CSC-318
Web Technology
(BSc CSIT, TU)

Ganesh Khatri
kh6ganesh@gmail.com

CSS3 Gradients

- let you display smooth transitions between two or more specified colors
- Earlier, you had to use images for these effects
- However, by using CSS3 gradients you can reduce download time and bandwidth usage
- In addition, elements with gradients look better when zoomed, because the gradient is generated by the browser

CSS3 Gradients: Linear Gradient

```
#grad1 {
    width:460px;
    height: 200px;
    background: red;
    background: -webkit-linear-gradient(red, yellow);
    /* For Safari 5.1 to 6.0 */
    background: -o-linear-gradient(red, yellow);
    /* For Opera 11.1 to 12.0 */
    background: -moz-linear-gradient(red, yellow);
    /* For Firefox 3.6 to 15 */
    background: linear-gradient(red, yellow);
}
```

Linear Gradient - Top to Bottom

This linear gradient starts at the top. It starts red, transitioning to yellow:

Note: Internet Explorer 9 and earlier versions do not support gradients.

CSS3 Gradients: Linear Gradient(left to right)

```css
#grad1 {
    height: 200px;
    background: red;
    background: -webkit-linear-gradient(left, red , yellow);
    /* For Safari 5.1 to 6.0 */
    background: -o-linear-gradient(right, red, yellow);
    /* For Opera 11.1 to 12.0 */
    background: -moz-linear-gradient(right, red, yellow);
    /* For Firefox 3.6 to 15 */
    background: linear-gradient(to right, red , yellow);
    /* Standard syntax (must be last) */
}
```

Linear Gradient - Left to Right

This linear gradient starts at the left. It starts red, transitioning to yellow:

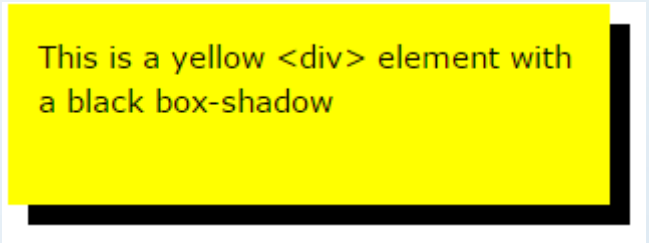
Note: Internet Explorer 9 and earlier versions do not support gradients.

CSS3 Shadow Effects

- With CSS3 you can add shadow to text and to elements
  - Text-shadow

Text shadow effect!

  - Box-shadow



CSS3 Text Shadow:example1

- Text-shadow: horizontal-shadow vertical shadow

```
h1 {
    text-shadow: 2px 2px;
}
```

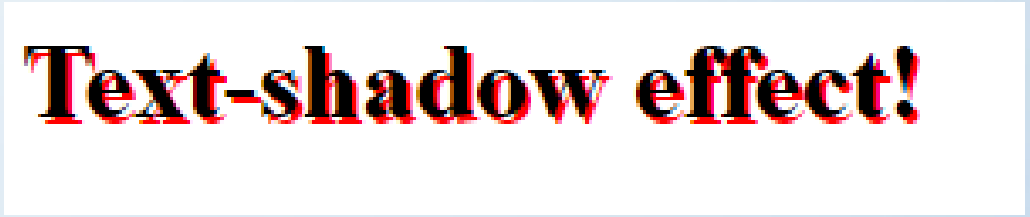
Text-shadow effect!

Note: Internet Explorer 9 and earlier versions, do not support the text-shadow property.

CSS3 Text Shadow:example2(with color)

- Text-shadow: horizontal-shadow vertical shadow

```
h1 {
    text-shadow: 2px 2px red;
}
```



CSS3 Text Shadow:example3(with blur)

- Text-shadow: horizontal-shadow vertical shadow

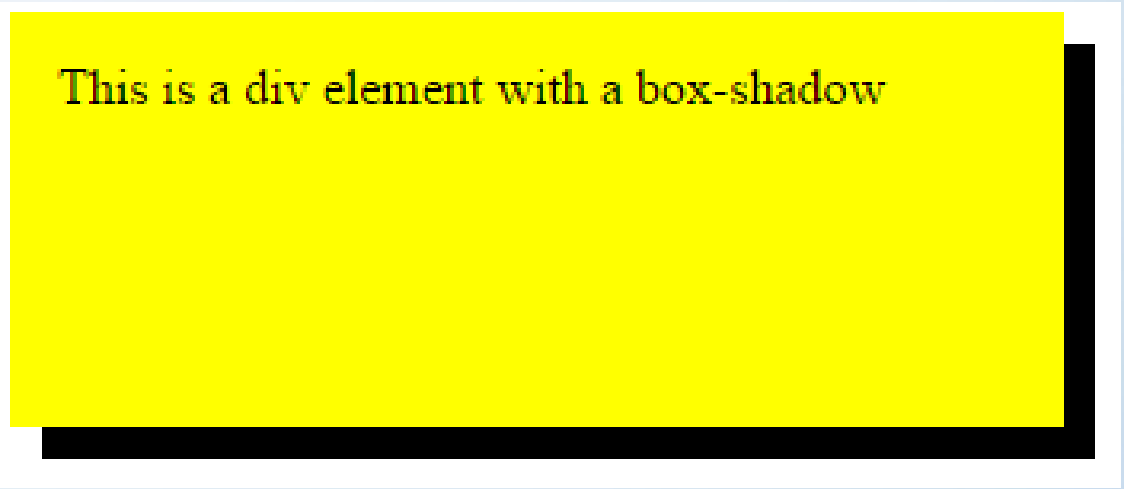
```
h1 {
    text-shadow: 2px 2px 8px red;
}
```

CSS3 Text Shadow:example4(multiple shadows)

```
h1 {
    text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;
}
```

CSS3 Box Shadow : example

```
div {
    width: 300px;
    height: 100px;
    padding: 15px;
    background-color: yellow;
    box-shadow: 10px 10px;
}
```



CSS3 Box Shadow : example(with color & blur)

```
<head>
<style>
#example1 {
  border: 1px solid;
  padding: 10px;
  box-shadow: 5px 10px 8px #888888;
}

#example2 {
  border: 1px solid;
  padding: 10px;
  box-shadow: 5px 10px 18px #888888;
}

#example3 {
  border: 1px solid;
  padding: 10px;
  box-shadow: 5px 10px 18px red;
}
</style>
</head>
```

box-shadow: 5px 10px 8px #888888:

The optional third value adds a blur effect to the shadow.

box-shadow: 5px 10px 18px #888888:

More blurred.

