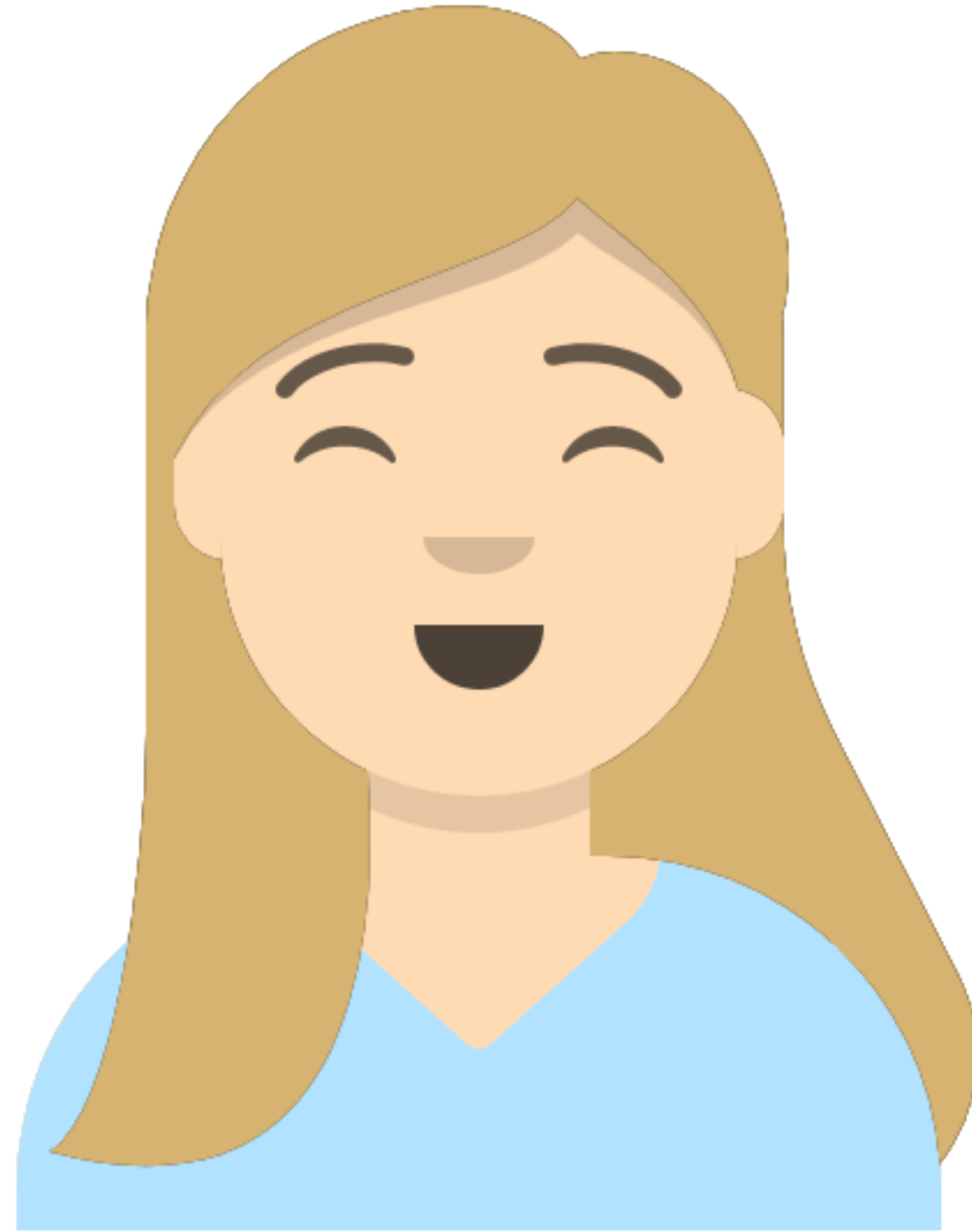




CSS Animations

Web, Mobile and Security
Frédéric Vlummens



Yup, me again

Overview

- CSS Positioning (different video)
- Responsive design techniques (different video)
- CSS Animation techniques

Overview

- CSS Positioning (different video)
- Responsive design techniques (different video)
- CSS Animation techniques
 - Transitions
 - Animations
 - Manipulating CSS transitions and animations in JS
 - Present and future

CSS transitions

changing a **css property value** from one to another in a smooth manner over a set period of time

transitions

required properties

1. transition-property

the CSS property you would like to animate (width, height, background-color, ...)

