

How To Create OTP Input Field Using HTML and CSS?

In this article, we will be learning how to create an OTP Input Field using simple HTML, CSS and JavaScript concepts. By the end of this article, you will create your own OTP input field and you will have a clear knowledge of how to create an OTP input field.

Hello, Coders! So far, we've learnt how to make pop-up logins, signup forms, and login forms. This post will teach you how to make an OTP Input Field using HTML, CSS, and JavaScript.

A one-time password, also known as a one-time PIN, one-time authorization code or dynamic password, is a password that is valid for only one login session or transaction, on a computer system or other digital device. By entering a one-time password, it is an authentication procedure that enables users to access any website. With the aid of this authentication system, users can log in safely using the OTP method.

Table of Contents

Step1: Html Otp Input Field Code:-

HTML is used to design the website's layout. So, first, we create the layout, then we style it, and finally, we add features to the button (on clicking the button menu opens).



SOURCECODE

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width,
initial-scale=1.0">

  <title>OTP Input field</title>

  <link rel="stylesheet" href="style.css">

</head>

<body>

  <html lang="en">
```

```
<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width,
initial-scale=1.0">

  <title>Autofocus Field</title>

  <link rel="stylesheet" type="text/css" href="otp.css">

</head>

<body>

  <div class="container">

    <h1>ENTER OTP</h1>

    <div class="userInput">

      <input type="text" id='ist' maxlength="1"
onkeyup="clickEvent(this, 'sec') ">

      <input type="text" id="sec" maxlength="1"
onkeyup="clickEvent(this, 'third') ">

      <input type="text" id="third" maxlength="1"
onkeyup="clickEvent(this, 'fourth') ">

      <input type="text" id="fourth" maxlength="1"
onkeyup="clickEvent(this, 'fifth') ">

      <input type="text" id="fifth" maxlength="1">

    </div>

    <button>CONFIRM</button>

  </div>

</body>
```

```
</html>

<script src="index.js"></script>

</body>

</html>
```

As you may have observed, most one-time passwords (OTPs) have a four-digit input size. To authenticate on a website, a user must first enter their 5-digit OTP and then click the confirm button. We'll construct something resembling to that.

- Our OTP input structure will be supported by a div container that we will first design.
- "Enter OTP" will now be added as our primary heading Utilising the H1 tag.
- The OTP input area will now be made for that. For that, we'll make a div with the class "userInput," inside of which we'll place four text-only input fields, and we'll give each one a onkeyup event.
- Now using the button tag we will add a confirm button .

We have added the basic structure of our OTP input field . Let's take look at our output.

Output:

ENTER OTP

CONFIRM

Step2: Adding the CSS Code For Styling Otp Input Field

Cascading Style Sheets (CSS) is a markup language for describing the presentation of a document written in HTML or XML. CSS, like HTML and JavaScript, is a key component of the World Wide Web.

Now we will look at our CSS code.

```
body {  
  margin: 0;  
  padding: 0;  
  height: 100vh;  
  background: #000000; /* fallback for old browsers */  
  background: -webkit-linear-gradient(  
    to right,  
    #434343,  
    #000000  
  ); /* Chrome 10-25, Safari 5.1-6 */  
  background: linear-gradient(  
    to right,  
    #434343,  
    #000000  
  ); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+,  
  Safari 7+ */  
}  
  
.container {  
  display: flex;  
  flex-flow: column;  
  height: 100%;
```

```
    align-items: space-around;
    justify-content: center;
}
```

```
.userInput {
    display: flex;
    justify-content: center;
}
```

```
input {
    margin: 10px;
    height: 35px;
    width: 65px;
    border: none;
    border-radius: 5px;
    text-align: center;
    font-family: arimo;
    font-size: 1.2rem;
    background: #eef2f3;
}
```

```
h1 {
    text-align: center;
    font-family: arimo;
    color: honeydew;
}
```

```
button {
    width: 150px;
    height: 40px;
    margin: 25px auto 0px auto;
    font-family: arimo;
    font-size: 1.1rem;
    border: none;
    border-radius: 5px;
    letter-spacing: 2px;
    cursor: pointer;
    background: #616161; /* fallback for old browsers */
    background: -webkit-linear-gradient(
        to right,
        #9bc5c3,
```

```

        #616161
    ); /* Chrome 10-25, Safari 5.1-6 */
    background: linear-gradient(
        to right,
        #9bc5c3,
        #616161
    ); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+,
Safari 7+ */

}

```

After we've added the CSS code, we'll go over it step by step. To save time, you can simply copy this code and paste it into your IDE. Let us now examine our code step by step.

Step1: We'll give our body padding and a zero margin by using the body tag. Our body is sized to be that height (100 vh). The Colour of the backdrop has been added for various browsers, however, they are all the same. We've included a background that is a gradation of black and Grey.

