# CSIT884
# Web Development

Lecture 05B – CSS and JavaScript

Lecture slides are mainly taken from UOW

# Objectives

- Use JavaScript to manipulate CSS properties
- Produce animated interactive effects with CSS properties

# Recap: Change content by JavaScript

- **Step 1:** give the HTML element that we want to change an **ID**
- **Step 2:** use the function

```
var e = document.getElementById("the-id");
```

to get the HTML element that we want to change

- **Step 3:** change the content of the HTML element

for `span, div`, etc.:

```
e.innerHTML = "the-new-content";
```

for input text field:

```
e.value = "the-new-value";
```

for image:

```
e.src = "the-new-image-src";
```

# Change style by JavaScript

- **Step 1:** give the HTML element that we want to change an **ID**
- **Step 2:** use the function

```
var e = document.getElementById("the-id");
```

to get the HTML element that we want to change

- **Step 3:** change the style of the HTML element

```
e.style.[cssProperty] = "the-new-style-value";
```

for example:

```
e.style.color = "pink";
e.style.fontSize = "25px";
e.style.fontStyle = "italic";
...
```

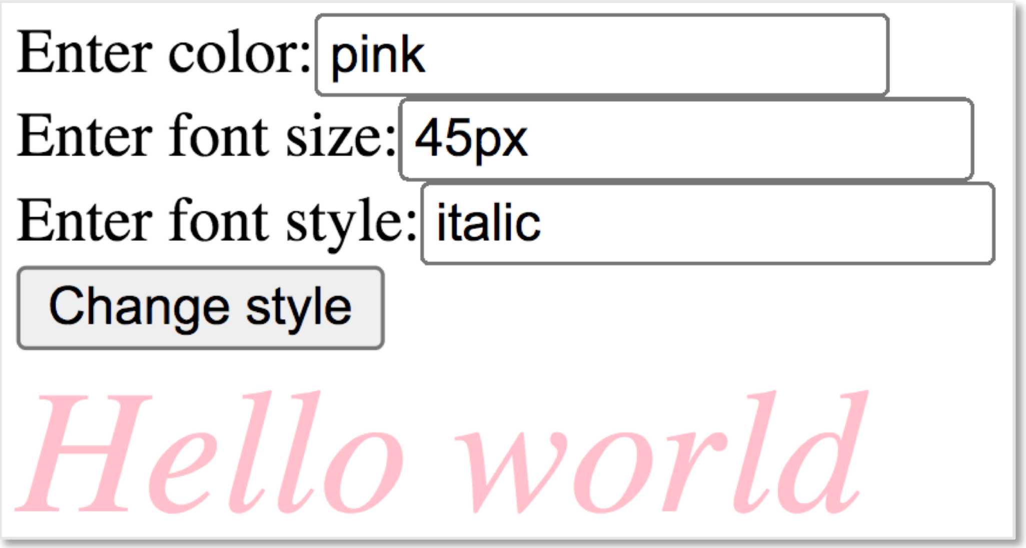
# Change style by JavaScript

- Original CSS properties are written in **hyphen convention**, when we translate them to JavaScript code, we need to use **camel case** convention:

| Original CSS property | Translate to JavaScript code |
|---|---|
| font-size | e.**style**.fontSize |
| font-style | e.**style**.fontStyle |
| background-color | e.**style**.backgroundColor |
| border-bottom-style | e.**style**.borderBottomStyle |
| border-left-color | e.**style**.borderLeftColor |
| padding-right | e.**style**.paddingRight |

# Example: change text style

The web page displays 3 text fields for user to enter color, font size and font style. When the user clicks the button, then we will change the style of the “Hello world” text.