2. transition-duration

how long the transition should take (in s or ms)

optional properties

3. transition-delay

how long should the transition wait to take effect (in s or ms)

4. transition-timing-function

determines the curve speed for the transitioned effect (linear, ease-in, ...)

example

```
article {  
  width: 200px;  
  background-color: #FF0000;  
  transition-property: width;  
  transition-duration: 1s;  
}
```

```
article:hover {  
  width: 500px;  
}
```


example

```
article {  
  width: 200px;  
  background-color: #FF0000;  
  transition-property: width;  
  transition-duration: 1s;  
}
```

applied here

```
article:hover {  
  width: 500px;  
}
```

executed here

Example with more properties

```
article {  
  width: 200px;  
  background-color: #FF0000;  
  transition-property: width;  
  transition-duration: 1s;  
  transition-delay: 0.5s;  
  transition-timing-function: ease-out;  
}
```

```
article:hover {  
  width: 500px;  
}
```

shorthand

```
article {  
  width: 200px;  
  background-color: #FF0000;  
  transition: width 1s ease-out 0.5s;  
}
```

```
article:hover {  
  width: 500px;  
}
```

transitionable properties in CSS

<https://developer.mozilla.org/en-US/docs/Web/CSS/CSS animated properties>

Notice: display is **not** on the list!

transform functions

transform property

rotate skew scale translate matrix



```
img {}
```



```
img {  
  transform: rotate(45deg);  
}
```

transform property

rotate **skew** scale translate matrix



```
img {}
```



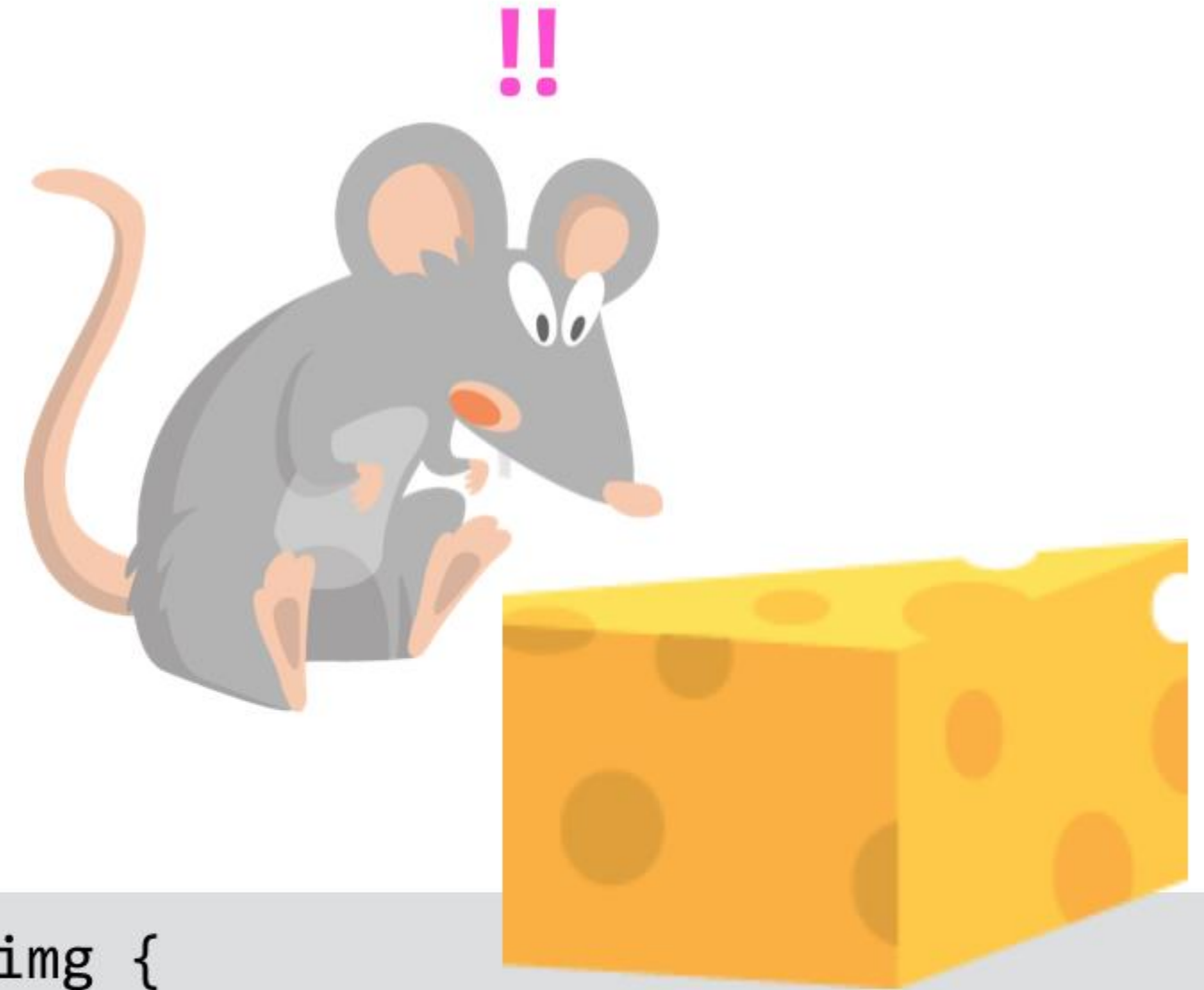
```
img {  
  transform: skew(20deg,20deg);  
}
```


