

Web Clients

Sabina Borbély 27.04.2020

Einbindung von SVG Signature Sketch in React

2 alternative Implementierungen:

- React Component als Wrapper um ein Custom Element
 - React Component als Wrapper um SVG
-

SVG Signature Sketch in React

wiederverwendbare Komponenten

Bei SVG Signature bieten sich die folgenden Operationen an:

- Änderung von **stroke** und **stroke-width** zur Beeinflussung von Linien
- Änderung von **fill** zur Veränderung der Füllfarbe
- Änderung von **opacity** zur Veränderung der Durchsichtigkeit eines Elements
- Veränderung von **Position, Größe und Rotation mittels transform**

Custom Element Attributes / React Component props:

- stroke fill
- stroke-width
- opacity
- background
- width, height

WEB COMPONENTS

- Shadow Dom
- HTML Templates
- Observed Attributes
- Lifecycle methods
- Custom elements
- Register the custom tag

REACT

- Functional Component
- Class Component
- Props - PropTypes
- Lifecycle methods
- Export / Import module

React Components

1

Main Component

ተሰጥቶ ለገጽ ልማት የሚያገለግል የሚሆን አጠቃላይ ንድፍ ነው።
ሌሎች ምሳሌዎች ለገጽ ልማት የሚያገለግሉ ምሳሌዎችን ያካትታል።

2



Parent Component

ተሰጥቶ ለገጽ ልማት የሚያገለግል የሚሆን አጠቃላይ ንድፍ ነው።
ሌሎች ምሳሌዎች ለገጽ ልማት የሚያገለግሉ ምሳሌዎችን ያካትታል።

3

Child Component



Parent Component

ተሰጥቶ ለገጽ ልማት የሚያገለግል የሚሆን አጠቃላይ ንድፍ ነው።
ሌሎች ምሳሌዎች ለገጽ ልማት የሚያገለግሉ ምሳሌዎችን ያካትታል።

01

Version

Web Components

```
export default class Signature extends HTMLElement {
  // Specify observed attributes so that
  // attributeChangedCallback will work
  static get observedAttributes() {
    return ['strokewidth', 'fillcolor', 'bkgcolor', 'strokeopacity', 'signaturewidth', 'signatureheight'];
  }

  constructor() {
    super();
    // lets create our shadow root
    this.shadowObj = this.attachShadow({ mode: 'open' });

    //gets invoked every time a copy of a Web Component get
    // added to the DOM
    connectedCallback() {
      this.init();
      this.shadowObj.querySelector('#signature')
        .addEventListener('click', (e) => this.handleClick(
    ));
    }

    .....
  }
}
```

01

Version

Using Web Components in React

```
componentDidMount() {  
  if (!customElements.get('signature-element')) {  
    customElements.define('signature-  
element', Signature);  
  }  
}
```

```
return (  
  <div style={borderStyle}>  
    < signature-element  
      strokeWidth={strokeWidth} fillColor={fillColor}  
      bgColor={bgColor} strokeopacity={strokeopacity}  
      signaturewidth={signaturewidth}  
      signatureheight={signatureheight}>  
      </signature-element>  
    </div>  
  );
```

```
export default class SignatureR2 extends Component {
```

```
.....
```

```
  constructor(props) {  
    super(props);
```

```
  
    this.pathInfo = [  
      { path: '' },  
      { path: '' },  
      { path: '' }];
```

```
    this.path0Ref = React.createRef();  
    this.path1Ref = React.createRef();  
    this.path2Ref = React.createRef();  
    this.signatureRef = React.createRef();
```

```
  
    /* our JavaScript and CSS are tied slightly more together now,  
    w, as they both need to be aware of what CSS custom variables we  
    use. This is certainly a trade-  
    off but it's one we are willing to make. */
```

```
    this.theme = {  
      "--stroke-opacity": props.strokeopacity,  
      "--stroke-color": props.fillcolor,  
      "--stroke-width": props.strokewidth,  
      "--bkg-color": props.bkgcolor,  
      "--signature-width": props.signaturewidth,  
      "--signature-height": props.signatureheight  
    };  
  }  
}
```

02

Version

React Class Component - SVG SignatureR2.js


```
componentDidMount() {  
  this.pathInfo = [  
    { path: this.path0Ref.current },  
    { path: this.path1Ref.current },  
    { path: this.path2Ref.current },  
  ]  
  
  this.pathInfo.forEach(it => {  
    it.total = it.path.getTotalLength();  
    it.path.style.strokeDasharray = it.total; // dash with full length, followed by equal empty space  
    it.path.style.strokeDashoffset = it.total; // show the empty space  
  
  });  
  
  this.updateCSSCustomProperties(this.props);  
  
  this.signatureRef.current.addEventListener('click', (e) => this.handleClick());  
}
```

02

Version

React Class Component - SVG SignatureR2.js

```

render() {
  return (
    <div className="SignatureR2" style={this.theme} >
      <div id="signature" ref={this.signatureRef} >
        <svg viewBox="0 -
2 43 62" fill="none" xmlns="http://www.w3.org/2000/svg">
          <path id="path0" ref={this.path0Ref} d="M18 12V45" stroke="
black" />
          <path id="path1" ref={this.path1Ref} d="M35 4C35 4 19.5 24
19.5 28.5C19.5 33 38.5 56 38.5 56" stroke="black" />
          <path id="path2" ref={this.path2Ref}
            d="M1.00002 14C2.24347 7.31013 30 13 9.00002 2C21.2046 -
0.449153 26.0539 2.28554 33 8.5C45 19.236 44.5 43 33 52.5C21.5 62 14 55
6.99999 54"
            stroke="black" />
          </svg>
        </div>
      </div>
    );
}

```

02

Version