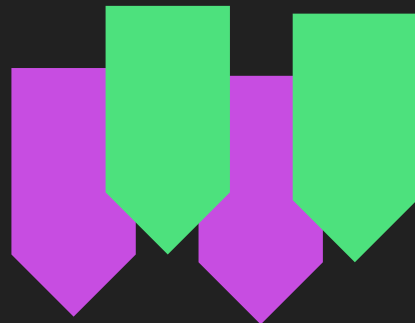


# DevX HTML/CSS

Class 13 - CSS Animations





# Review

We learned about *Advanced CSS*:

- Pseudo-classes

- :first-child, :last-child, :nth-child
- :hover

- Pseudo Elements

- :before
- :after

- Attribute Selectors


- [title], [href], [href\*="google"]

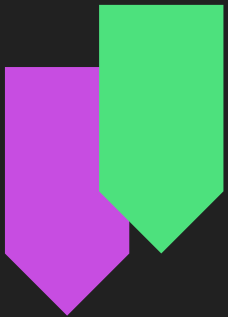
- calc()

- Shadows

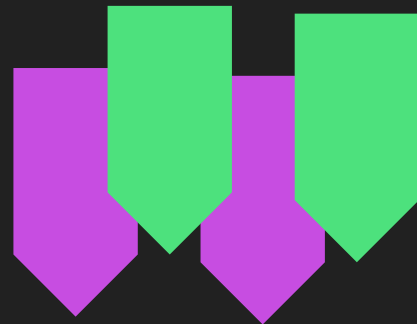
- Box-shadows
- Text-shadows

Questions?





# Role of Animations





# Role of Animations

Animations are a niche topic of HTML/CSS.

They can provide sites with a nice flair, feature, or UI experience.


Animations can:

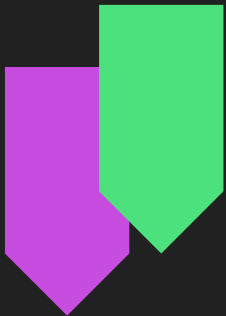
- Draw attention to a part of the page.
- Can help the user understand the relationship between two elements.
- Make layout changes feel smooth and pleasant for the user.

They require good knowledge of HTML and CSS. This makes them challenging to implement. But adding these small touches can help the user enjoy and understand the site better.

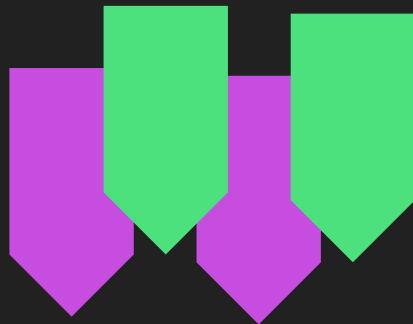
Animations can be implemented using CSS transition, CSS animation, and Javascript.

We'll cover only the first two in this class.





# Transitions





# Transitions

**Transition** is a CSS property.


If an element's properties change (height, width, color, etc), then it animates a transition between the earlier state and later state of the change.

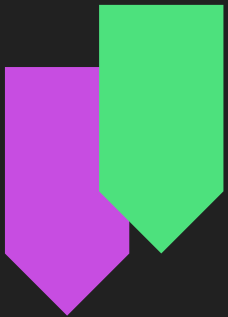
Eg: A box goes from width: 100px; to width: 200px. Transition will animate the growth from 100px to 200px.

Transition can take time values. Eg, "1s", ".5s", "10s". This indicates the speed of the transition.

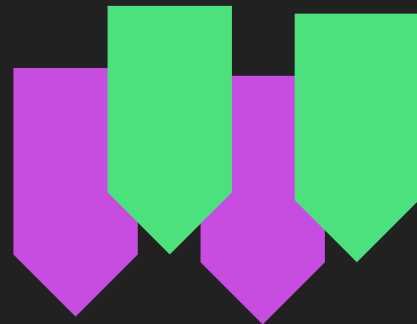
Note: You can limit the transition, to only apply to certain features, like color or size.

*Demonstrate!*



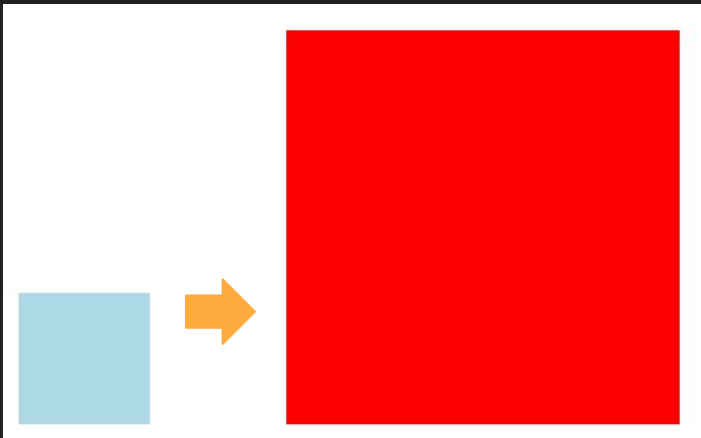


# Activity: Growing Box



# Activity: Growing Box

```
div.box {  
  height: 100px;  
  width: 100px;  
  background-color: lightblue;  
}  
div.box:hover {  
}  
  
<div class="box"></div>
```

