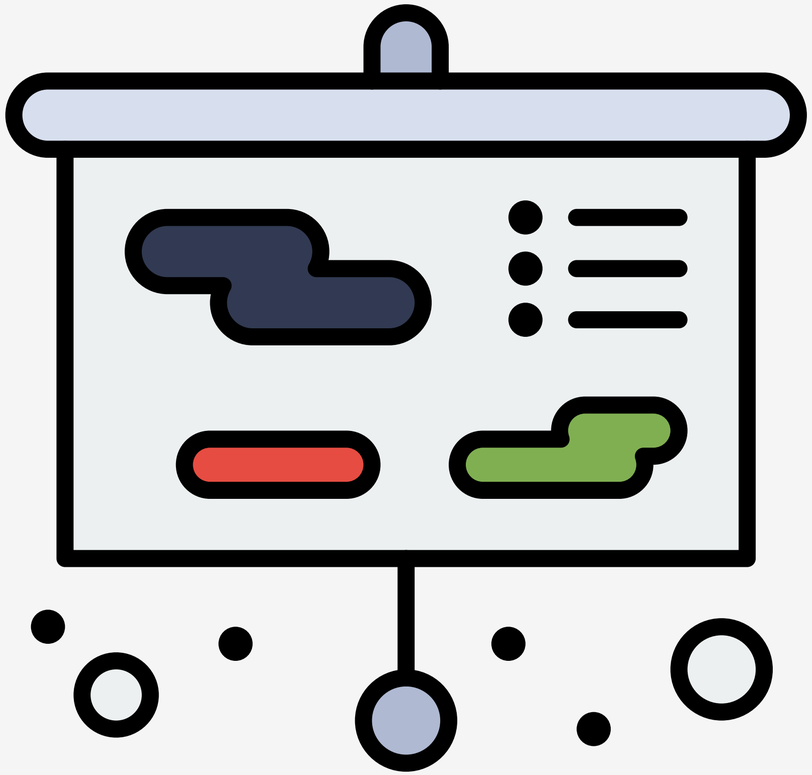
RESPONSIVE
WEB DESIGN II

Lecture 10

TODAY'S TOPICS



- CSS Transitions
- CSS Transforms
- **Participation:** Pseudo Judo

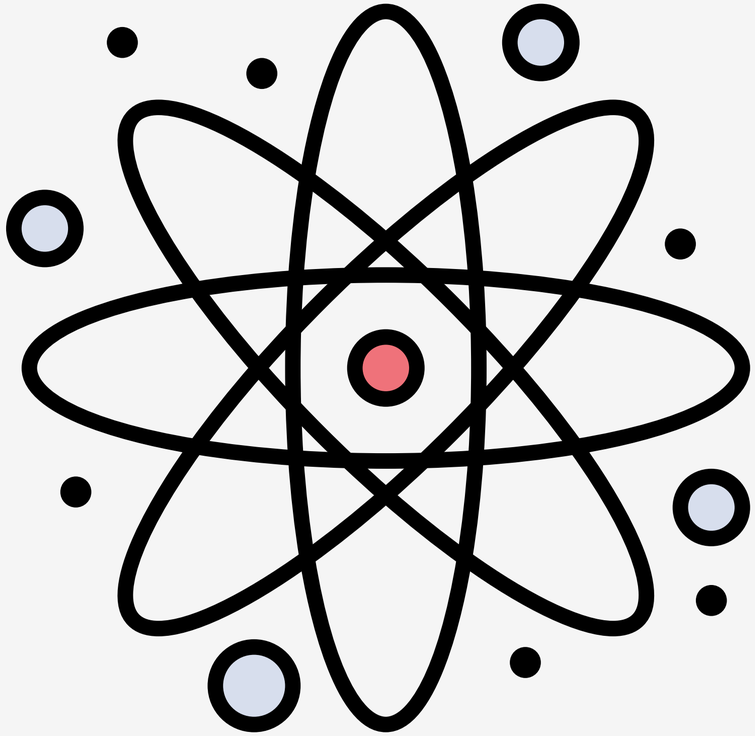
ANNOUNCEMENTS

- Sign-in Sheet

QUESTIONS

CSS TRANSITIONS

CSS TRANSITIONS

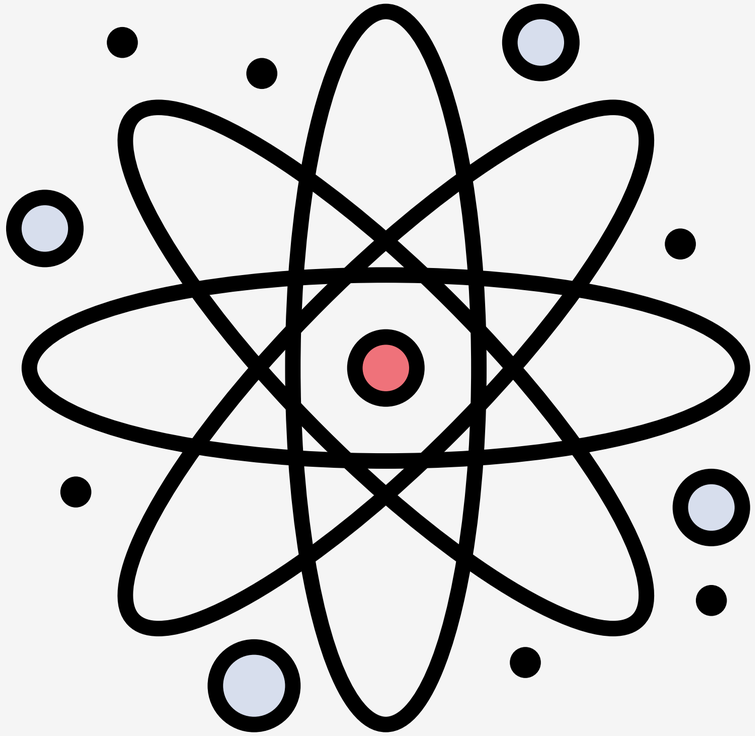


- When the appearance of an element changes (e.g. `:hover`), by default, the change happens instantly
- The transition properties can control how that change occurs
- There are four properties:
  - `transition-property`
  - `transition-duration`
  - `transition-delay`
  - `transition-timing-function`

CSS TRANSITIONS

```css
.box {