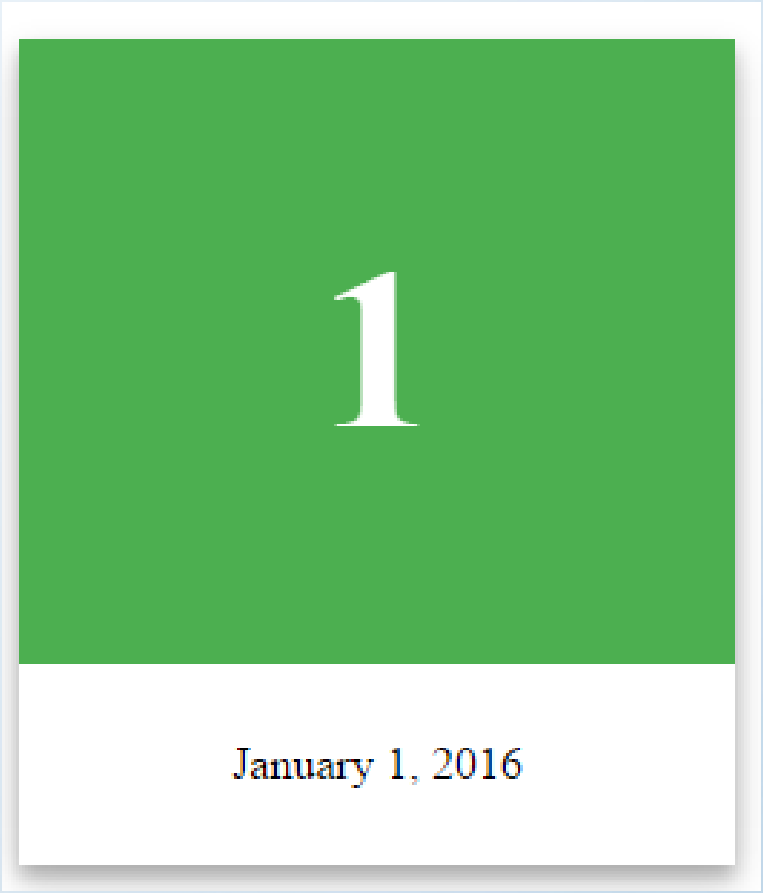
box-shadow: 5px 10px 18px red:

More blurred and red.

Activate Windows
Go to PC settings to activate Windows.

CSS3 Box Shadow : example

```
<style>
div.card {
  width: 250px;
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px
20px 0 rgba(0, 0, 0, 0.19);
  text-align: center;
}
div.header {
    background-color: #4CAF50;color: white;
    padding: 10px;font-size: 40px;
}
div.container { padding: 10px; }
</style>

<h2>Cards</h2>
<p>The box-shadow property can be used to create paper-
like cards:</p>
<div class="card">
  <div class="header">
    <h1>1</h1>
  </div>
  <div class="container">
    <p>January 1, 2016</p>
  </div>
</div>
```

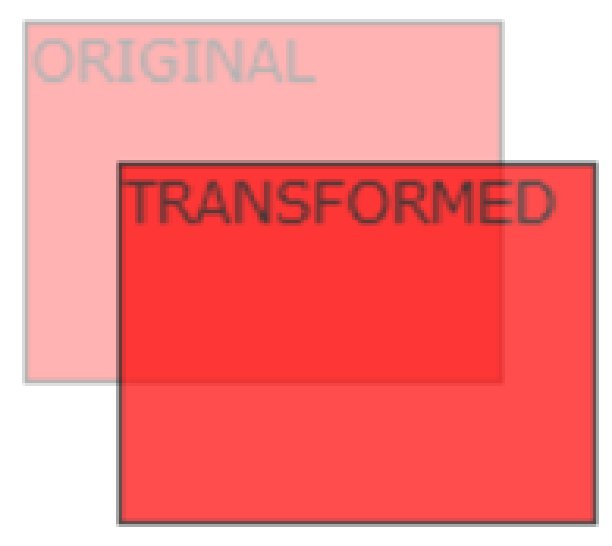


CSS3 2D Transforms

- allow you to translate, rotate, scale, and skew elements.
- is an effect that lets an element change shape, size and position.
- CSS3 supports 2D and 3D transformations
- following 2D transformation methods
  - Translate()
  - Rotate()
  - scale()
  - skewX()
  - skewY()
  - Matrix()

CSS3 2D Transforms(translate)

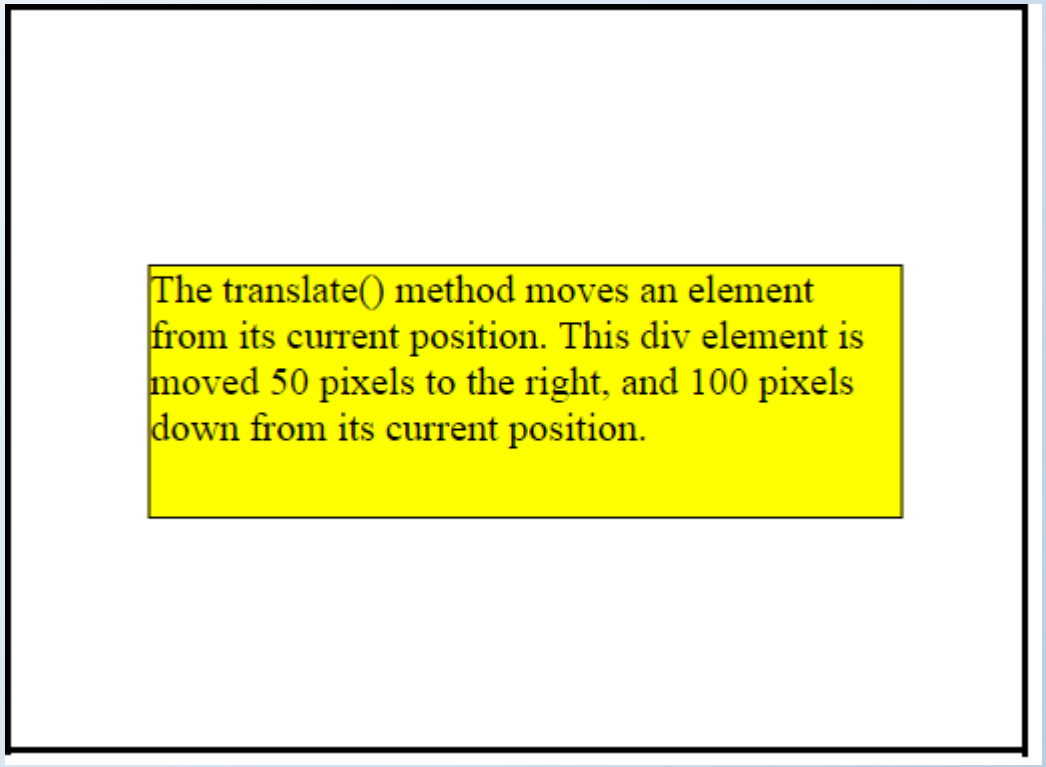
- moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).



