



BridgeLabz

Employability Delivered

HTML, CSS & Javascript



Work with HTML 5 Inputs and Forms

HTML 5 Form Input Fields

text input type for Name: *Name is Incorrect*

password input type:

email input type:

url input type:

number input type:

tel input type:

search input type:

Choose a date and time: 

Choose your salary:



[Please Try on Mobile HTML Input Form](#)

- [Email Field](#)
- [URL Field](#)
- [Number Field](#)
 - [Number Field Attributes](#)
- [Range Fields](#)
 - [Range Field Attributes](#)
- [Tel Fields](#)
- [Search Fields](#)
- [Date Field](#)
- [Time Field](#)
- [Datetime Field](#)
- [Month Field](#)
- [Week Field](#)
- [Color Field](#)
- [Combo Box Fields](#)
- [New Attributes](#)
 - [autofocus](#)
 - [placeholder](#)
 - [required](#)
 - [Multiple](#)
 - [pattern](#)
 - [autocomplete](#)
 - [min + max](#)
 - [step](#)
 - [form](#)

HTML 5 Input Forms Decoded

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <title>Basic input types</title>
    <link rel="stylesheet" href="../css/simple_form.css">
  </head>
  <body>
    <form>
      <header class="headerContainer"><h1>HTML 5 Form Input Fields</h1></header>
      <div>
        <label for="text">text input type for Name: </label>
        <input type="text" id="text" name="text">
        <error-output class="text-error" for="text"></error-output>
      </div>
      <div>
        <label for="pwd">password input type: </label>
        <input type="password" id="pwd" name="pwd">
      </div>
      <div>
        <label for="email">email input type: </label>
        <input type="email" id="email" name="email">
      </div>
      <div>
        <label for="url">url input type: </label>
        <input type="url" id="url" name="url">
      </div>
      <div>
        <label for="number">number input type: </label>
        <input type="number" id="number" name="number">
      </div>
      <div>
        <label for="tel">tel input type: </label>
        <input type="tel" id="tel" name="tel">
      </div>
```

```
<div>
  <label for="search">search input type: </label>
  <input type="search" id="search" name="search">
</div>
<div class="nativeDateTimePicker">
  <label for="party">Choose a date and time:</label>
  <input type="datetime-local" id="party" name="bday">
  <span class="validity"></span>
</div>
<div>
  <label for="salary">Choose your salary: </label>
  <input type="range" name="salary" id="salary" min="300000"
         max="500000" step="100" value="400000">
  <output class="salary-output" for="salary">400000</output>
</div>
<div>
  <button>Submit</button>
</div>
</form>
<script>
  const text = document.querySelector('#text');
  const textError = document.querySelector('.text-error');
  text.addEventListener('input', function() {
    let nameRegex = RegExp('^[A-Z]{1}[a-z]{2,}$');
    if (nameRegex.test(text.value))
      textError.textContent = "";
    else textError.textContent = "Name is Incorrect";
  });

  const salary = document.querySelector('#salary');
  const output = document.querySelector('.salary-output');
  output.textContent = salary.value;
  salary.addEventListener('input', function() {
    output.textContent = salary.value;
  });
</script>
</body>
</html>
```

HTML 5 Input Forms CSS Decoded