  transition-property: background;
  transition-duration: 0.3s;
  transition-timing-function: ease-out;
  transition-delay: 0.5s;
}

/* shorthand */
.box {
  transition: background 0.3s ease-out 0.5s;
}
```

CSS TRANSITIONS



- The same transition can be applied to all properties by using the `all` keyword
- Multiple different transitions can be applied by separating each transition statement with a comma

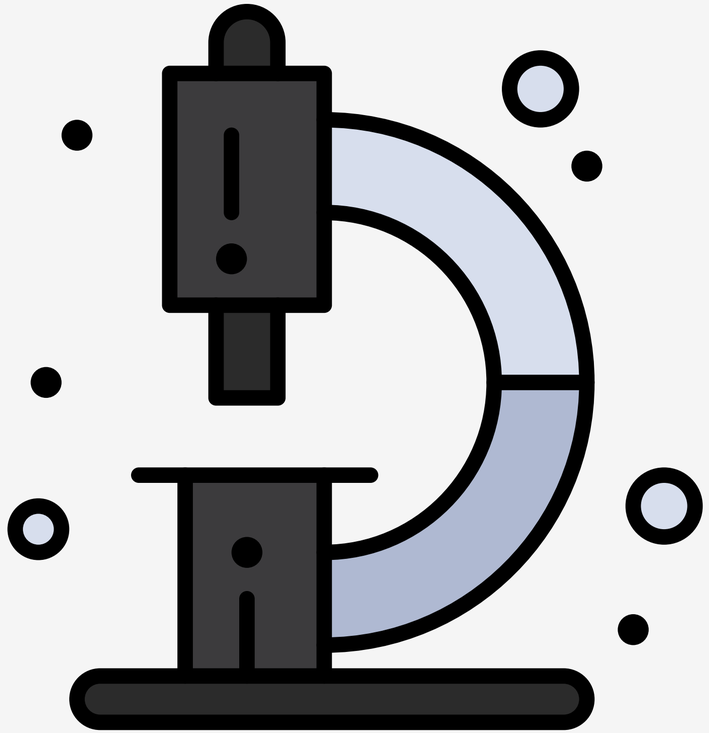
CSS TRANSITIONS

```css
/* transition is applied to all properties */
.box {
  transition: all 0.3s ease-out 0.5s;
}

/* transition both width and height */
.box {
  transition: width 0.3s 0.5s, height 0.5s 1s;
}
```