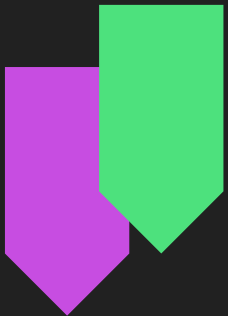


We're going to do a simple animation exercise.

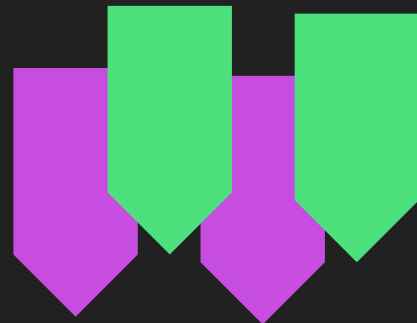
- 1 Make a new HTML file and add the HTML/CSS on the left.
- 2 Change the animation so:
  - The box takes on the color red on hover.
  - The box grows to 300px x 300px on hover.
  - The change is animated (use transition property)

If you have issues try googling your question!





# Animation And Keyframes



# Keyframes

Keyframes are a new CSS keyword for us.

Keyframes are a set of animation instructions. They change the properties of an element over the course of the animation, somewhat like transition.

Transition can only go back and forth, but keyframes can take any number of steps.

Keyframes can be “from” and “to”, or use %’s to make the animation.

```
@keyframes redToYellow2 {  
  from {  
    background-color: red;  
  }  
  to {  
    background-color: yellow;  
  }  
}
```

```
@keyframes redToYellow1 {  
  0% {  
    background-color: red;  
  }  
  25% {  
    background-color: green;  
  }  
  50% {  
    background-color: blue;  
  }  
  100% {  
    background-color: yellow;  
  }  
}
```

Demonstrate!



# Animation Property

**Animation** is a property. It uses **Keyframes**.

Note two properties: **animation-name**, and **animation-duration**.

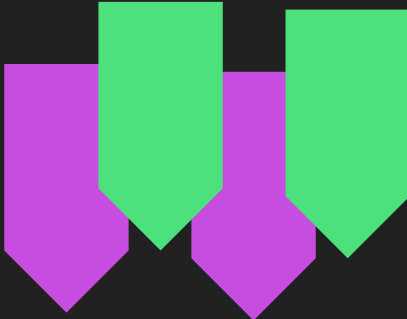
The value of **animation-name** must match a **@keyframes** name. It will perform the animation on the keyframe with the right name.

The **animation-duration** property instructs your browser on how long the animation should last.

CSS handles the rest!

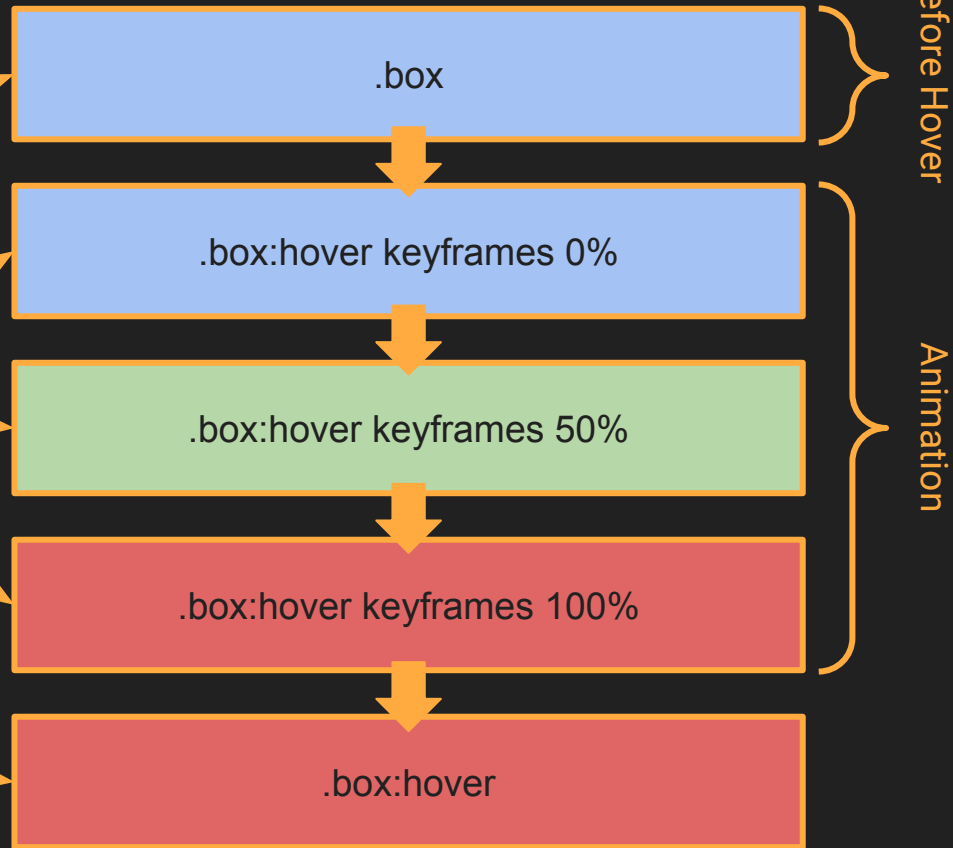
```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: redToYellow1;  
  animation-duration: 4s;  
}
```