transform property

rotate skew **scale** translate matrix



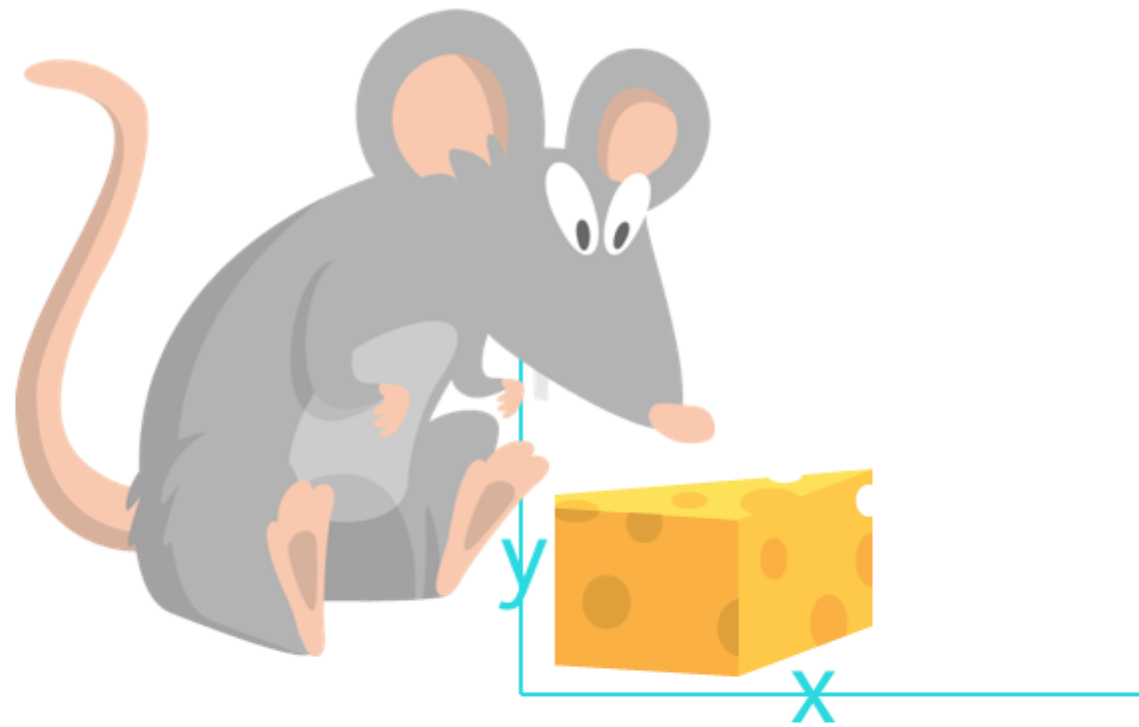
```
img {}
```



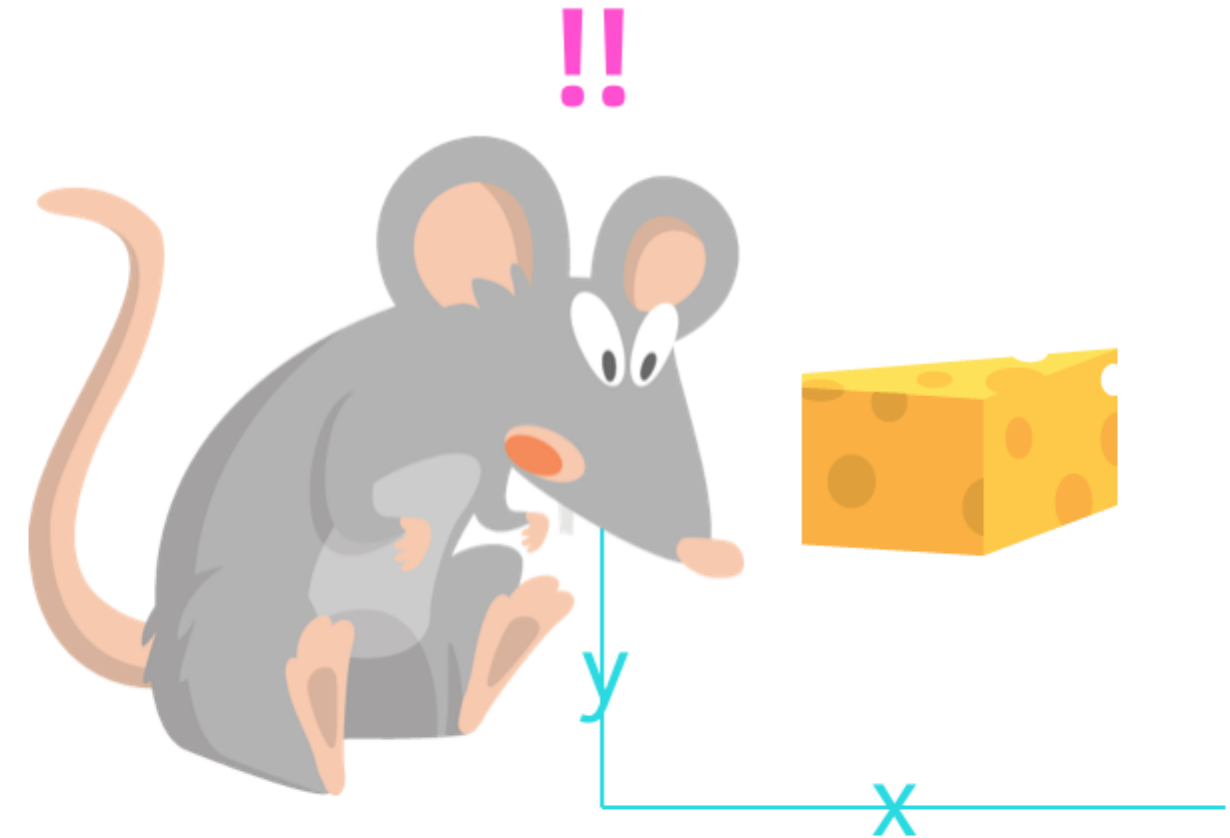
```
img {  
  transform: scale(2,2);  
}
```

transform property

rotate skew scale **translate** matrix



```
img {}
```



```
img {  
  transform: translate(30px,-100px);  
}
```

move on x,y coordinates

transform property

rotate skew scale translate **matrix**

```
img {  
  transform: rotate(15deg) translateX(230px) scale(1.5, 2.6) skew(220deg, -150deg)  
  translateX(230px)  
}
```