```
css > # simple_form.css > ...
1  html, input {
2    font-family: sans-serif;
3  }
4
5  body {
6    width: 100%;
7    max-width: 600px;
8    margin: 0px auto;
9    padding: 2em;
10 }
11
12 .headerContainer {
13   display: flex;
14   flex-flow: column wrap;
15   align-content: center;
16   background: #tomato;
17   font-weight: bold;
18   margin-bottom: 20px;
19 }
20
21 form {
22   margin-top: 50px;
23 }
24
25 div {
26   margin-bottom: 20px;
27   display: flex;
28   justify-content: space-between;
29   align-items: center;
30 }
31
32 label, input, button {
33   font-size: 16px;
34   line-height: 1.5
35 }
36
37 label {
38   text-align: right;
39   width: 30%;
40   margin-right: 2%;
41 }
42
43 input {
44   flex: auto;
45 }
46
47 button {
48   width: 70%;
49   margin: 0 auto;
50 }
51
52 error-output {
53   font-size: 12px;
54   font-style: italic;
55   color: #red;
56 }
```

JS Events

DOM Events	Event Names	HTML Attribute	DOM Property	Event Listener
Mouse events	click	onclick	elem.onclick	elem.addEventListener ('click', function() {})
	Contextmenu	oncontextmenu	elem.oncontextmenu	elem.addEventListener ('contextmenu', function() {})
	mouseover / mouseout	onmouseover / onmouseout	elem.onmouseover / elem.onmouseout	elem.addEventListener ('mouseover', function() {})
	mousedown / mouseup	onmousedown / onmouseup	elem.onmousedown / elem.onmouseup	elem.addEventListener ('mousedown', function() {})
	mousemove	onmousemove	elem.onmousemove	elem.addEventListener ('mousemove', function() {})
Keyboard events	keydown	onkeydown	elem.onkeydown = function() {}	elem.addEventListener ('keydown', function() {})
	keyup	onkeyup	elem.onkeyup = function() {}	elem.addEventListener ('keyup', function() {})
Form element events	submit	onsubmit	elem.onsubmit = function() {}	elem.addEventListener ('submit', function() {})
	focus	onfocus	elem.onfocus = function() {}	elem.addEventListener ('focus', function() {})
input, select, or textarea	input	oninput	elem.oninput	elem.addEventListener ('input', function() {})
CSS events	transitionend	ontransitionend	elem.ontransitionend	elem.addEventListener ('transitionend', function() {})
Document events	DOMContentLoaded	onDOMContentLoaded	elem.onDOMContentLoaded	elem.addEventListener ('DOMContentLoaded', function() {})

HTML

Input Event Handling

```
1 <input value="Click me" onclick="alert('Click!')" type="button">
```

Click me

Click and Try Inline Function

```
1 <script>
2   function countRabbits() {
3     for(let i=1; i<=3; i++) {
4       alert("Rabbit number " + i);
5     }
6   </script>
7
8 <input type="button" onclick="countRabbits()" value="Count rabbits!">
```

Count rabbits!

Click and Try Function Call

JavaScript

```
1 const input = document.querySelector('input');
2 const log = document.getElementById('values');
3
4 input.addEventListener('input', updateValue);
5
6 function updateValue(e) {
7   log.textContent = e.target.value;
8 }
```