Enter color: pink

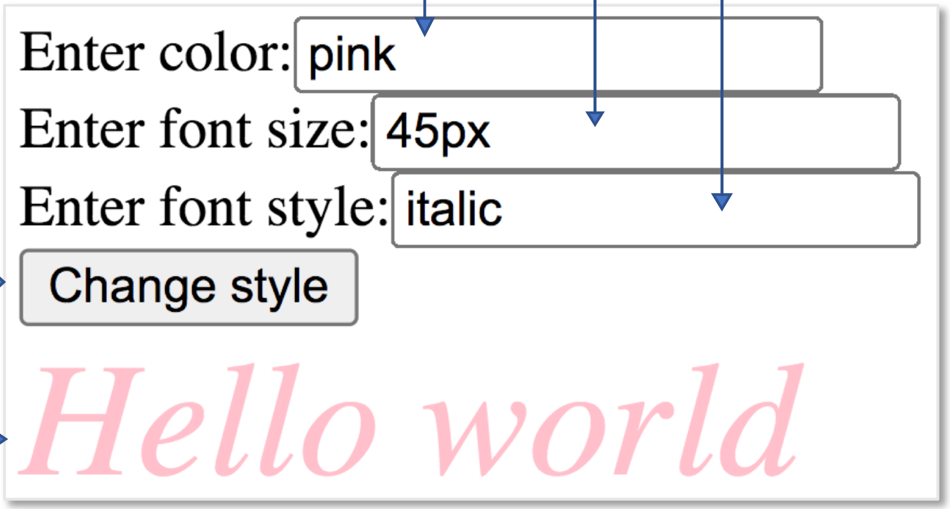
Enter font size: 45px

Enter font style: italic

*Hello world*

# Example: change text style

```
<input type="text" id="colorInput" />

<input type="text" id="fontSizeInput" />

<input type="text" id="fontStyleInput" />

<button onClick="changeStyle();">Change style</button>

<span id="hello">Hello world</span>
```



Enter color: pink

Enter font size: 45px

Enter font style: italic

*Hello world*

# Example: change text style

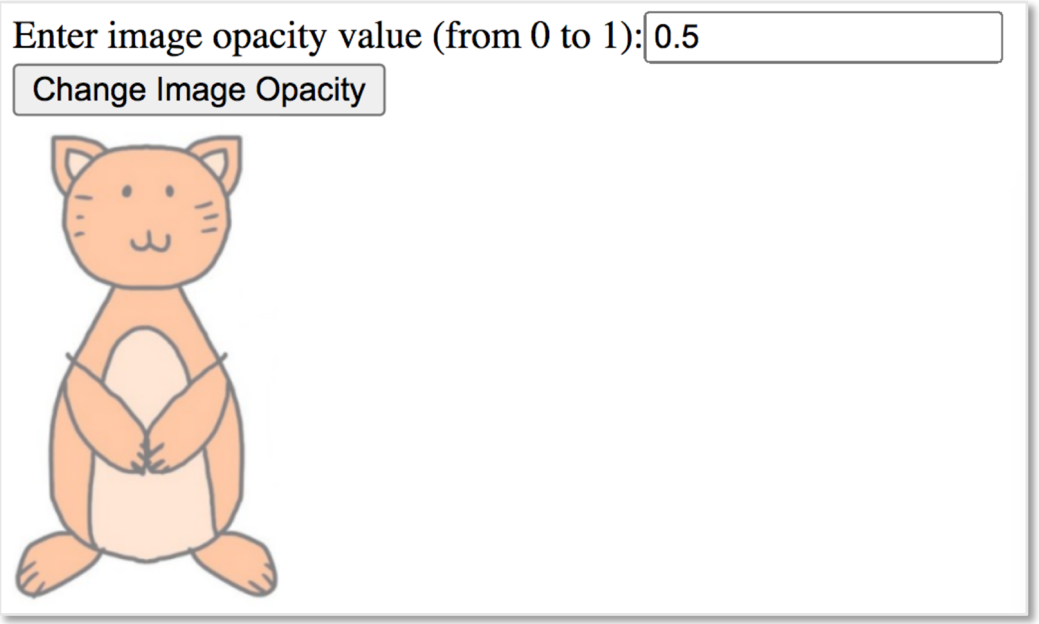
```
function changeStyle(){
 // get user input from text fields


 // get the Hello World span


 // change color, font size and font style


}
```

# Example: change text style

```
function changeStyle(){
 // get user input from text fields
 var colorTf = document.getElementById("colorInput");
 var colorValue = colorTf.value;
 var fontSizeTf = document.getElementById("fontSizeInput");
 var fontSizeValue = fontSizeTf.value;
 var fontStyleTf = document.getElementById("fontStyleInput");
 var fontStyleValue = fontStyleTf.value;


 // get the Hello World span
 var helloSpan = document.getElementById("hello");
 // change color, font size and font style
 helloSpan.style.color = colorValue;
 helloSpan.style.fontSize = fontSizeValue;
 helloSpan.style.fontStyle = fontStyleValue;
}
```