```

body {
    margin: 0;
    padding: 0;
    height: 100vh;
    background: #000000; /* fallback for old browsers */
    background: -webkit-linear-gradient(
        to right,
        #434343,
        #000000
    ); /* Chrome 10-25, Safari 5.1-6 */
    background: linear-gradient(
        to right,
        #434343,
        #000000
    ); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+,
Safari 7+ */

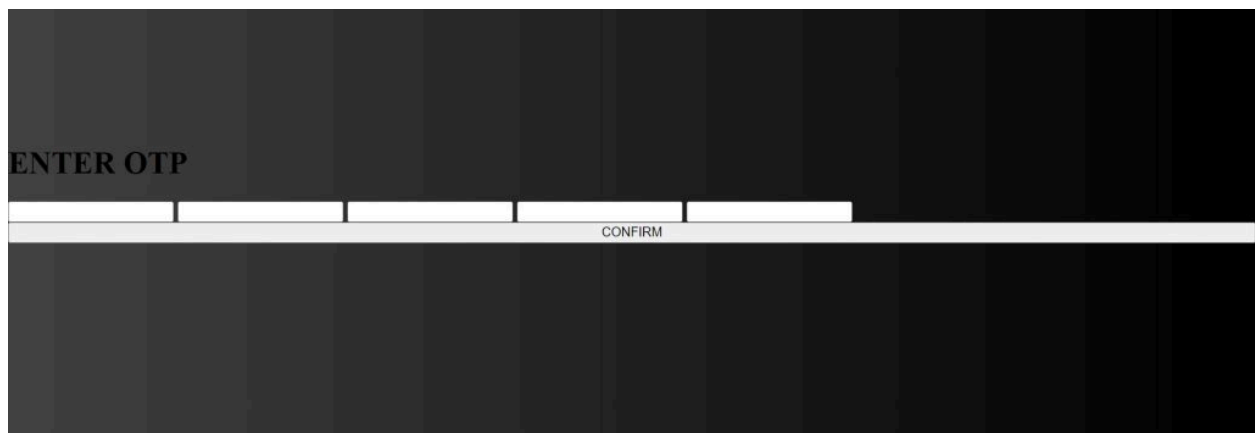
}

```



Step2: Our OTP container will now get some design. We set its presentation to be “flex,” with column-wise flow, using the class selector (.container). The container’s height is set to 100%. The objects are arranged so that the space around them is evenly distributed. “Center” has been specified as the position.

```
.container {  
  display: flex;  
  flex-flow: column;  
  height: 100%;  
  align-items: space-around;  
  justify-content: center;  
}
```



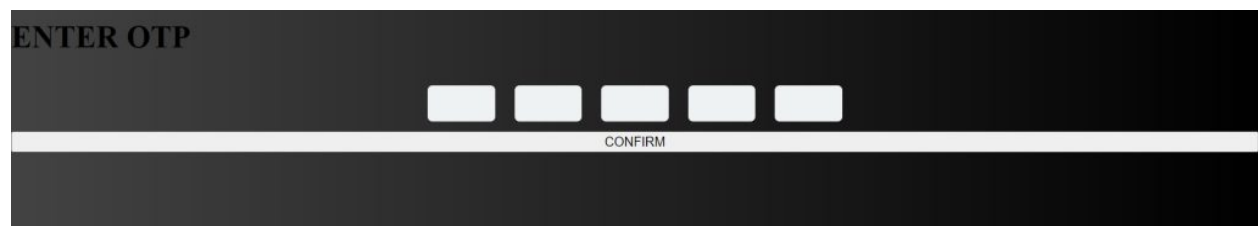
Step3: We will now style our User Input container. The objects are center-aligned, and the display is set to flex.

We'll now style our input by Utilising the tag selector. The margin is set to "10px." Our was configured with 35 and 65 pixels for height and width, respectively. We have added a 5px border radius to our input field's corner to give it some curviness. The backdrop Colour is set to light blue, and the font size is "1.2 rem."

```
.userInput {  
  display: flex;  
  justify-content: center;  
}
```

```
input {  
  margin: 10px;  
  height: 35px;  
  width: 65px;  
  border: none;  
  border-radius: 5px;  
  text-align: center;  
  font-family: arimo;  
  font-size: 1.2rem;  
  background: #eef2f3;  
}
```

```
: flex;
```



Step4: Now that our buttons and project heading have some style, we can finish. Our heading will now have styling, shall we? The “Arimo” font family is used with the text, which is positioned to the “Centre.” “Honey dew” is the selected Colour for the font.

Finally, let’s give our button some style. We chose “150px” and “40px” for the width and height, respectively. The text’s font size is set to “1.2 em.” We increased the letter spacing between the text by 2 pixels. To give our button more curve to our button edges, we also added a 5 px border radius. To make a gradient backdrop Colour for our button, we combined the Colours cyan blue and Grey