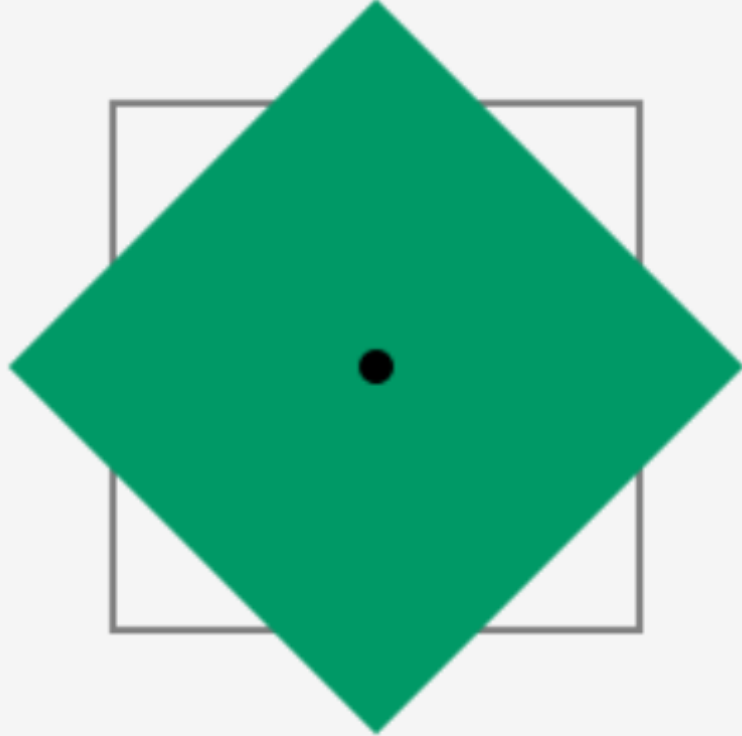
simplify



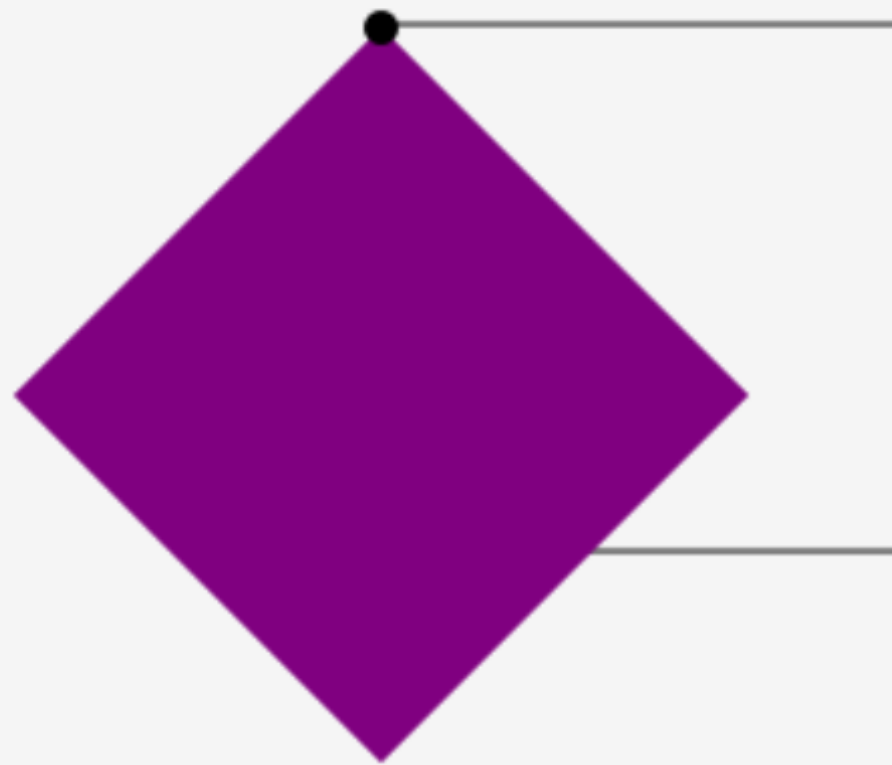
```
img {  
  transform: matrix(1.06, 1.84, 0.54, 2.8, 466px, 482px)  
}
```