*Demonstrate!*

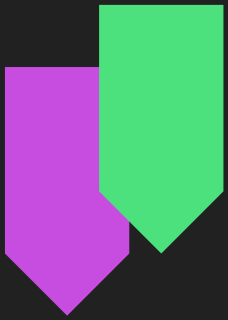


# Animation Breakdown

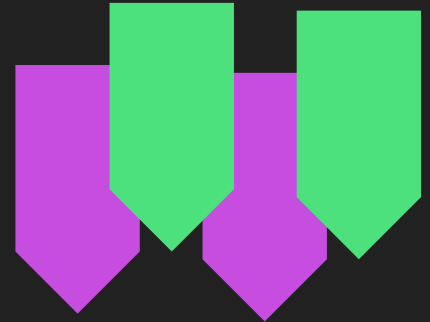
```
.box {  
  height: 100px;  
  width: 100px;  
  background-color: lightblue;  
}  
  
@keyframes blueToRed {  
  0% { background-color: lightblue; }  
  50% { background-color: green; }  
  100% { background-color: red; }  
}  
  
.box:hover{  
  height: 300px;  
  width: 300px;  
  animation: blueToRed 1s linear;  
  background-color: red;  
}
```



*Note that it helps to have the start and end of the animation match the CSS rules.*



## Activity: Growing Box 2



```
div.box {  
  height: 100px;  
  width: 100px;  
  background-color: lightblue;  
  transition: 1s;  
}
```

```
div.box:hover {  
  animation: grow 3s linear;  
  width: 300px;  
  height: 300px;  
}  
  
@keyframes grow {  
  0% {  
    height: 100px; width: 100px;  
  }  
  50% {  
    height: 300px; width: 100px;  
  }  
  100% {  
    height: 300px; width: 300px;  
  }  
}
```

## Activity: Growing Box 2

```
<div class="box"></div>
```

We're going to do a simple keyframes exercise.

1

Make a new HTML file, and add the above HTML to it. Also add the CSS on the left.

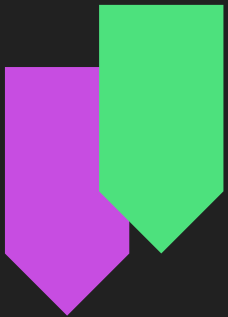
2

Then hover over the div.

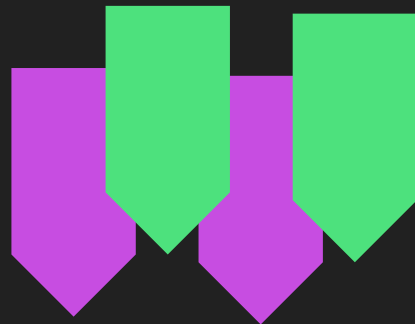
3

The exercise is to change the animation so:

- The box color changes to red, and stays red on hover
- The box grows to 500x500 by the end of the animation, and stays that size on hover.



# Animation Properties





# Animation Properties

The property we've used is `animation`. Animation is shorthand, and lets you set many other properties, like `-name`, `-duration`, and `-timing-function`.


Other properties also let you adjust the animation. If you want to do more animations in the future, they give you more control. Eg:

- Play in reverse
- Slow down or speed up
- Repeat
- Not having to set the CSS before or after the animation to match the start/end
- Delay the start

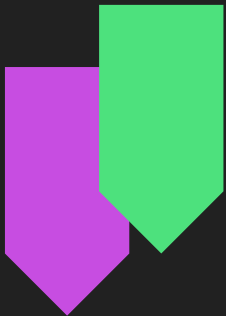
There are other properties that have to do with animation:

- `@keyframes`
- `animation-name`
- `animation-duration`
- `animation-delay`
- `animation-iteration-count`
- `animation-direction`
- `animation-timing-function`
- `animation-fill-mode`
- `animation`

*We are not going to cover these. But feel free to try them out!*







**Q&A**