HANDS-ON

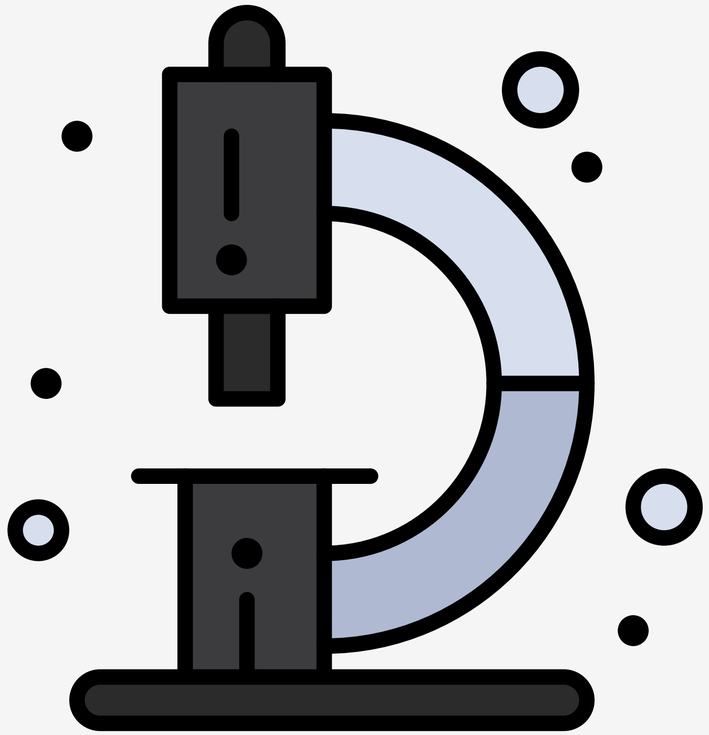
CSS TRANSFORM

CSS TRANSFORM



- The `transform` property is used to change the shape and / or position
- The `transform` property can perform 2D and 3D transformations
- Transformation are accomplished through transform functions
  - `translate()`
  - `rotate()`
  - `scale()`
  - `skew()`

TRANSFORM-ORIGIN

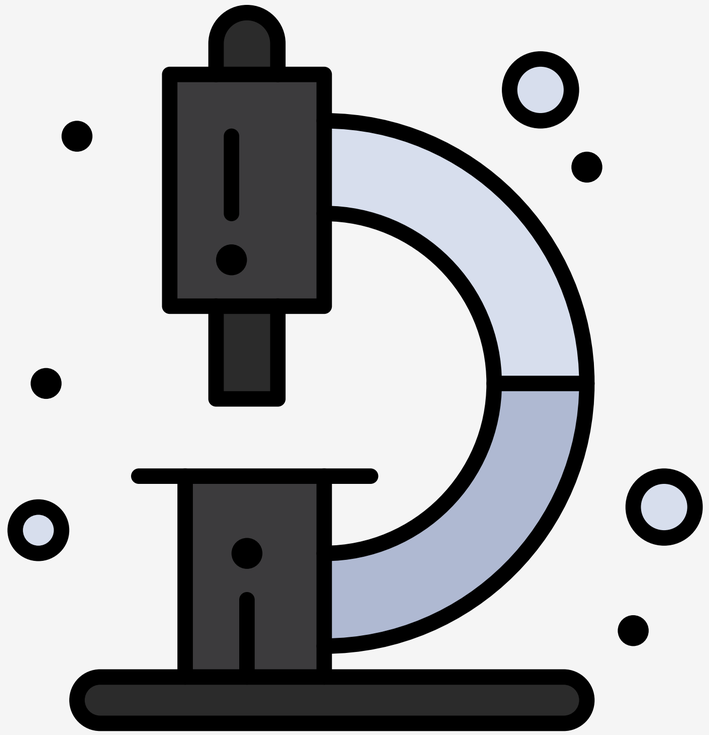


- The `transform-origin` property specifies the transformation point of an element
- The `transform-origin` property takes one or two values which to move the transformation point
- The transformation point can be outside an element
- The default for `transform-origin` is the center of the element

TRANSFORM ORIGIN

```css
/* default origin */
.box {
  transform-origin: center;
}

/* using keywords */
.box {
  transform-origin: top left;
}

/* using values */
.box {
  transform-origin: 100px -50px;
}
```

