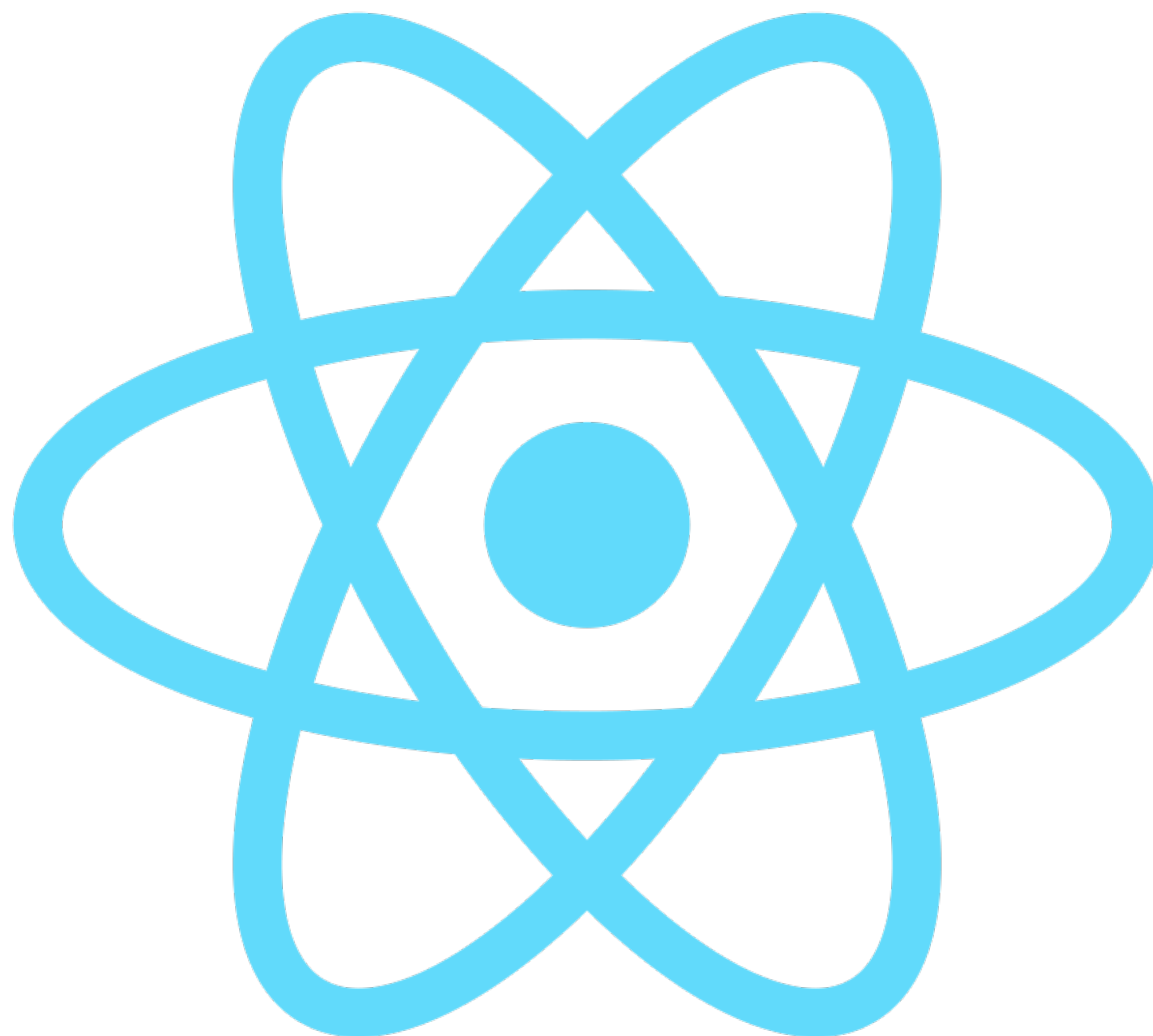
```

    }

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

**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

- React is declarative rather than imperative, which makes dealing with stateful form data easier.
  - `render (stateOfFormData) => whatTheFormShouldLookLike`
- The biggest challenge is structuring where that state lives

# HOW TO GET DATA



# HOW TO GET DATA

- Right now, we're used to code where we get form data off of the `event` object
- Change events in React create events too - with one gotcha
- React Synthetic Event - wraps the native event
  - "Pooled" - cannot be used asynchronously

# CONTROLLED/UNCONTROLLED

- *Controlled* form components have their values directly controlled by state
  - Changes to the value on state update what you see in the form
  - Changes to what you see in the form update the value on state
- *Uncontrolled* form components do not
  - Changes to what you see in the form update the value on state
  - ...But NOT the other way around

# CONTROLLED/UNCONTROLLED

- **Controlled components**
  - MUCH easier to programmatically handle your form's data
  - A bit tedious to set up initially
- **Uncontrolled components**
  - No need to use `this.state` to manage form data
  - Harder to deal with



# WHERE DOES FORM STATE BELONG?

# APP STATE V LOCAL STATE

- Many components in your app might need access to who the user is - this is known as application state
  - Should be managed high up on your state tree
- Only a form needs to know about the data inside it
  - Can be managed lower down

```
    <AppContainer> // this.state = { user: {}, favoriteDog: '' }  
    /      \  
<Kittens> //stateless  <Puppies> // stateless  
                        \  
                        <FavPuppyFormContainer> // this.state = { inputValue: '' }  
                        \  
                        <FavPuppyForm> // has an <input>
```