Transform origin

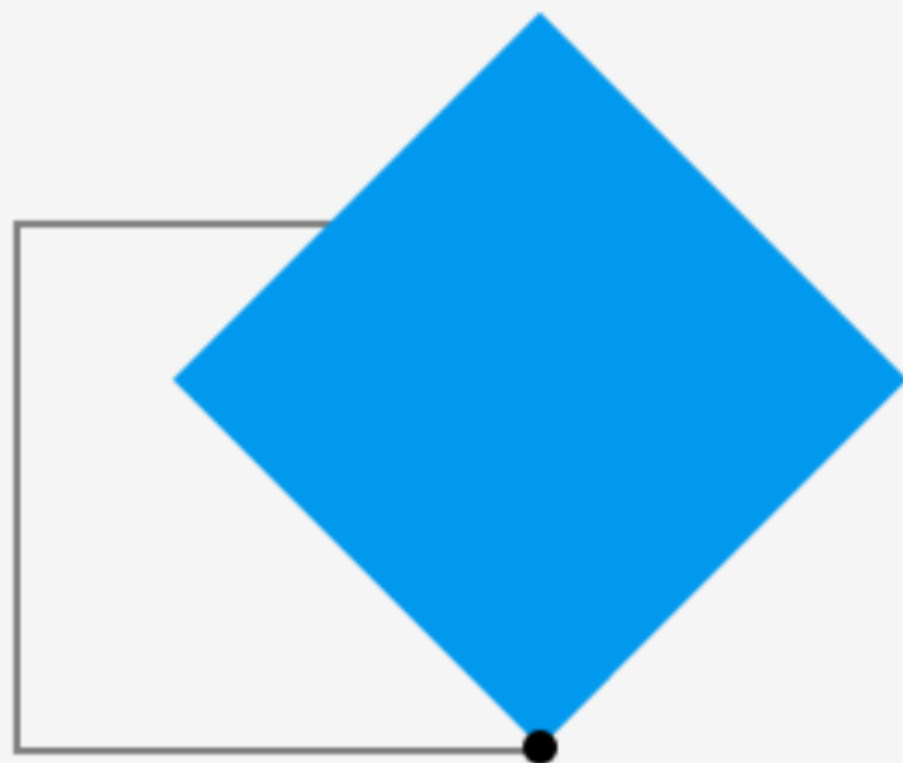
change origin point to be used for CSS
transform functions



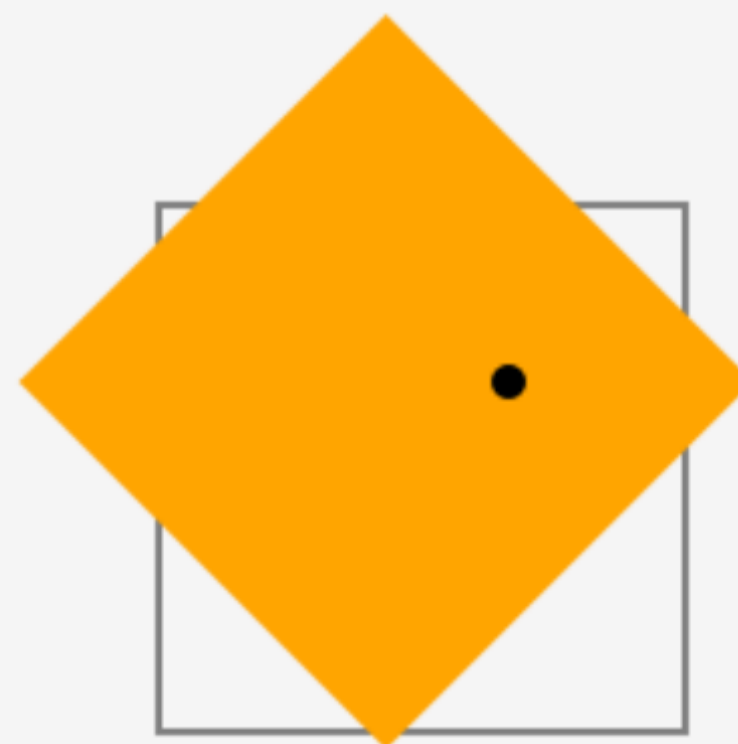
`transform-origin: 50% 50%;`



`0 0` or top left



`100% 100%`
or
bottom right



`80px 60px`

result of rotating an element using `transform: rotate(45deg)` with different `transform-origin` values.