```
h1 {  
  
    text-align: center;  
  
    font-family: arimo;  
  
    color: honeydew;  
  
}  
  
button {  
  
    width: 150px;  
  
    height: 40px;  
  
    margin: 25px auto 0px auto;  
  
    font-family: arimo;  
  
    font-size: 1.1rem;  
  
    border: none;
```

```
border-radius: 5px;

letter-spacing: 2px;

cursor: pointer;

background: #616161; /* fallback for old browsers */

background: -webkit-linear-gradient(

  to right,

    #9bc5c3,

    #616161

); /* Chrome 10-25, Safari 5.1-6 */

background: linear-gradient(

  to right,

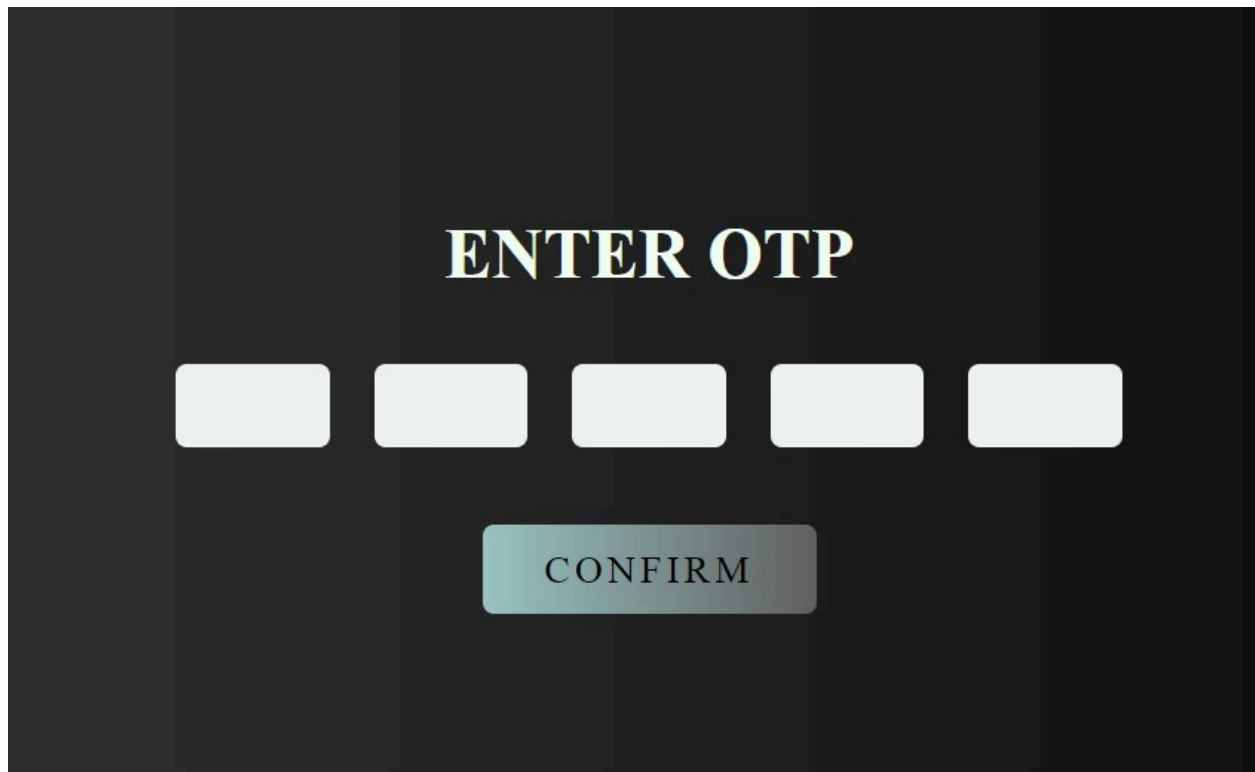
    #9bc5c3,

    #616161

); /* W3C, IE 10+/ Edge, Firefox 16+, Chrome 26+, Opera 12+,
Safari 7+ */

}
```

Output:



Step3: Adding JavaScript Code

```
function clickEvent(first,last){  
    if(first.value.length){  
        document.getElementById(last).focus();  
    }  
  
}
```

We recently added the function ClickEvent with the parameters first and last to javascript. This function will simply determine whether the first input contains the value before and will automatically focusing on the second and subsequent inputs for the third and fourth input.

The project is now finished, we have completed the OTP input field using HTML, CSS and Javascript. Now, look at the live preview.

Final Output Of Otp Input Field Html Css:-

Video Player

https://codewithrandom.com/wp-content/uploads/2022/10/Untitled-7.mp4?_=1

Now We have Successfully created our OTP Input field using HTML , CSS and JavaScript. You can use this project directly by copying into your IDE. WE hope you understood the project