```
1 <input id="elem" type="button" value="Click me">
2 <script>
3   elem.onclick = function() {
4     alert('Thank you');
5   };
6 </script>
```

Click me

Click and Try on adding DOM Property

JS Event Samples

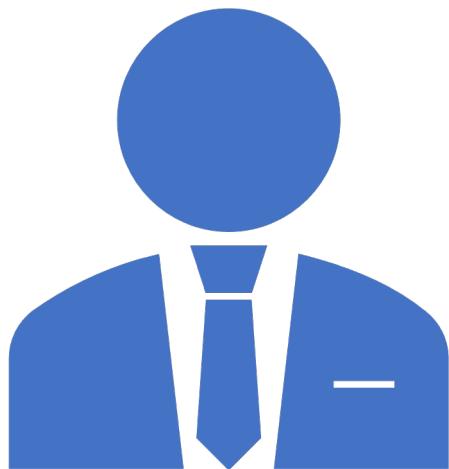
Modify and Play



UC 1

**As a User need to
enter a valid First
Name**

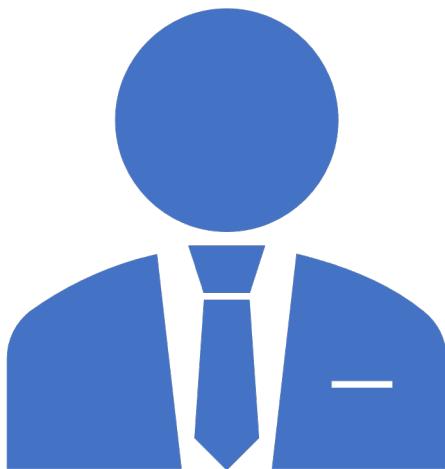
- First name starts with Cap and has minimum 3 characters



UC 2

As a User need to enter a valid email

- E.g. abc.xyz@bl.co.in
- Email has 3 mandatory parts (abc, bl & co) and 2 optional (xyz & in) with precise @ and . positions



UC 3

As a User need to follow pre-defined Mobile Format

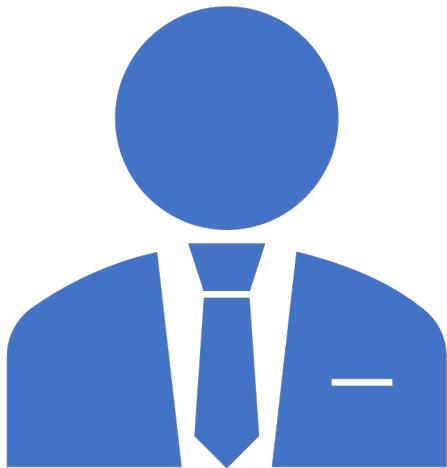
- E.g. 91 9919819801
- Country code follow by space and 10 digit number



UC 4

As a User need to
follow pre-defined
Password rules.
**Rule1 – minimum 8
Characters**

- NOTE – All rules must be passed



UC 5

Rule1 – minimum 8 Characters

**Rule2 – Should have at least 1
Upper Case**

**Rule3 – Should have at least 1
numeric number in the password**

**Rule4 – Has exactly 1 Special
Character**

- NOTE – All rules must be passed

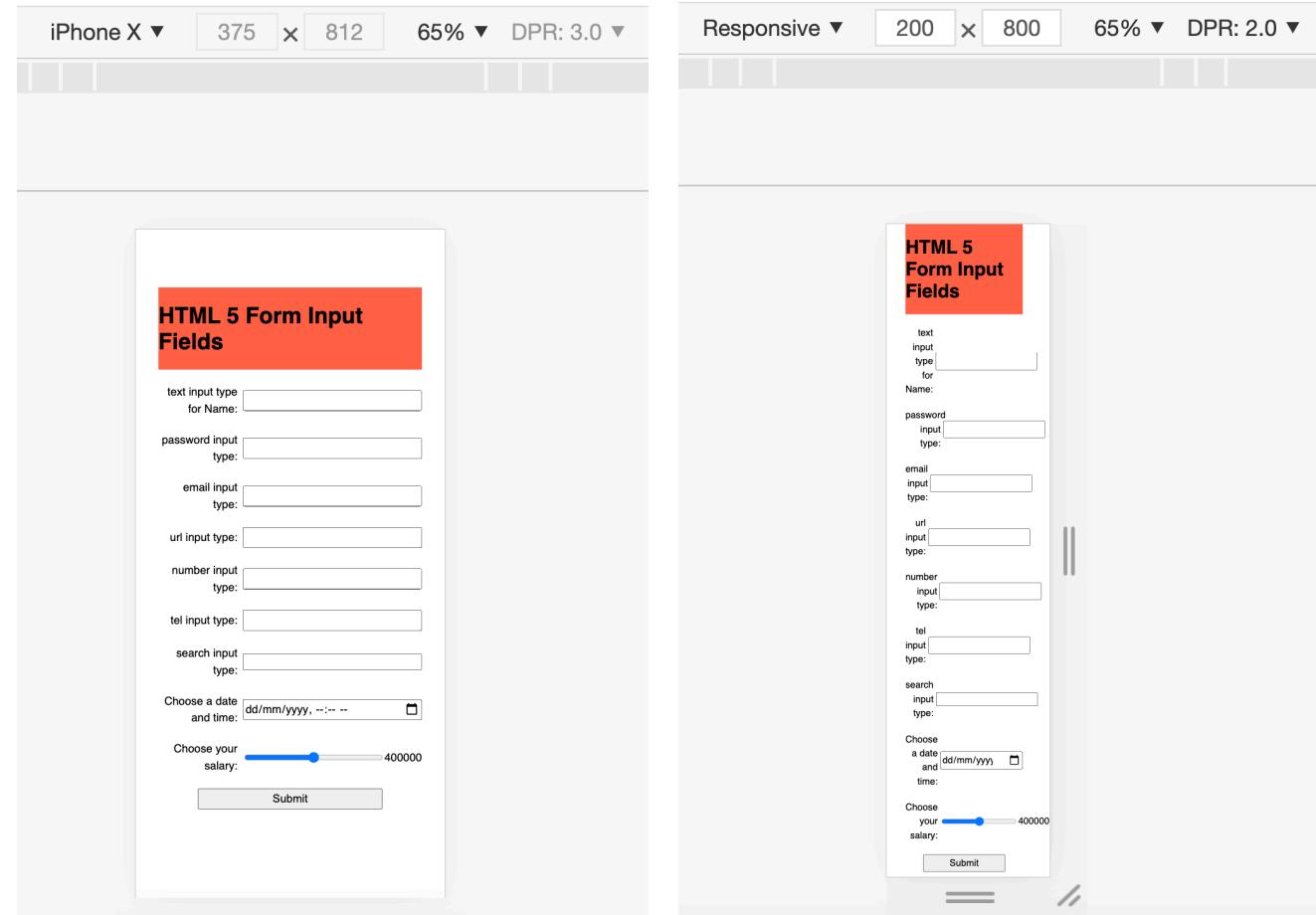


Media Queries and Responsive Designs

Responsive Design & Media Queries

- Media queries can **modify the appearance** (and even behavior) of a website or app based on a matched set of conditions about the user's device, browser or system settings.
- Media queries are commonly associated with CSS, but they can be used in HTML and JavaScript as well.