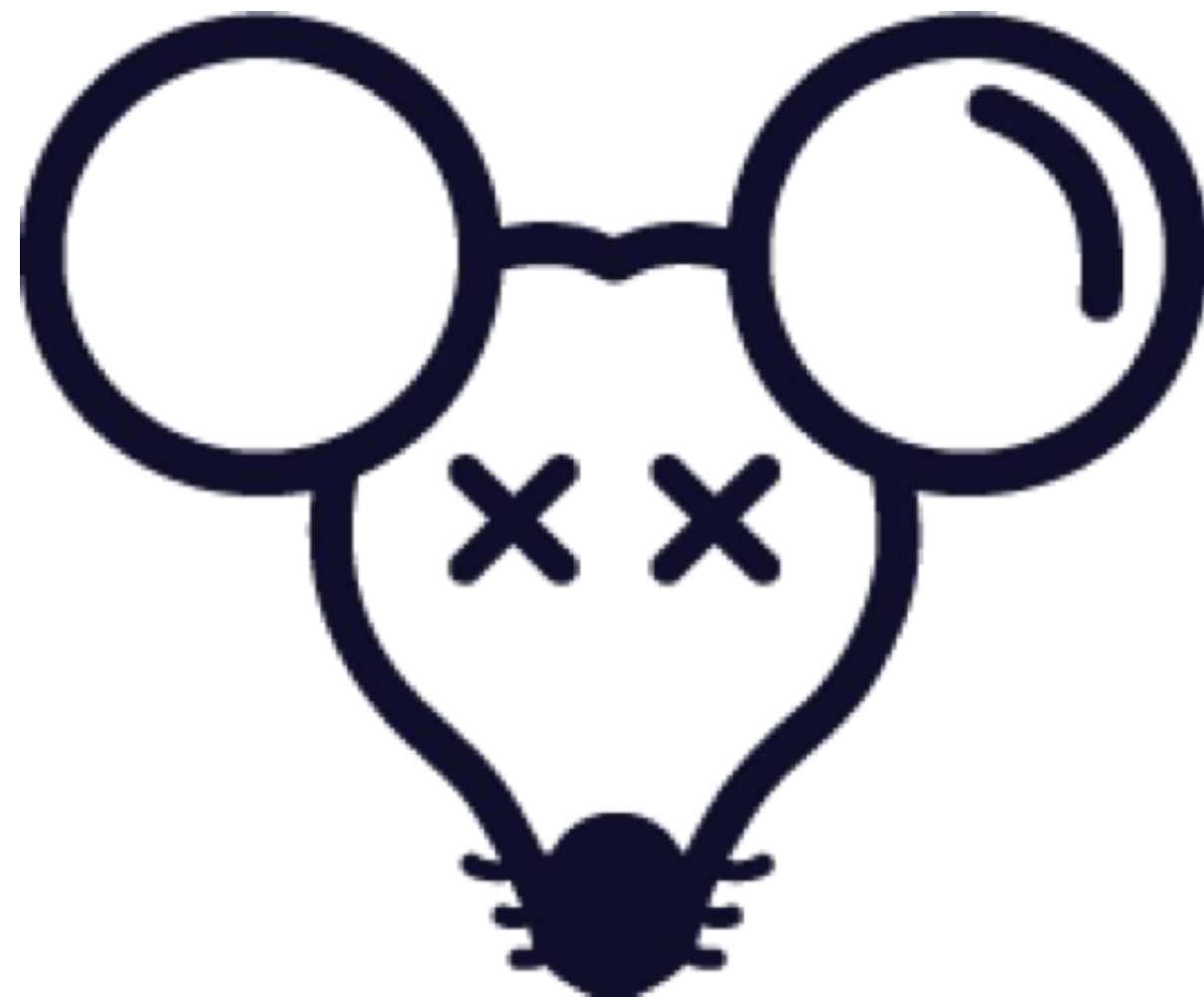
Transform 3D

change origin point to be used for CSS
transform functions

https://www.w3schools.com/css/css3_3dtransforms.asp

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function>

It's not working on my device



Vendor prefixes



-webkit-



-moz-



-o-



-ms-

CSS animations

difference to transitions

When you need more control over the flow of
the animation

options

The **animation** shorthand CSS property applies an animation between styles. It is a shorthand for `animation-name`, `animation-duration`, `animation-timing-function`, `animation-delay`, `animation-iteration-count`, `animation-direction`, `animation-fill-mode`, and `animation-play-state`.

Animations are twofold

- describe animation
- couple animation

```
@keyframes bounce {  
  0%, 20%, 50%, 80%, 100% { transform: translateY(0); }  
  40% { transform: translateY(-25px); }  
  60% { transform: translateY(-15px); }  
}  
  
.element {  
  animation-name: bounce;  
  animation-duration: 3s;  
}
```

```
@keyframes bounce {  
  0%, 20%, 50%, 80%, 100% { transform: translateY(0); }  
  40% { transform: translateY(-25px); }  
  60% { transform: translateY(-15px); }  
}
```

describe

```
.element {  
  animation-name: bounce;  
  animation-duration: 3s;  
}
```

couple

Think of an example

Manipulation CSS transitions with JS

We can detect when an animation or transition has ended in javascript

```
const transition = document.querySelector('.transition');  
  
transition.addEventListener('transitionend', () => {  
  console.log('Transition ended');  
});
```

And much more

```
const message = document.querySelector('.message');
const el = document.querySelector('.transition');