CSS3 2D Transforms(translate)

- moves an element from its current position (according to the parameters given for the X-axis and the Y-axis).

```css
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
    -ms-transform: translate(50px,100px); /* IE 9 */
    -webkit-transform: translate(50px,100px); /* Safari */
    transform: translate(50px,100px); /* Standard syntax */
}
```



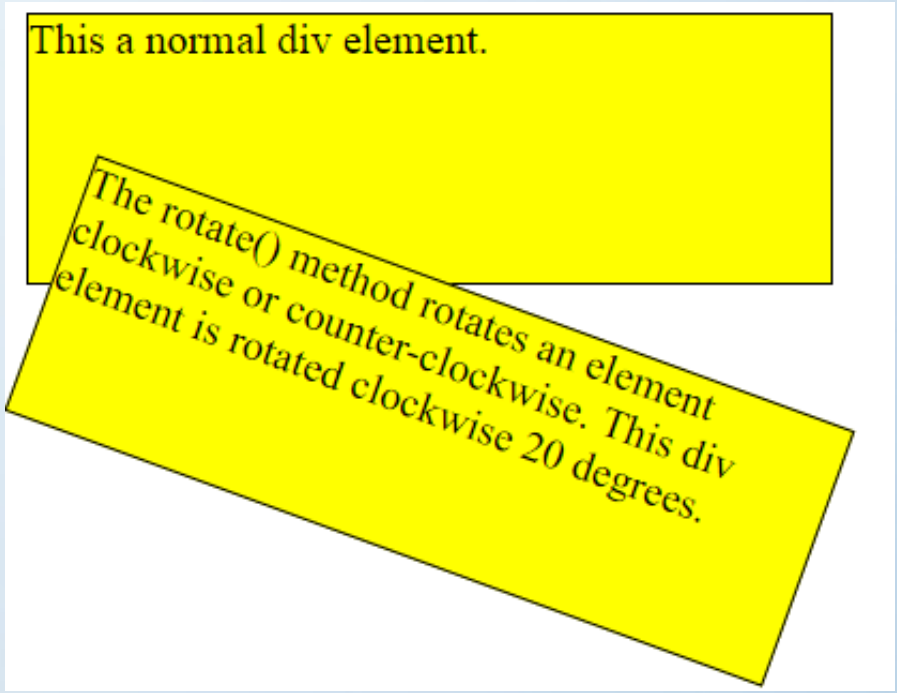
CSS3 2D Transforms(rotate)

- The rotate() method rotates an element clockwise or counter-clockwise according to a given degree

CSS3 2D Transforms(rotate clockwise)

- The rotate() method rotates an element clockwise or counter-clockwise according to a given degree

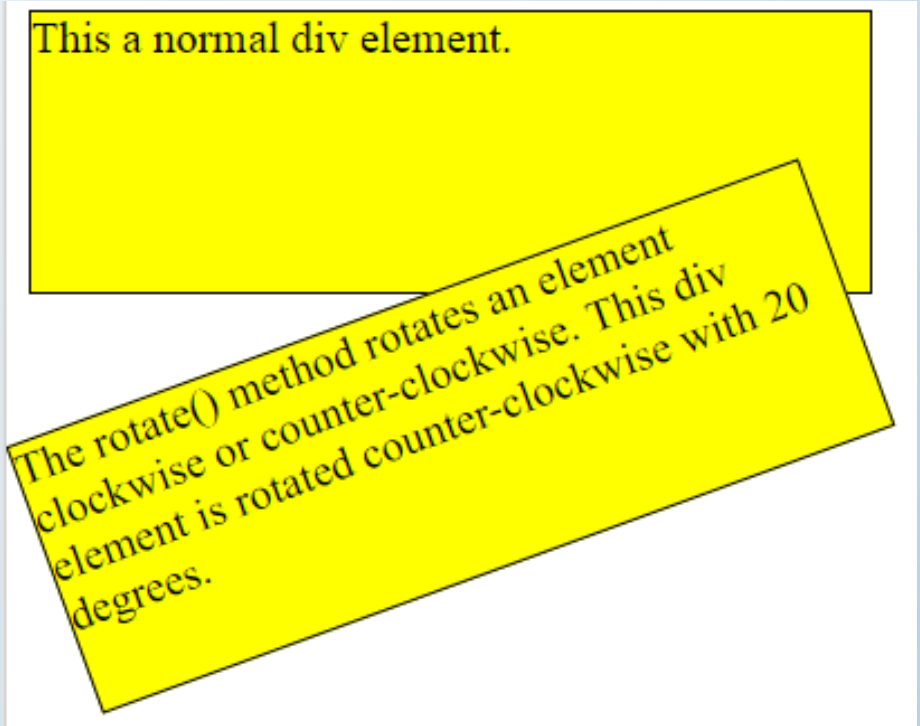
```
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
}

div#myDiv {
    -ms-transform: rotate(20deg); /* IE 9 */
    -webkit-transform: rotate(20deg); /* Safari */
    transform: rotate(20deg); /* Standard syntax */
}
```



CSS3 2D Transforms(rotate anti-clockwise)

- The rotate() method rotates an element clockwise or counter-clockwise according to a given degree

```
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
}

div#myDiv {
    -ms-transform: rotate(-20deg); /* IE 9 */
    -webkit-transform: rotate(-20deg); /* Safari */
    transform: rotate(-20deg); /* Standard syntax */
}
```



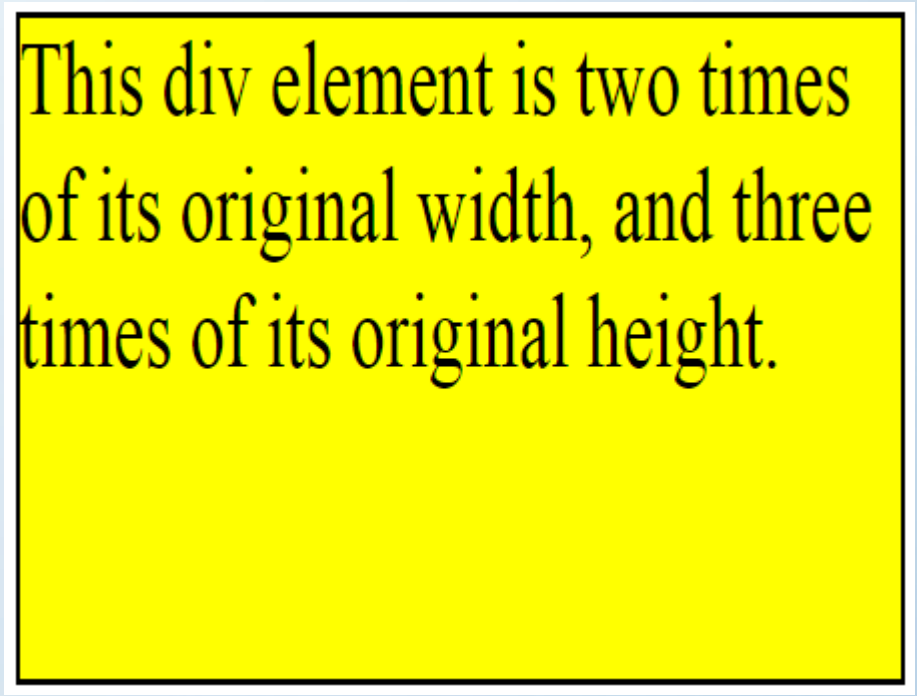
CSS3 2D Transforms(scale)

- increases or decreases the size of an element (according to the parameters given for the width and height).

CSS3 2D Transforms(scale increase)

- increases or decreases the size of an element (according to the parameters given for the width and height).

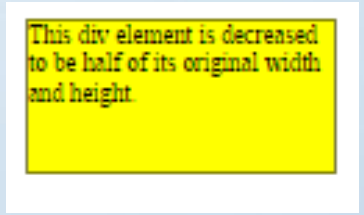
```
div {
    margin: 150px;
    width: 200px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
    border: 1px solid black;
    -ms-transform: scale(2,3); /* IE 9 */
    -webkit-transform: scale(2,3); /* Safari */
    transform: scale(2,3); /* Standard syntax */
}
```



CSS3 2D Transforms(scale decrease)

- increases or decreases the size of an element (according to the parameters given for the width and height).

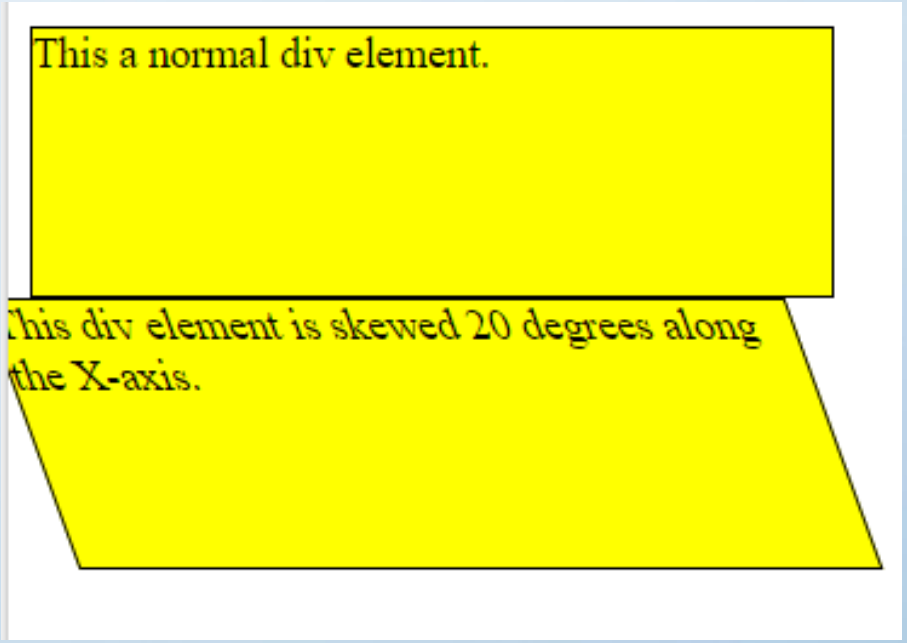
```css
div {
    margin: 150px;
    width: 200px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
    border: 1px solid black;
    -ms-transform: scale(0.5,0.5); /* IE 9 */
    -webkit-transform: scale(0.5,0.5); /* Safari */
    transform: scale(0.5,0.5); /* Standard syntax */
}
```



CSS3 2D Transforms (skewX)

- skews an element along the X-axis by the given angle

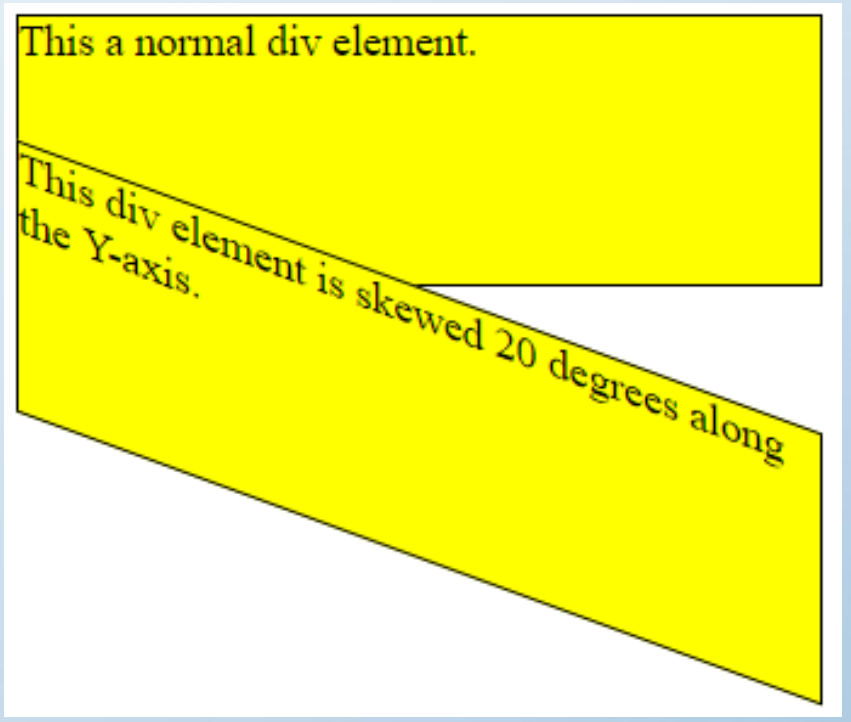
```css
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
}

div#myDiv {
    -ms-transform: skewX(20deg); /* IE 9 */
    -webkit-transform: skewX(20deg); /* Safari */
    transform: skewX(20deg); /* Standard syntax */
}
```



CSS3 2D Transforms (skewY)

- skews an element along the Y-axis by the given angle

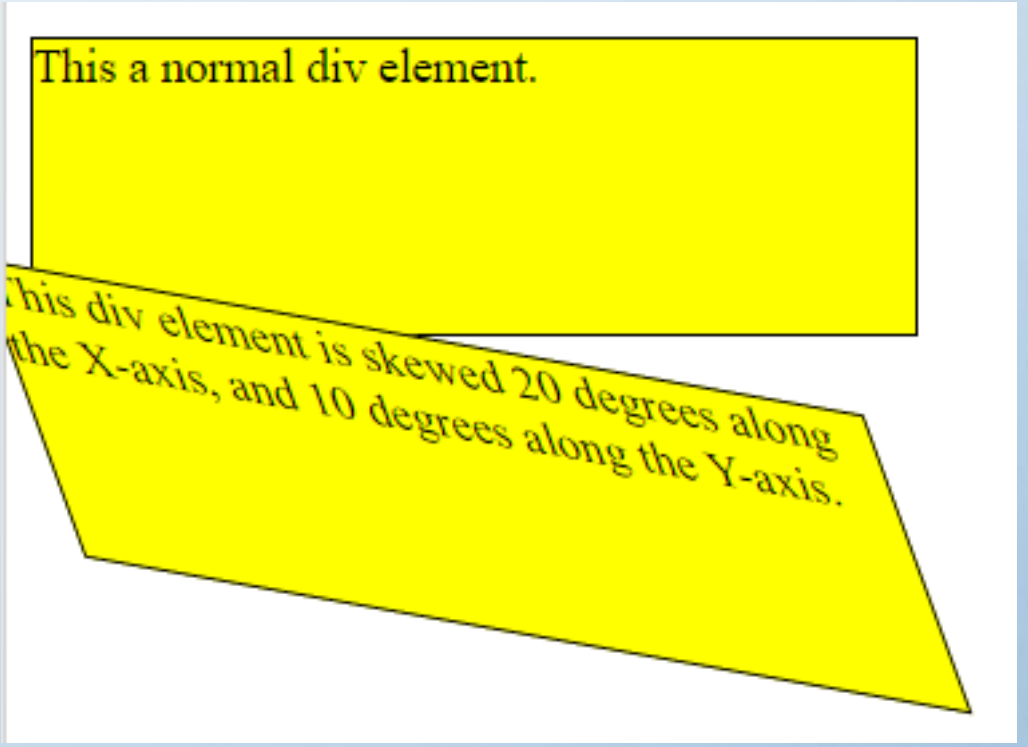
```css
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
}

div#myDiv {
    -ms-transform: skewY(20deg); /* IE 9 */
    -webkit-transform: skewY(20deg); /* Safari */
    transform: skewY(20deg); /* Standard syntax */
}
```