TRANSLATE

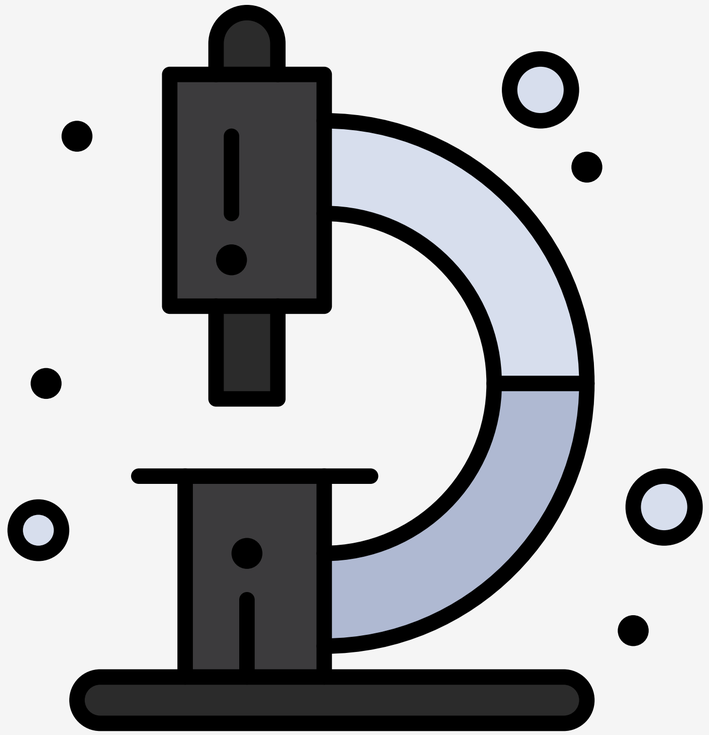


- The `translate()` function relocates an element along x and y axis.
- The `translate()` function takes one or two arguments, the distance the element will move
- The arguments can be positive or negative, and can be a length or a percentage

CSS TRANSITIONS

```css
/* the box moves down 100px and right 50px */
.box {
  transform: translate(50px, 100px)
}

/* the box moves up 50px and left 100px */
.box {
  transform: translate(-100px, -50px)
}
```

ROTATE



- The `rotate()` function defines how an element should rotate around transformation point
- The `rotate()` function takes an argument that specifies the angle at which the element will rotate
- The argument can be in degrees (`deg`), turns (`turn`), gradians (`grad`), or radians (`rad`)
- A positive value rotate clockwise and a negative value rotate counter-clockwise

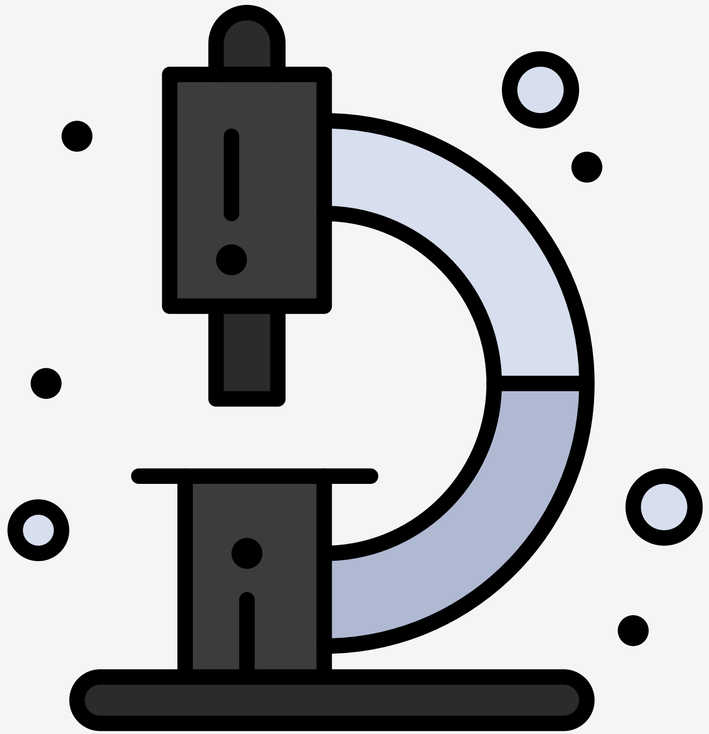
ROTATE

```css
/* one full circle in degrees */
.box {
  transform: rotate(360deg);
}

/* one full circle in turns */
.box {
  transform: rotate(1turn);
}

/* one full circle in gradians */
.box {
  transform: rotate(400grad);
}

/* one full circle in radians */
.box {
  transform: rotate(6.2832rad);
}
```