el.addEventListener('transitionrun', function() {
  message.textContent = 'transitionrun fired';
});

el.addEventListener('transitionstart', function() {
  message.textContent = 'transitionstart fired';
});

el.addEventListener('transitioncancel', function() {
  message.textContent = 'transitioncancel fired';
});

el.addEventListener('transitionend', function() {
  message.textContent = 'transitionend fired';
});
```

https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/transitionend_event

Same for animations!

Present and future

Getting Started with

GSAP

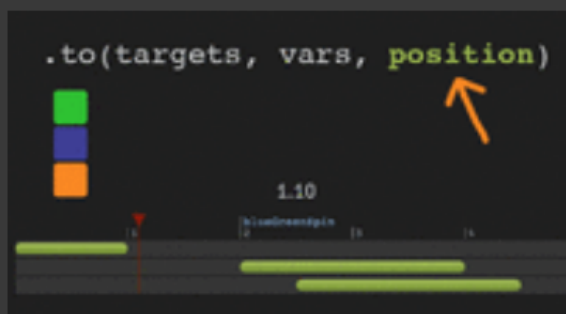
Animating with code may seem intimidating at first, but don't worry, our platform was engineered to make it simple and intuitive.

GET STARTED



Getting Started with GSAP

This is the best place to get started with GSAP. There are plenty of videos and demos that will get you animating in no time.



Timeline Tip: Understanding the Position Parameter

The secret to building gorgeous sequences with precise timing is understanding the super-flexible "position" parameter which controls the placement of your tweens, labels, callbacks, pauses, and even nested timelines.



Video: Sequence JavaScript Animations Like a Pro with GSAP's TimelineLite

This video walks you through some common problems that professional animators face every day and shows you how GSAP's TimelineLite tackles these challenges with ease.

The future: CSS motion path

https://blog.logrocket.com/css-motion-path-the-end-of-gsap/?utm_source=CSS-Weekly&utm_campaign=Issue-399&utm_medium=email#utm_source=CSS-Weekly&utm_campaign=Issue-399&utm_medium=email

For small stuff

<https://emilkowalski.github.io/css-effects-snippets/>