

Client-Side Web Development

Class 5.1

Today's Topics

- CSS Transition
- CSS Transform
- **Exercise:** Pseudo Judo

Announcements

Schedule Change

Work Period

Recording

CSS Zen Garden

Any Questions?

Pseudo-Classes

A ***pseudo-class*** is a keyword added to a selector that specifies a special state of the selected element.

A *pseudo-class* will inherit the CSS property of the elements normal state

There are many different *pseudo-classes*

Different elements can have different
pseudo-classes

```
/* any element */
div:hover {
  border: 3px solid red;
}

/* buttons and links */
button:active {
  border: 1px inset black;
}

/* inputs */
input:invalid {
  background-color: red;
}

/* checkboxes and radio buttons*/
input[type="checkbox"]:checked {
  background-color: green;
}
```

CSS Transition

CSS Transitions provide a way to control the speed in which CSS property changes.

**Instead of having property change
take effect immediately, you can
cause the change to take place over
a period of time.**

**There are 4 CSS Transition
properties**

```
// Defaults of transition properties
.button {
  transition-property: all;
  transition-duration: 0s;
  transition-timing-function: ease;
  transition-delay: 0s;
}
```

```
// Shorthand
.button {
  transition: all 0s ease 0s;
}
```

transition-property

**This property set which CSS
properties will be transitioned. The
default is `all`.**

Using **all** will affect every
transitional property the same

**It is possible to transition properties
of an element different by using a
comma to separate each transition**

```
// Affects all properties
.button {
  transition: all 1s ease 0s;
}
```

```
// Does NOT work!
.button {
  transition: color 1s;
  transition: width 2s;
}
```

```
// Proper Way
.button {
  transition: color 1s, width 2s;
}
```


Not All CSS properties are animatable.

transition-duration

Sets how long a property will take to transition from one state to the next. The default is 0s.

The transition-duration value
can be in *seconds* (s) or
***microseconds* (ms)**

transition-delay

Sets how long to wait before the transition should start. The default is 0s.

**The transition-delay value can
be in *seconds* (s) or
microseconds (ms)**

When using the shorthand, the first time value will always be duration and the second delay.

transition-timing-function

Sets the speed curve of the transition.

**There are several predefined
timing-function values.**

Timing Function Values

- ease (default)
- ease-in
- ease-out
- ease-in-out
- linear
- step-start
- step-end

Timing Function Functions

- `steps()`
- `cubic-bezier()`
- `frames()`

Examples

CSS Transform

The ***CSS Transform*** property lets you modify the coordinate space of the CSS visual formatting model.

With ***CSS Transform***, elements can be translated, rotated, scaled, and skewed.

translate()

**The `translate()` function moves
an element in a
X and Y direction**

```
// moves up and to the right
```

```
.box {  
  transform: translate(30px, -30px)  
}
```

```
// moves down and to the left
```

```
.box {  
  transform: translate(-30px, 30px)  
}
```

```
// does not move
```

```
.box {  
  transform: translate(0px, 0px)  
}
```

scale()

The `scale()` function will increase or decrease the size of the element by multiple provided

Using the `scale()` function on an element does **NOT** affect adjacent elements

```
// Increase the size in both directions
```

```
.box {  
  transform: scale(3)  
}
```

```
// Increase 2 in X direction
```

```
.box {  
  transform: scale(2,0)  
}
```

```
// Decrease the size
```

```
.box {  
  transform: scale(0.5)  
}
```


rotate()

The `rotate()` will rotate an element by the given angle on the Z-axis

Use `rotateX()` or `rotateY()` to rotate an element on the X and Y axis, respectively

**The angle value takes deg or turn
units**

```
/* Rotate clockwise half way around */
```

```
.box {  
  transform: rotate(180deg)  
}
```

```
/* Rotate counter-clockwise all around */
```

```
.box {  
  transform: rotate(-360deg)  
}
```

```
/* Rotate clockwise 3 times around */
```

```
.box {  
  transform: rotate(3turn)  
}
```

```
/* Flips on X axis */
```

```
.box {  
  transform: rotateX(3turn)  
}
```

```
/* Spins on Y axis */
```

```
.box {  
  transform: rotateY(3turn)  
}
```

`skew()`

The `skew()` function skews the element in X and Y direction by the provide angle

The value must be `deg` units

Examples

Exercise: Pseudo Judo

For next class...

- **Project:** CSS Zen Garden

Next Week...

- CSS Animation
- Animation Techniques & Best Practices