```
<!-- Served to all users -->
<link rel="stylesheet" href="all.css" media="all" />
<!-- Served to screens that are at least 20em wide --&gt;
&lt;link rel="stylesheet" href="small.css" media="(min-width: 20em)" /&gt;</pre>
```





Anatomy of a Media Query

@media screen (min-width: 320px) and (max-width: 768px)

AT-RULE

MEDIA TYPE

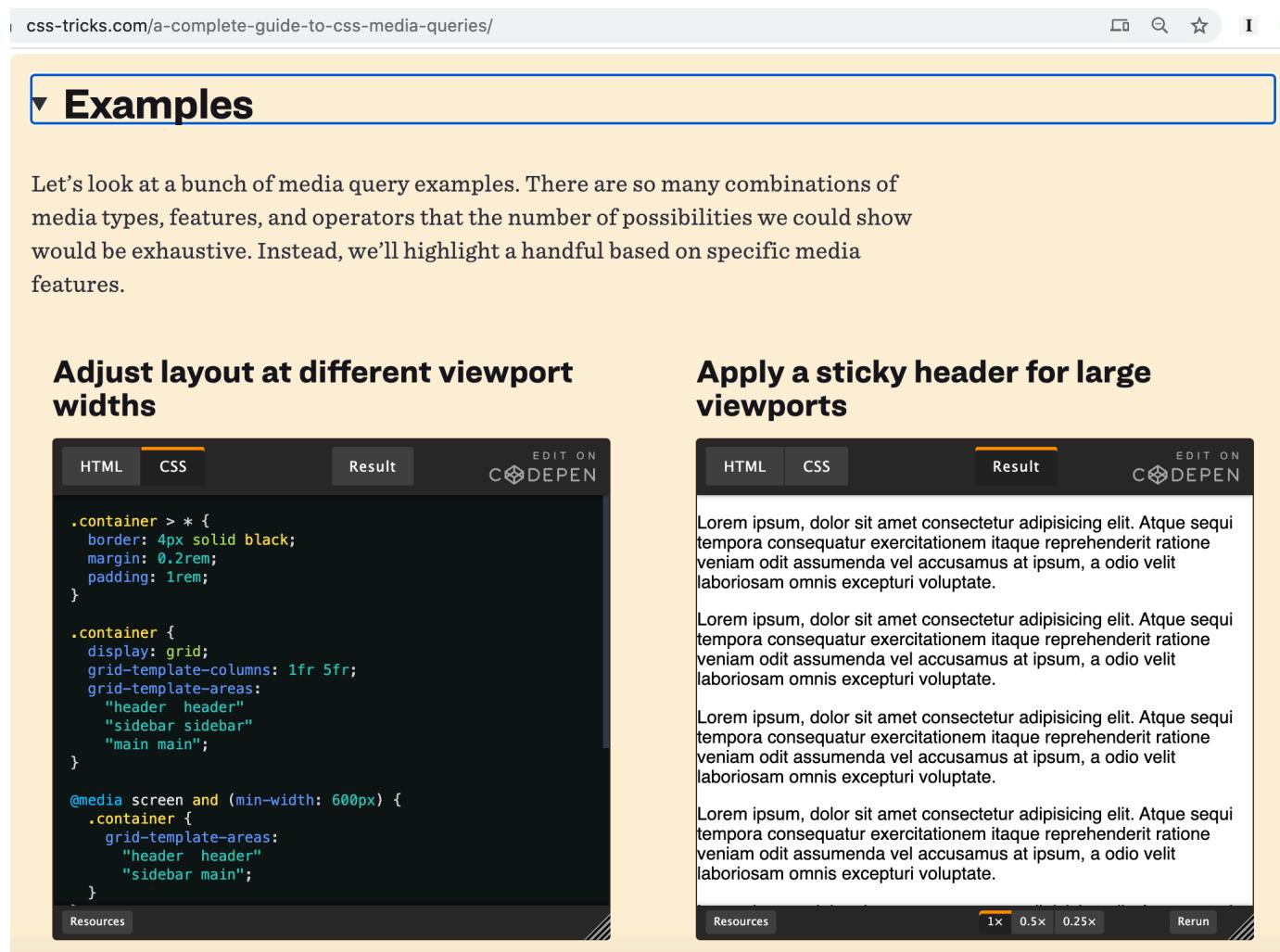
MEDIA FEATURE

OPERATOR

MEDIA FEATURE

- **AT RULE** – The first ingredient in a media query recipe is the **@media** rule itself
- **MEDIA TYPE** – Type of media, for e.g. **screen** – matches devices with **screen**. Others are all, print, speech...
- **MEDIA FEATURE** – Features of the media/device. For **screen** type both min-width and max-width are feature. Please click the [link](#) for more details.
- **OPERATORS** – They are and, or (comma separated) and not.

Click here - Media Queries Example



The screenshot shows a browser window displaying a guide to CSS media queries. The URL in the address bar is css-tricks.com/a-complete-guide-to-css-media-queries/. The page features a section titled "Examples" with two examples shown side-by-side on CodePen.

Example 1: Adjust layout at different viewport widths

Code:

