

Customize iframe elements with CSS



The Clover iframe lets you customize the payment and checkout page to match your website branding. You can use an object containing Cascading Style Sheets (CSS) to define the style of the iframe and the elements inside it.

Define iframe elements

When creating an iframe, use your custom JavaScript style object to:

- Define the iframe style. The iframe always inherits 100% of the width and height of its container element. Use the CSS applied to the container to change the iframe dimensions.
- Define the styles inside the iframe, such as the border and custom font of card and page elements:
 - `cardNumber`
 - `cardHolderName`
 - `cardDate`
 - `cardCvv`
 - `cardPostalCod`
 - `cardStreetAddress`
 - `paymentRequestButton`

For more information on card and page elements, see [Clover iframe features](#).

Code sample

JavaScript

```
// const clover = new Clover('7e525404872b8186b82b5057cfacebff');  
const clover = new Clover('080391318a9041e4d122082d369b64c7', {  
  merchantId: 'M1V5CF1WV0FSJ'
```

```

    });
    const elements = clover.elements();
    const styles = {
      body: {
        fontFamily: 'Roboto, Open Sans, sans-serif',
        fontSize: '16px',
      },
      input: {
        fontSize: '20px'
      }
    };
    const cardNumber = elements.create('CARD_NUMBER', styles);
    const cardName = elements.create('CARD_NAME', styles);
    const cardDate = elements.create('CARD_DATE', styles);
    const cardCvv = elements.create('CARD_CVV', styles);
    const cardPostalCode = elements.create('CARD_POSTAL_CODE', styles);
    const cardStreetAddress = elements.create('CARD_STREET_ADDRESS', styles);
    cardNumber.mount('#card-number');
    cardName.mount('#card-name');
    cardDate.mount('#card-date');
    cardCvv.mount('#card-cvv');
    cardPostalCode.mount('#card-postal-code');
    cardStreetAddress.mount('#card-street-address');

    // const paymentRequest = clover.paymentRequest(paymentReqData);
    const paymentRequestButton = elements.create('PAYMENT_REQUEST_BUTTON', {
      paymentReqData
    });
    paymentRequestButton.mount('#payment-request-button');
    paymentRequestButton.addEventListener('paymentMethod', function(ev) {
      alert(JSON.stringify(ev));
    })
    paymentRequestButton.addEventListener('paymentMethodStart', function(ev) {
      console.log("Gpay in progress");
    })
    // });

```

Design the form submission and GPay button

- **Form submission button**—If you are adding a form submission button, use your website CSS container element to define the style and dimension of the button. This lets you create a button that matches your website style standards.
- **GPay button**—If you are adding Google Pay™ option in your payment form, you must create a `#paymentRequestButton` container in your website CSS to define the size and width of the button.

See [Clover iframe features](#).

Code sample

In this example, the form submission button—Submit Payment—is `.button-container`, and the GPay button container is `#paymentRequestButton`. All Google Pay assets and buttons must follow the [Google Pay Web Brand Guidelines](#).

CSS

```
.container .button-container {
  display: flex;
  flex-direction: row;
  justify-content: center;
}

.container .button-container button {
  background-color: #xxxxxx;
  border:
  border-radius: xx;
  color: white;
  font-size: xx;
  display: xxxx;
  height: xx;
  width: xx;
}

#paymentRequestButton. {
  width: 85%;
  height: 40px;
  margin: 0 auto;
}

@media (min-width: 200px) {
  body {
    width: auto;
  }
}

@media (min-width: 600px) {
  body {
    width: 600px
  }
}

.hr {
  width: 100%;
  height: 10px;
  border-bottom: 1px solid black;
```



```
text-align: center;
margin: 20px 0;
}

.hr span {
```

 Updated 12 months ago

← Add Google Pay button to the
Clover iframe

Integrate with Apple Pay:
Implementation options →

Did this page help you?  Yes  No