ROTATE

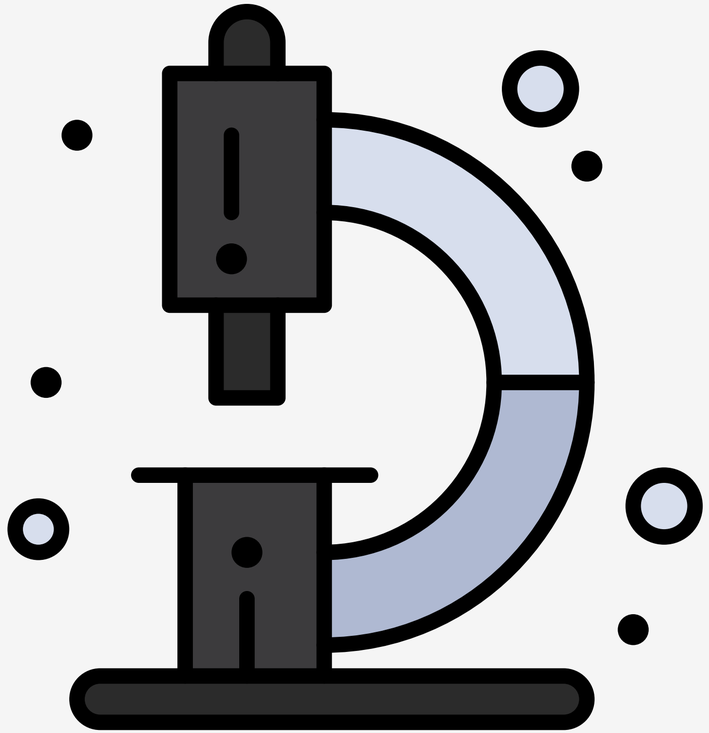
```css
/* rotates 1/4 circle to the right */
.box {
  transform: rotate(90deg);
}

/* rotates 1/4 circle to the left */
.box {
  transform: rotate(-0.25turn);
}
```

SCALE



- The `scale()` function resizes an element.
- The `scale()` function take one or two arguments.
- A value `1` represent the elements normal size
- A value greater than `1` the element will scale up, less than `1` the element will scale down

SCALE

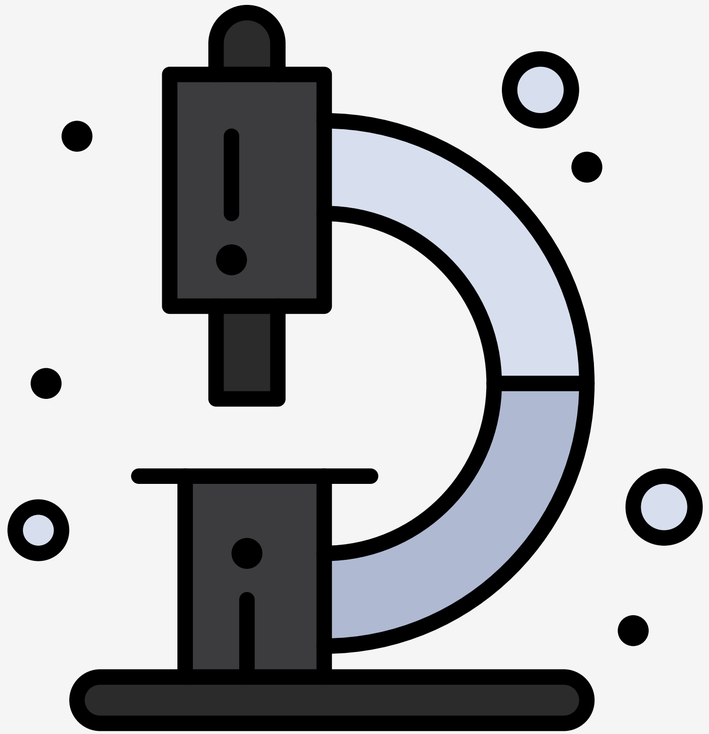


- Any content inside an element will also be scaled
- When element is scaled, it will have no affect on other elements

SCALE

```
/* scale up in both directions */
.box {
  transform: scale(2);
}

/* scale down in the X direction */
/* scale up in the Y direction */
.box {
  transform: scale(0.5, 2);
}
```