# Example: change image style

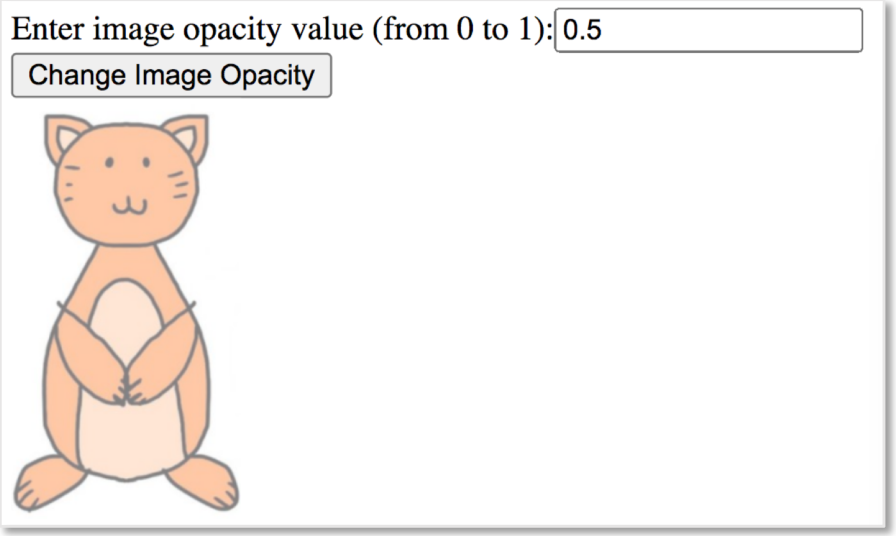
The web page displays a text field for user to enter image opacity value. When the user clicks the button, then we will change the image style.



Enter image opacity value (from 0 to 1): 0.5

# Example: change image style

```
<input type="text" id="opacity" />

<button onClick="changeOpacity()">

Change Image Opacity

</button>

<img id= "cat" src= "cat.png" />
```



Enter image opacity value (from 0 to 1): 0.5

# Example: change image style

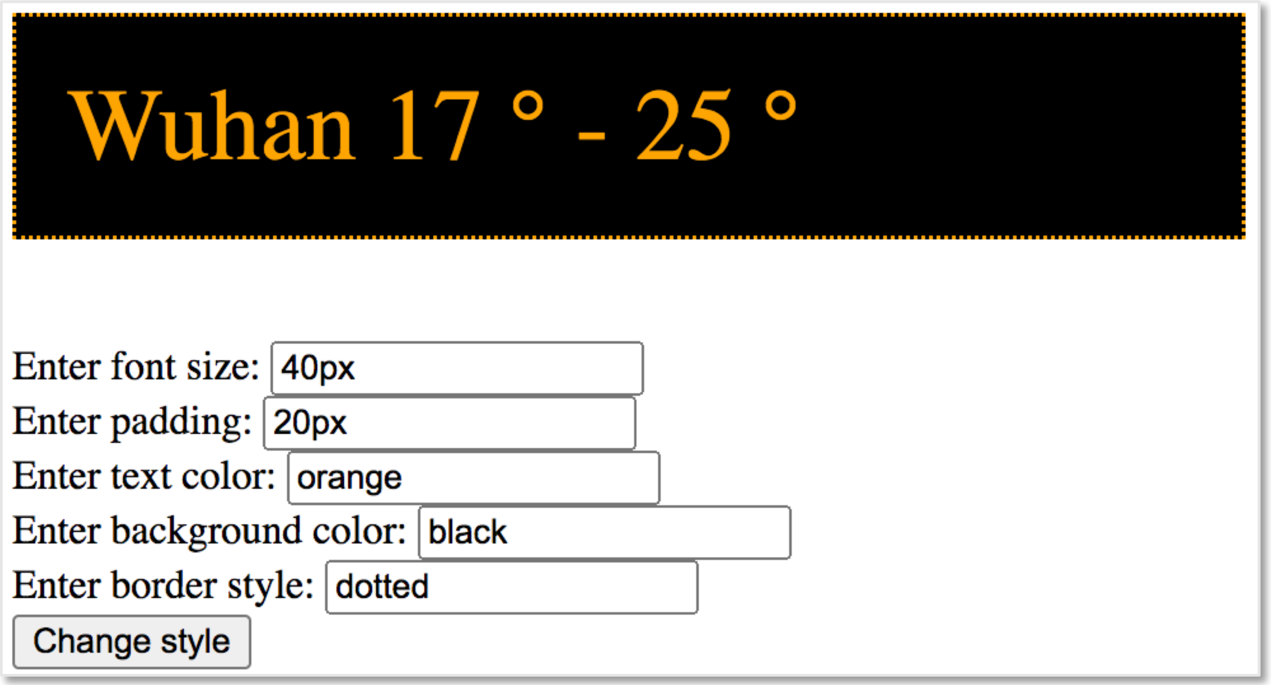
```
function changeOpacity(){
 // get the opacity value



 // set the image opacity



}
```