CSS3 2D Transforms (skew)

- The skew() method skews an element along the X and Y-axis by the given angles

```
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
}

div#myDiv {
    -ms-transform: skew(20deg,10deg); /* IE 9 */
    -webkit-transform: skew(20deg,10deg); /* Safari */
    transform: skew(20deg,10deg); /* Standard syntax */
}
```

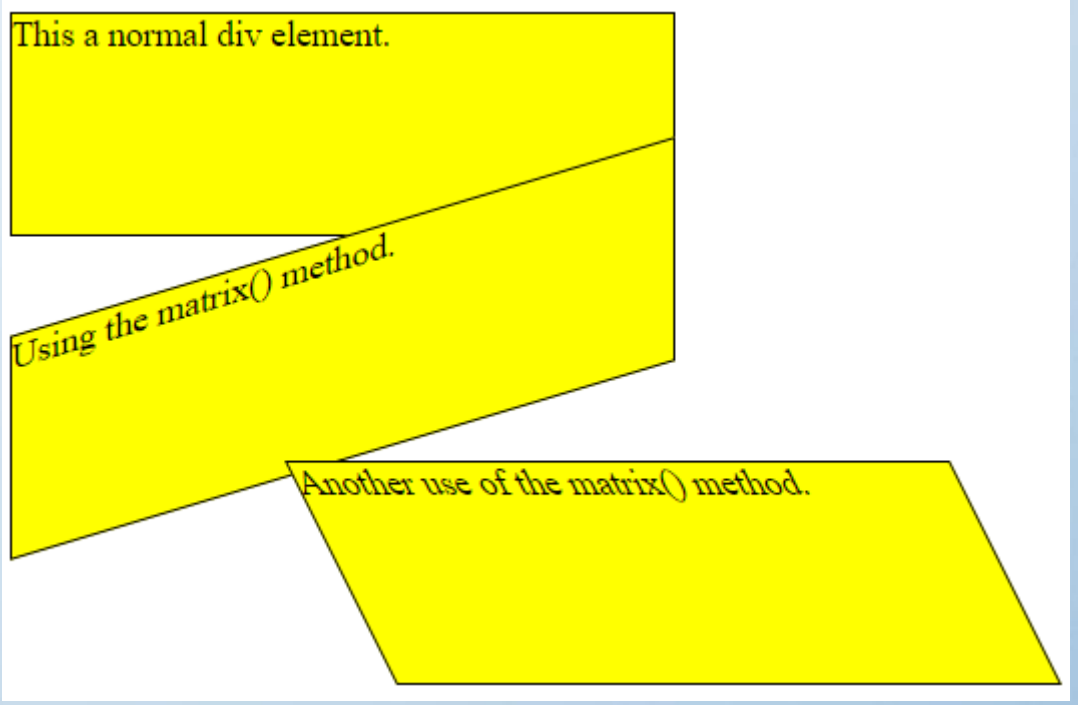


CSS3 2D Transforms (matrix)

- combines all the 2D transform methods into one
- takes six parameters, containing mathematic functions, which allows you to rotate, scale, move (translate), and skew elements
- The parameters are as follow: matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),rotate()):

CSS3 2D Transforms (matrix)

```css
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    border: 1px solid black;
}

div#myDiv1 {
    -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */
    -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /* Safari */
    transform: matrix(1, -0.3, 0, 1, 0, 0); /* Standard syntax */
}

div#myDiv2 {
    -ms-transform: matrix(1, 0, 0.5, 1, 150, 0); /* IE 9 */
    -webkit-transform: matrix(1, 0, 0.5, 1, 150, 0); /* Safari */
    transform: matrix(1, 0, 0.5, 1, 150, 0); /* Standard syntax */
}
```



CSS3 Transitions

- allows you to change property values smoothly (from one value to another), over a given duration.
- To create a transition effect, you must specify two things
  - the CSS property you want to add an effect to
  - the duration of the effect
- **Note:** If the duration part is not specified, the transition will have no effect, because the default value is 0

CSS3 Transitions

```css
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s; /* For Safari 3.1 to 6.0 */
    transition: width 2s;
}

div:hover {
    width: 300px;
}
```

CSS3 Transitions

- The following example adds a transition effect for both the width and height property, with a duration of 2 seconds for the width and 4 seconds for the height:

```
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s, height 4s; /* For Safari 3.1 to 6.0
*/
    transition: width 2s, height 4s;
}

div:hover {
    width: 300px;
    height: 300px;
}
```

CSS3 Transitions

- Specify the Speed Curve of the Transition
- The transition-timing-function property specifies the speed curve of the transition effect
  - ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
  - linear - specifies a transition effect with the same speed from start to end
  - ease-in - specifies a transition effect with a slow start
  - ease-out - specifies a transition effect with a slow end
  - ease-in-out - specifies a transition effect with a slow start and end
  - cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

CSS3 Transitions

- Delay the transition effect
- The transition-delay property specifies a delay (in seconds) for the transition effect

```css
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 3s; /* Safari */
    -webkit-transition-delay: 1s; /* Safari */
    transition: width 3s;
    transition-delay: 1s;
}
```

CSS3 Transitions

- Transition + Transformation
- The following example also adds a transformation to the transition effect

```
div {
    width: 300px;
    height: 300px;
    background: red;
    -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /*
Safari */
    transition: width 2s, height 2s, transform 2s;
}

div:hover {
    width: 350px;
    height: 350px;
    -webkit-transform: rotate(180deg); /* Safari */
    transform: rotate(180deg);
}
```

CSS3 Animations

- CSS3 animations allows animation of most HTML elements without using JavaScript or Flash!
- An animation lets an element gradually change from one style to another.
- You can change as many CSS properties you want, as many times you want.
- To use CSS3 animation, you must first specify some keyframes for the animation.
- Keyframes hold what styles the element will have at certain times
- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
- To get an animation to work, you must bind the animation to an element.

CSS3 Animations

- The following example binds the "example" animation to the element. The animation will last for 4 seconds, and it will gradually change the background-color of the element from "red" to "yellow"

```css
div {
    width: 100px;
    height: 100px;
    background-color: red;
    -webkit-animation-name: example; /* Safari 4.0 - 8.0 */
    -webkit-animation-duration: 4s; /* Safari 4.0 - 8.0 */
    animation-name: example;
    animation-duration: 4s;
}

/* Safari 4.0 - 8.0 */
@-webkit-keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}

/* Standard syntax */
@keyframes example {
    from {background-color: red;}
    to {background-color: yellow;}
}
```

CSS3 Animations

- The following example will change the background-color of the element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

```css
div {
    width: 100px;
    height: 100px;
    background-color: red;
    -webkit-animation-name: example; /* Safari 4.0 - 8.0 */
    -webkit-animation-duration: 4s; /* Safari 4.0 - 8.0 */
    animation-name: example;
    animation-duration: 4s;
}

/* Safari 4.0 - 8.0 */
@-webkit-keyframes example {
    0%   {background-color: red;}
    25%  {background-color: yellow;}
    50%  {background-color: blue;}
    100% {background-color: green;}
}

/* Standard syntax */
@keyframes example {
    0%   {background-color: red;}
    25%  {background-color: yellow;}
    50%  {background-color: blue;}
    100% {background-color: green;}
}
```

CSS3 Animations

- The following example will change both the background-color and the position of the element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete

```css
div {
    width: 100px;
    height: 100px;
    background-color: red;
    position: relative;
    -webkit-animation-name: example; /* Safari 4.0 - 8.0 */
    -webkit-animation-duration: 4s; /* Safari 4.0 - 8.0 */
    animation-name: example;
    animation-duration: 4s;
}

/* Standard syntax */
@keyframes example {
    0%   {background-color:red; left:0px; top:0px;}
    25%  {background-color:yellow; left:200px; top:0px;}
    50%  {background-color:blue; left:200px; top:200px;}
    75%  {background-color:green; left:0px; top:200px;}
    100% {background-color:red; left:0px; top:0px;}
}
```

CSS3 Animations

- The animation-delay property specifies a delay for the start of an animation
- The following example has a 2 seconds delay before starting the animation:

```
div {
    width: 100px;
    height: 100px;
    background-color: red;
    position: relative;
    -webkit-animation-name: example; /* Safari 4.0 - 8.0 */
    -webkit-animation-duration: 4s; /* Safari 4.0 - 8.0 */
    -webkit-animation-delay: 2s; /* Safari 4.0 - 8.0 */
    animation-name: example;
    animation-duration: 4s;
    animation-delay: 2s;
}

/* Standard syntax */
@keyframes example {
    0%   {background-color:red; left:0px; top:0px;}
    25%  {background-color:yellow; left:200px; top:0px;}
    50%  {background-color:blue; left:200px; top:200px;}
    75%  {background-color:green; left:0px; top:200px;}
    100% {background-color:red; left:0px; top:0px;}
}
```

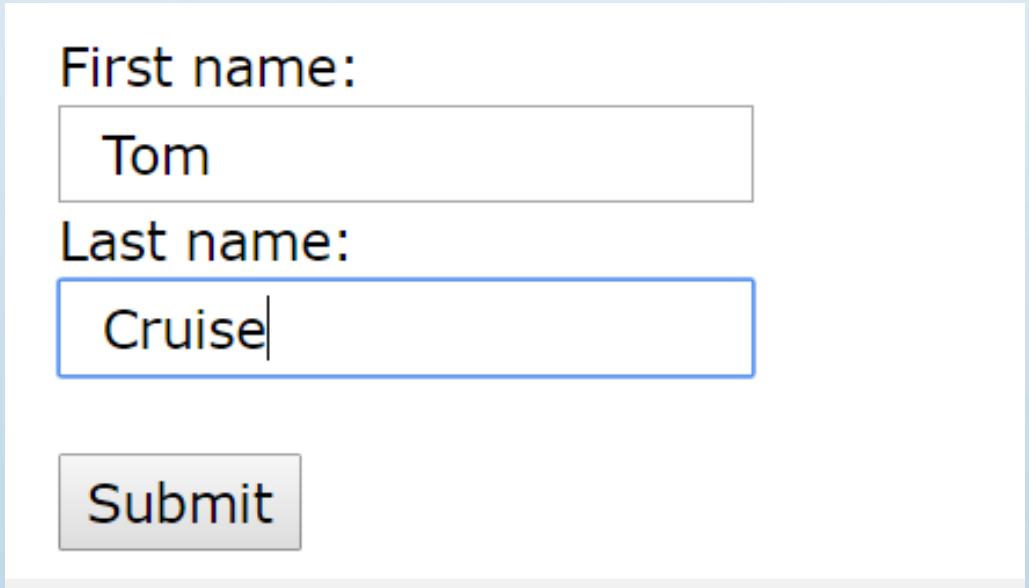
CSS3 Animations

- The animation-iteration-count property specifies the number of times an animation should run.
- The following example will run the animation 3 times before it stops:

```
div {
    width: 100px;
    height: 100px;
    background-color: red;
    position: relative;
    -webkit-animation-name: example; /* Safari 4.0 - 8.0 */
    -webkit-animation-duration: 4s; /* Safari 4.0 - 8.0 */
    -webkit-animation-iteration-count: 3; /* Safari 4.0 - 8.0 */
    animation-name: example;
    animation-duration: 4s;
    animation-iteration-count: 3;
}

/* Standard syntax */
@keyframes example {
    0%   {background-color:red; left:0px; top:0px;}
    25%  {background-color:yellow; left:200px; top:0px;}
    50%  {background-color:blue; left:200px; top:200px;}
    75%  {background-color:green; left:0px; top:200px;}
    100% {background-color:red; left:0px; top:0px;}
}
```

Introduction to Forms

- The HTML **<form>** element defines a form that is used to collect user input:
- Used to create a GUI that helps to get user information and sent to the server
- An HTML form contains **form elements**
- Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more



The <input> Element

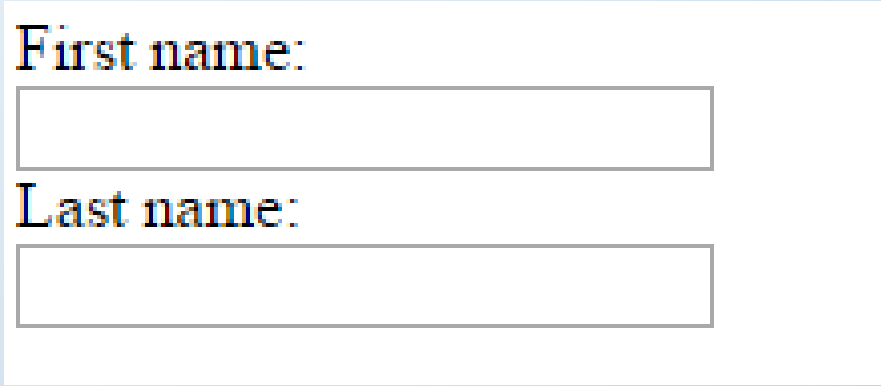
- The **<input>** element is the most important form element
- can be displayed in several ways, depending on the **type** attribute.

Type	Description
<input type="text">	Defines a one-line text input field
<input type="radio">	Defines a radio button (for selecting one of many choices)
<input type="submit">	Defines a submit button (for submitting the form)

Text Input

- **<input type="text">** defines a one-line input field for **text input**

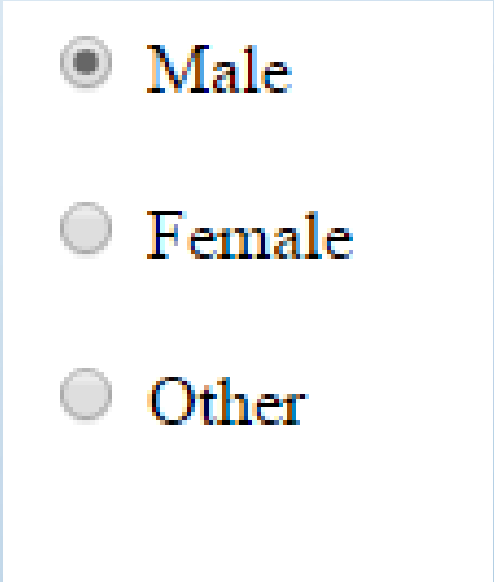
```
<form>
  First name:<br>
  <input type="text" name="firstname">
  <br>
  Last name:<br>
  <input type="text" name="lastname">
</form>
```



Radio Button Input

- **<input type="radio">** defines a **radio button**.
- Radio buttons let a user select ONE of a limited number of choices:

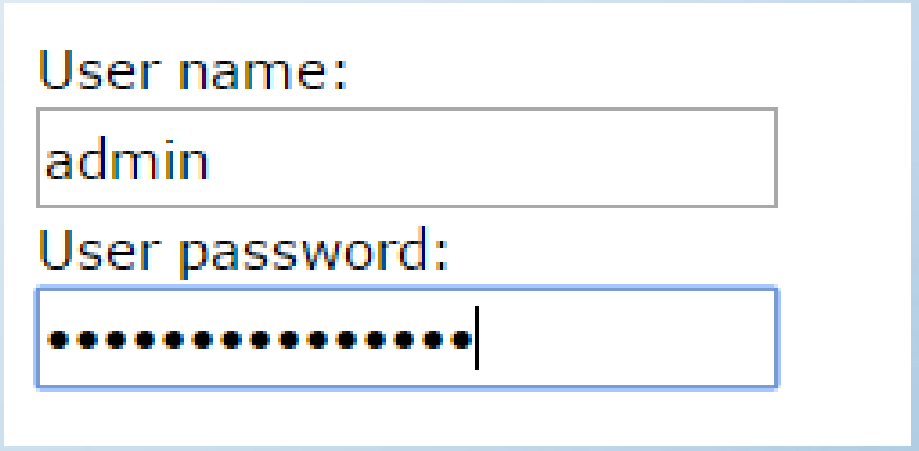
```
<form>

  <input type="radio" name="gender" value="male" checked> Male<br>

  <input type="radio" name="gender" value="female"> Female<br>

  <input type="radio" name="gender" value="other"> Other

</form>
```



Password Input

- **<input type="password">** defines a **password field**:

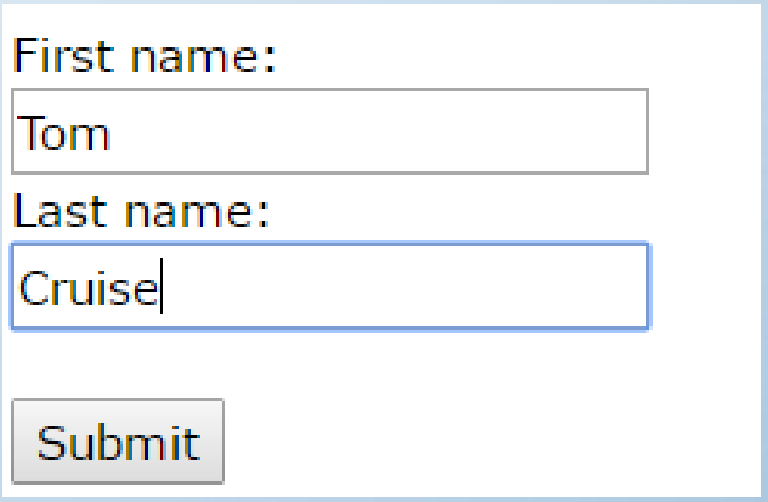
```
<form>
  User name:<br>
  <input type="text" name="username"><br>
  User password:<br>
  <input type="password" name="psw">
</form>
```



Input Type Submit

- **<input type="submit">** defines a button for **submitting** form data to a **form-handler**
- The form-handler is typically a server page with a script for processing input data
- The form-handler is specified in the form's **action** attribute

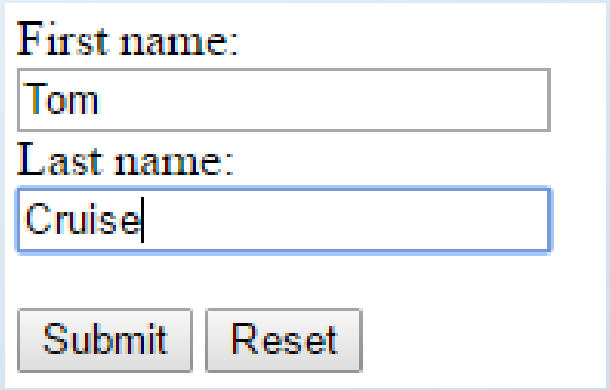
```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

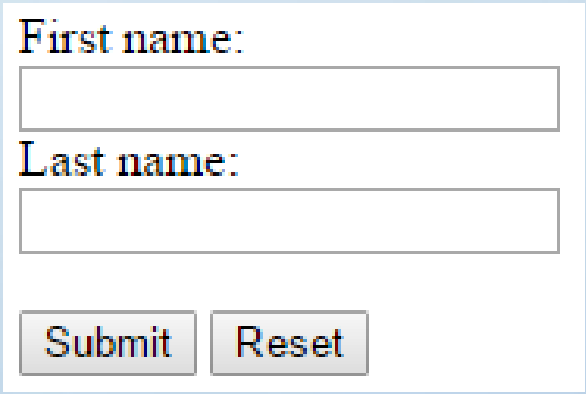


Input Type Reset

- **<input type="reset">** defines a **reset button** that will reset all form values to their default values

```
<form action="/action_page.php"
First name:<br>
<input type="text" name="firstn
<br>
Last name:<br>
<input type="text" name="lastna
<br><br>
<input type="submit" value="Sub
<input type="reset">
</form>
```

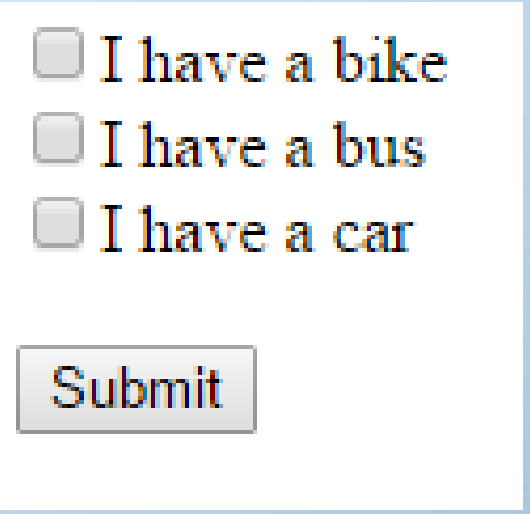




Input Type Checkbox

- **<input type="checkbox">** defines a **checkbox**
- Checkboxes let a user select ZERO or MORE options of a limited number of choices

```
<form action="/action_page.php">
  <input type="checkbox" name="vehicle1" value="Bike">I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Bus">I have a bus<br>
  <input type="checkbox" name="vehicle3" value="Car">I have a car <br><br>
  <input type="submit">
</form>
```



Input Type Button

- **<input type="button">** defines a **button**

