

Note (For Remember): The box-shadow property can take 5-arguments: x, y, softness, spread, color.

```
box-shadow: 0 0 0 1px gold;
```

CSS FRAMEWORKS

- Bootstrap
- Foundation
- Semantic UI
- UI Kit
- Materialize
- Milligram

```
textarea {  
  resize: none;  
}
```

For Text-Field elements we have many attributes that we can apply:

- The placeholder.
- The value.
- The maxlength && minlength: for String fields like text, email, ...etc.
- The disabled; also, can be used for checkbox && Radio.
- The readonly.
- The autofocus.
- The list.
- The autocomplete="off".
- The selected for select-element.
- The multiple for select-element && file.
- The checked for checkbox && Radio.
- The accept for file.
- The min && max for numbers.

The textarea-element doesn't have value-attribute, we should set the text between the element and closed element: `<textarea>Value Here</textarea>`.

```
<input type="text" list="my-country" autocomplete="off">
  <datalist id="my-country">
    <option>Syria</option>
    <option>Damascus</option>
    <option>Lattakia</option>
    <option>Tartous</option>
  </datalist>
```

Note: we must set the value-attribute like: `value=""` to tell the browser this is only message.

```
<select required multiple>
  <option value="">Default</option>
  <option value="1">NodeJS</option>
  <option value="2">React</option>
  <option value="3">Flask</option>
</select>
```

```

<select required>
  <option value="">Select a Course...</option>
  <optgroup label="Front-End">
    <option value="1">HTML</option>
    <option value="2">React</option>
    <option value="3">Next.JS</option>
  </optgroup>
  <optgroup label="Back-End">
    <option value="4">NodeJS</option>
    <option value="5">Flask</option>
    <option value="6">FastAPI</option>
    <option value="7">Strapi</option>
    <option value="8">Laravel</option>
  </optgroup>
</select>

```

The name-attribute is used to group the radio buttons.

To make multiple files selected by type, we separate them by comma.

```

<div>
  <input type="file" name="" id="" accept=".pdf">
</div>

```

Note: we can use mime-type to select the files:

```

<div>
<input type="file" name="" id="" accept="image/*, application/json,
application/pdf">
</div>

```

The other approach to grouping the elements, is using *section-element* instead of *fieldset-element*, and *h2-element* instead of *legend-element*.

TRANSFORMATIONS

- rotate()
- scale()
- skew()
- translate()

The positive values for rotating in clock wise.

The Negative Values for rotating in

```
.box:hover{
  transform: rotate(15deg);
  -webkit-transform: rotate(15deg);
  -moz-transform: rotate(15deg);
  -ms-transform: rotate(15deg);
  -o-transform: rotate(15deg);
}
```

```
.box{
  width: 100px;
  height: 100px;
  background-color: lightgreen;
  padding: 10px;
  /*box-sizing: border-box;*/
  transition: transform 0.5s;
  -webkit-transition: transform 0.5s;
  -moz-transition: transform 0.5s;
  -ms-transition: transform 0.5s;
  -o-transition: transform 0.5s;
}
```

```
.box:hover{
  transform: rotate(15deg) scale(1.5);
  -webkit-transform: rotate(15deg) scale(1.5);
  -moz-transform: rotate(15deg) scale(1.5);
  -ms-transform: rotate(15deg) scale(1.5);
  -o-transform: rotate(15deg) scale(1.5);
}
```

```
.box:hover{
  transform: translate(15px);
  -webkit-transform: translate(15px);
  -moz-transform: translate(15px);
  -ms-transform: translate(15px);
  -o-transform: translate(15px);
}
```

Here if we supply one value it will for horizontal-axis.

If we supply two values, they will for *horizontal-axis* && *vertical-axis*.

```
.box:hover{
  transform: translate(15px, 15px);
  -webkit-transform: translate(15px, 15px);
  -moz-transform: translate(15px, 15px);
  -ms-transform: translate(15px, 15px);
  -o-transform: translate(15px, 15px);
}
```

For transform we have many Values or Attributes:

- The rotate function: this is for rotating on z-axis.
- The rotateX function.
- The rotateY function.
- The rotate function rotate the object from center.
- The translate function.
- The skew function.
- The scale function.
- The translateX function.
- The translateY function.
- The translateZ function.

Note: The order of these functions is important, the browser will execute the transform in sequential order.

When we set the translateZ, we must define the perspective-function:

The positive values will make it closer to screen.

The negative values will make it away from screen.

```
.box:hover {  
  transform: perspective(200px) translateZ(50px);  
  -webkit-transform: perspective(200px) translateZ(50px);  
  -moz-transform: perspective(200px) translateZ(50px);  
  -ms-transform: perspective(200px) translateZ(50px);  
  -o-transform: perspective(200px) translateZ(50px);  
}
```

In this way we can make 3d-slider: (search for it)

```
.box:hover {  
  transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  -webkit-transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  -moz-transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  -ms-transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  -o-transform: perspective(200px) translateZ(50px) rotateY(45deg);  
}
```

The *transform-origin* to change the point of rotation.

The *default point* is center.

The *0 0 points* are for top left corner.

We can set different units for transform-origin like px, %, ...etc.

```
.box:hover {  
  transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  -webkit-transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  -moz-transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  -ms-transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  -o-transform: perspective(200px) translateZ(50px) rotateY(45deg);  
  transform-origin: 10px 10px;  
}
```

To make group of animation:

```
<div class="container">
  <div class="box">
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
  </div>
  <div class="box">
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
  </div>
</div>
```

```
.container {
  transition: transform 0.5s;
  -webkit-transition: transform 0.5s;
  -moz-transition: transform 0.5s;
  -ms-transition: transform 0.5s;
  -o-transition: transform 0.5s;
}

.container:hover {
  transform: perspective(200px) translateZ(50px) rotateY(45deg);
  -webkit-transform: perspective(200px) translateZ(50px) rotateY(45deg);
  -moz-transform: perspective(200px) translateZ(50px) rotateY(45deg);
  -ms-transform: perspective(200px) translateZ(50px) rotateY(45deg);
  -o-transform: perspective(200px) translateZ(50px) rotateY(45deg);
  transform-origin: 10px 10px;
}
```

For make clean code we should add .box.

For make main perspective for all container children:

```
.container {
  transition: transform 0.5s;
  -webkit-transition: transform 0.5s;
  -moz-transition: transform 0.5s;
  -ms-transition: transform 0.5s;
  -o-transition: transform 0.5s;
  perspective: 200px;
}

.container:hover .box {
  transform: rotateY(45deg);
  -webkit-transform: rotateY(45deg);
  -moz-transform: rotateY(45deg);
  -ms-transform: rotateY(45deg);
  -o-transform: rotateY(45deg);
  transform-origin: 10px 10px;
}
```

```
.container:hover .box-special{
  transform: rotateX(45deg);
  -webkit-transform: rotateX(45deg);
  -moz-transform: rotateX(45deg);
  -ms-transform: rotateX(45deg);
  -o-transform: rotateX(45deg);
}

<div class="container">
  <div class="box">
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
  </div>
  <div class="box box-special">
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
  </div>
</div>
```