# Example: change image style

```
function changeOpacity(){
 // get the opacity value
 var opacityField = document.getElementById("opacity");
 var opacityValue = Number(opacityField.value);
 // set the image opacity
 var image = document.getElementById("cat");
 image.style.opacity = opacityValue;
}
```

# Example: change paragraph style

The web page displays text fields for user to enter font size, padding, text color, background color, border style. When the user clicks the button, then we will change the paragraph style.



Wuhan 17 ° - 25 °

Enter font size: 40px
Enter padding: 20px
Enter text color: orange
Enter background color: black
Enter border style: dotted

# Example: change paragraph style

```html
<p id="city">Wuhan 17 &deg; - 25 &deg;</p>

<input type="text" id="fontSizeInput" />

<input type="text" id="paddingInput" />

<input type="text" id="colorInput" />

<input type="text" id="bgInput" />

<input type="text" id="borderInput" />

<button onClick="changeStyle();">Change style</button>
```

# Example: change paragraph style

```
function changeStyle(){

 // get user input from text fields



 // change style



}
```

# Example: change paragraph style

```
function changeStyle(){
  // get user input from text fields
  var fontSize = document.getElementById("fontSizeInput").value;
  var padding = document.getElementById("paddingInput").value;
  var color = document.getElementById("colorInput").value;
  var bgColor = document.getElementById("bgInput").value;
  var border = document.getElementById("borderInput").value;

  // change style
  var cityPar = document.getElementById("city");
  cityPar.style.fontSize = fontSize;
  cityPar.style.padding = padding;
  cityPar.style.color = color;
  cityPar.style.backgroundColor = bgColor;
  cityPar.style.borderStyle = border;
}
```

# CSS Transitions

- Often called tweening
- Smooth out changes to property values between two states over time by filling in the frames in between
- CSS Transitions can enrich interfaces and improve usability
- Transitions require a beginning state and an end state
  - the end state needs to be triggered by a state change such as `:hover`, `:focus`, or `:active`

# CSS Transitions

- When applying a transition, one needs to specify:
  - **`transition-property`**: which CSS property to change (required)
  - **`transition-duration`**: duration of change (required)
  - **`transition-timing-function`**: how transition accelerates (ease, linear, ease-in, ease-out, ease-in-out)
  - **`transition-delay`**: is there a pause before it starts
- Shorthand `transition` property combines all of these properties into one declaration
  - **`transition: property duration timing-function delay;`**

# CSS Transitions

```
<style>
 .trans1 {
   text-align: center;
   padding: 1px;
   height: 25px;
   width: 200px;
   color: white;
   background-color: mediumblue;
   transition-property: background-color;
   transition-duration: 3s;
   transition-timing-function: ease-in-out;
   transition-delay: 1s;
 }
 .trans1:hover, .trans1:focus {
   background-color: red;
 }
```

```
. . .
<p class="trans1">Hello world!</p>
<p class="trans2">Shorten version!</p>
. . .
```

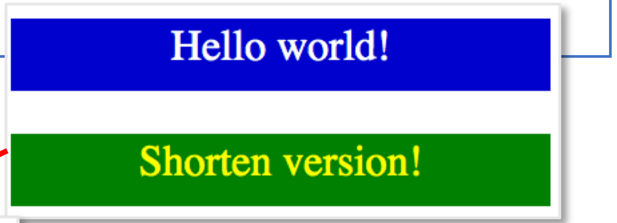
```
.trans2 {
   text-align: center;
   padding: 1px;
   height: 25px;
   width: 200px;
   color: yellow;
   background-color: green;
   transition: background-color 3s ease-in-out 1s;
 }
 .trans2:hover, .trans1:focus {
   background-color: blue;
 }
</style>
```