```
<input type="button" onclick="alert('Hello World!')" value="Click Me!">
```

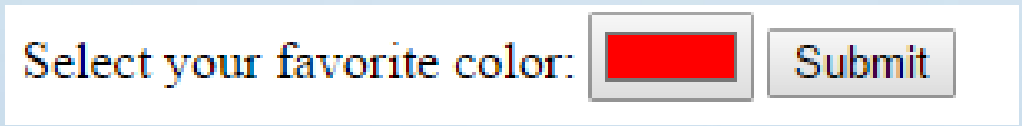
HTML5 Input Types

- HTML5 added several new input types:
  - color
  - date
  - datetime-local
  - email
  - month
  - number
  - range
  - search
  - tel
  - time
  - url
  - week

Input type Color

- The **<input type="color">** is used for input fields that should contain a color
- Depending on browser support, a color picker can show up in the input field

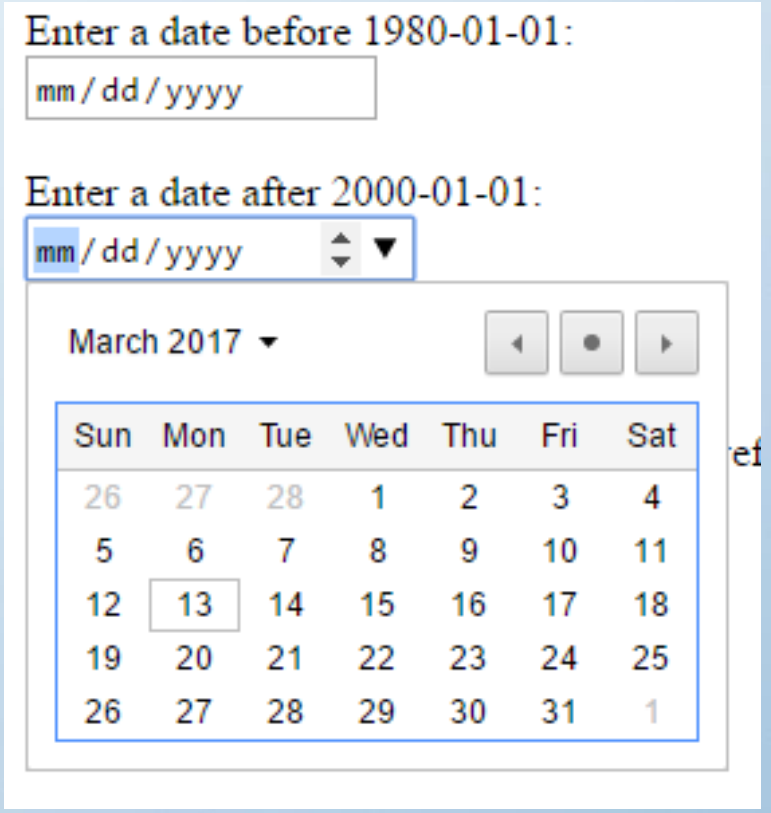
```
<form>
  Select your favorite color:
  <input type="color" name="favcolor">
</form>
```



Input type Date

- The **<input type="date">** is used for input fields that should contain a date
- Depending on browser support, a date picker can show up in the input field

```
<form action="/action_page.php">
  Enter a date before 1980-01-01:<br>
  <input type="date" name="bday" max="1979-12-31"><br><br>
  Enter a date after 2000-01-01:<br>
  <input type="date" name="bday" min="2000-01-02"><br><br>
  <input type="submit">
</form>
```



Input type Number

- The **<input type="number">** defines a **numeric** input field
- You can also set restrictions on what numbers are accepted.

```
<form action="/action_page.php">
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
  <input type="submit">
</form>
```

Numeric restrictions will apply in the input field:

Quantity (between 1 and 5): 5 Submit

Note: type="number" is not supported in IE9 and earlier.

Input type Time

- The **<input type="time">** allows the user to select a time.

```
<form>
  Select a time:
  <input type="time" name="usr_time">
</form>
```

Depending on browser support:
A time picker might pop-up when you enter the input field.

Select a time: 12:01 AM Submit

The Action Attribute

- The **action** attribute defines the action to be performed when the form is submitted.
- Normally, the form data is sent to a web page on the server when the user clicks on the submit button

```
<form action="/action_page.php">
```

The Method Attribute

- The **method** attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data
- The default method when submitting form data is GET.
- However, when GET is used, the submitted form data will be **visible in the page address field**

```
<form action="/action_page.php" method="get">
```

```
/action_page.php?firstname=Mickey&lastname=Mouse
```

The Method Attribute

- Always use POST if the form data contains sensitive or personal information
- The POST method does not display the submitted form data in the page address field
- POST has no size limitations, and can be used to send large amounts of data

The Name Attribute

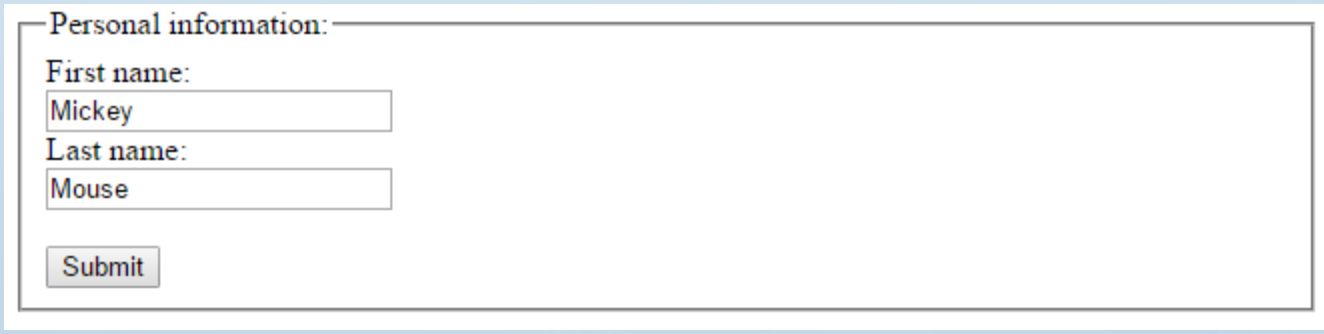
- Each input field must have a **name** attribute to be submitted.
- If the name attribute is omitted, the data of that input field will not be sent at all
- This example will only submit the "Last name" input field:

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

Grouping Form Data with <fieldset>

- The **<fieldset>** element is used to group related data in a form.
- The **<legend>** element defines a caption for the <fieldset> element.

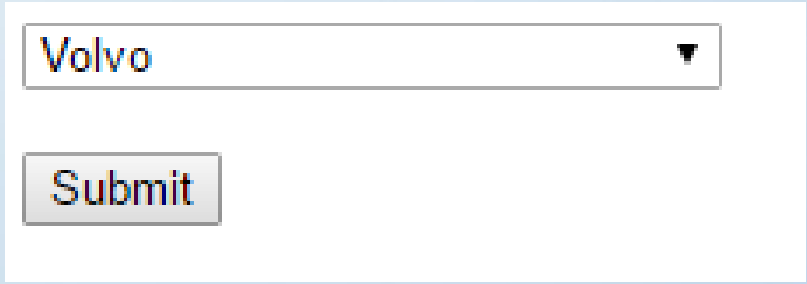
```
<form action="/action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey">
    <br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse">
    <br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```



The Select Element

- The **<select>** element defines a **drop-down list**:
- The **<option>** elements defines an option that can be selected.
- By default, the first item in the drop-down list is selected.
- To define a pre-selected option, add the **selected** attribute to the option:

```
<form action="/action_page.php">
  <select name="cars" style="width:200px">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select>
  <br><br>
  <input type="submit">
</form>
```



The Textarea Element

- The **<textarea>** element defines a multi-line input field (**a text area**):

```
<form action="/action_page.php">
  <textarea name="message" rows="7" cols="50">The cat was playing in the
garden.</textarea>
  <br>
  <input type="submit">
</form>
```