```
.container > * {  
  border: 4px solid black;  
  margin: 0.2rem;  
  padding: 1rem;  
}  
  
.container {  
  display: grid;  
  grid-template-columns: 1fr 5fr;  
  grid-template-areas:  
    "header header"  
    "sidebar sidebar"  
    "main main";  
}  
  
@media screen and (min-width: 600px) {  
  .container {  
    grid-template-areas:  
      "header header"  
      "sidebar main";  
  }  
}
```

Example 2: Apply a sticky header for large viewports

Code:

```
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Atque sequi tempora consequatur exercitationem itaque reprehenderit ratione veniam odit assumenda vel accusamus at ipsum, a odio velit laboriosam omnis excepturi voluptate.  
  
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Atque sequi tempora consequatur exercitationem itaque reprehenderit ratione veniam odit assumenda vel accusamus at ipsum, a odio velit laboriosam omnis excepturi voluptate.  
  
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Atque sequi tempora consequatur exercitationem itaque reprehenderit ratione veniam odit assumenda vel accusamus at ipsum, a odio velit laboriosam omnis excepturi voluptate.  
  
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Atque sequi tempora consequatur exercitationem itaque reprehenderit ratione veniam odit assumenda vel accusamus at ipsum, a odio velit laboriosam omnis excepturi voluptate.
```

References

- [HTML Basics](#)
- [CSS Basics](#)
- [HTML / CSS in a nutshell](#)
- [HTML Element Ref](#)
- [Design webpage structure](#)
- [Flex Box](#)
- [A Guide to Flex Box](#)
- [Flex Layout Example](#)
- [Flex Live Demos](#)
- [Flex Try it out](#)
- [HTML-CSS-JS Try it out](#)
- [Open in Mobile HTML Form](#)
- [HTML Form Fields](#)
- [HTML 5 Input Fields](#)
- [Complete Guide to CSS Media Queries](#)



BridgeLabz

Employability Delivered

Thank
You