SKEW



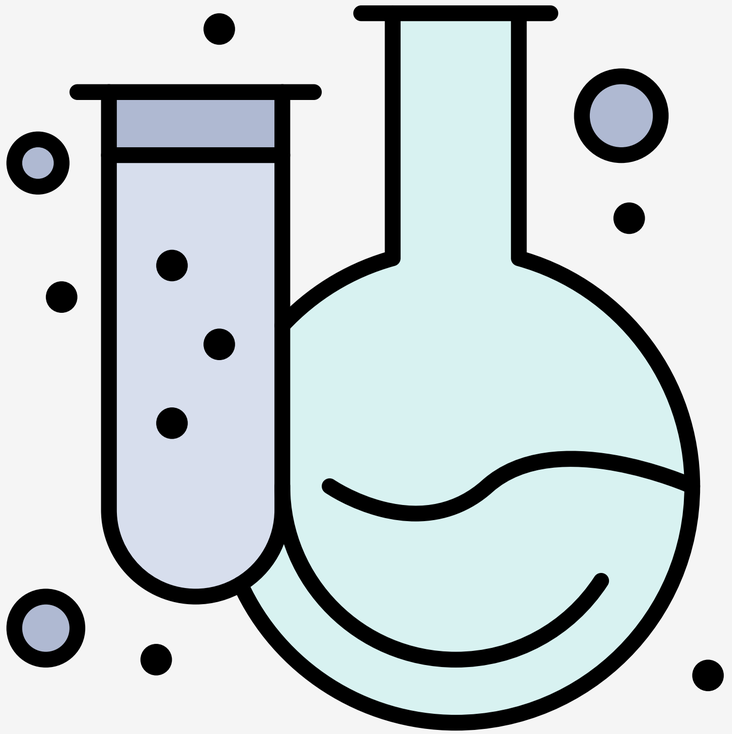
- The skew() function will skew an element by specified angle
- The skew() function takes one or two arguments
- The arguments can be in degrees (`deg`), turns (`turn`), gradians (`grad`), or radians (`rad`)
- The arguments can be positive or negative values
- Any content inside an element will also be skewed

SKEW

```css
/* leans to the left */
.box {
  transform: skew(15deg);
}

/* leans to the right */
.box {
  transform: skew(-0.06turn, 18deg);
}
```

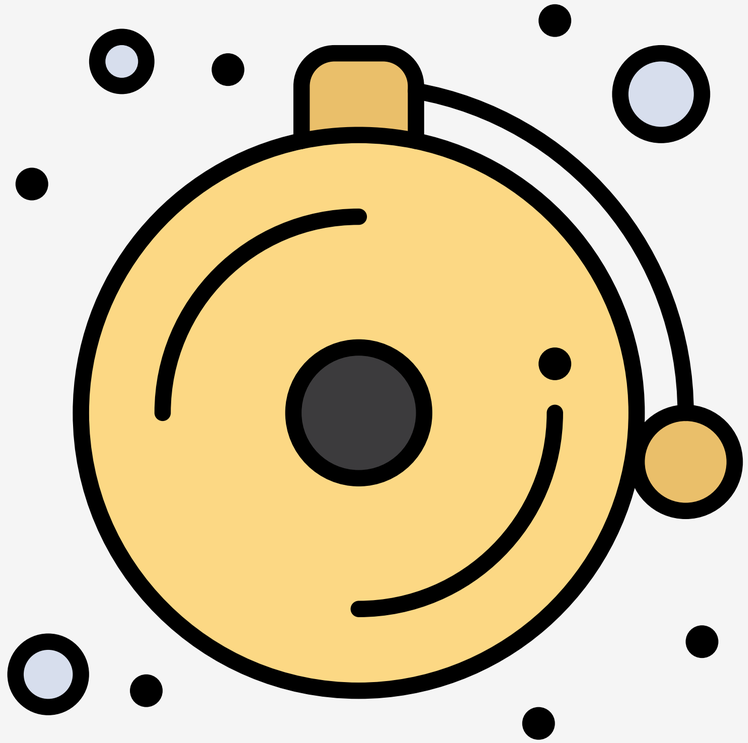
HANDS-ON

PSEUDO JUDO



- *FORK THE PEN!*
- Use the transition and transform properties to help the ninja do his moves
- Submit the URL to your pen
- *DUE:* **Tue. Feb. 11 @ 11:59 PM**

NEXT TIME...



- CSS Animation
- Participation: Doomsday Button