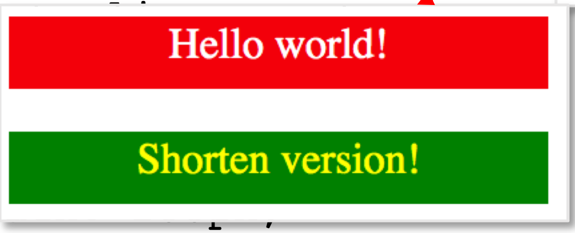
# CSS Transitions

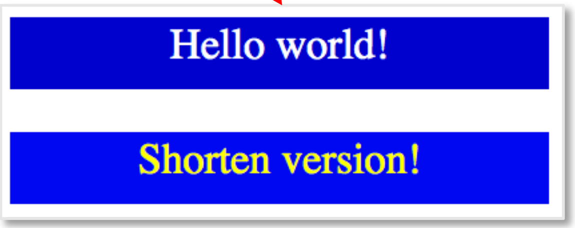
```
. . .
<p class="trans1">Hello world!</p>
<p class="trans2">Shorten version!</p>
. . .
```

```
<style>
 .trans1 {
   text-align: center;
   padding: 1px;
   height: 25px;
   width: 200px;
   color: white;
   background-color: mediumblue;
   transition-property: background-color;
   transition-duration: 3s;
   transition-timing-function: ease-in-out;
   transition-delay: 1s;
 }
 .trans1:hover, .trans1:focus {
   background-color: red;
 }
```

```
.trans2 {
   te
   pa
   he
   wi
   color: yellow;
   background-color: green;
   transition: background-color 3s ease-in-out 1s;
 }
 .trans2:hover, .trans1:focus {
   background-color: blue;
 }
</style>
```







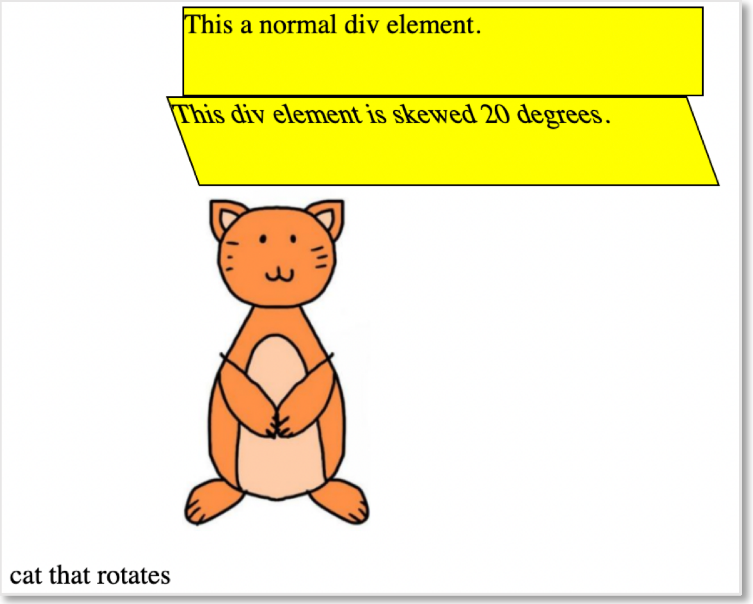
# CSS Transforms

- The CSS3 Transforms allows to rotate, relocate, resize, and skew HTML elements in two- and three-dimensional space
- Transforms can be applied
  - to the normal state of an element (so it appears in its transformed state when the page loads), or
  - when users interact with the element (for example via `:hover` or a JavaScript event)
- The two-dimensional transform functions include: `rotate()`, `translate()`, `scale()`, and `skew()`

# CSS Transforms

```
<div>
This a normal div element.
</div>

<div id="myDiv">
This div element is skewed
20 degrees.
</div>

<img src="cat.png">
<p>cat that rotates</p>
```

```
<style>
 div {
  margin-left:100px;
  width: 300px;
  height: 50px;
  vertical-align: middle;
  background-color: yellow;
  border: 1px solid black;
 }
 div#myDiv {
  transform: skew(20deg);
 }
 div#myDiv:hover {
  transform: skew(20deg) scale(1.5);
 }
```



This a normal div element.

This div element is skewed 20 degrees.

cat that rotates

```
 img {
  margin-left: 100px;
  height = 100px;
  weight = 100px;
 }
 img:hover {
  transform: rotate(-50deg);
 }
</style>
```

# References

- Jennifer Niederst Robbins, Learning Web Design - A Beginner's guide to HTML, CSS, JavaScript and Web Graphics, 5th edition, O’Reilly Media, 2018.
- http://www.w3schools.com/js
- http://www.w3schools.com/css
- https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
- http://developer.mozilla.org/en-US/docs/Web/JavaScript
- https://developer.mozilla.org/enUS/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions