Maydm

Web Development

Day 5: Advanced CSS & Intro to Programming

Transitions, Animations & Block Coding

Maydm

Icebreaker!

Maydm

Today’s Schedule

Morning:

- Recap of CSS
- Pseudo classes
- Transitions
- Animations
- Project: Build an Animated Solar System

Afternoon:

- Field Trip!
- Intro to Programming
- Block coding
- Scratch

Recap of CSS

- How do you add CSS to elements?
- What is a CSS selector?
- What is CSS specificity?

Pseudo-classes

A pseudo-class is used to define a special state of an element.

It can be used to:

- Style an element when a user mouses over it
- Style visited and unvisited links differently
- Style an element when it gets focus

They are defined by a CSS selector followed by a **colon plus a keyword**.

Pseudo-classes

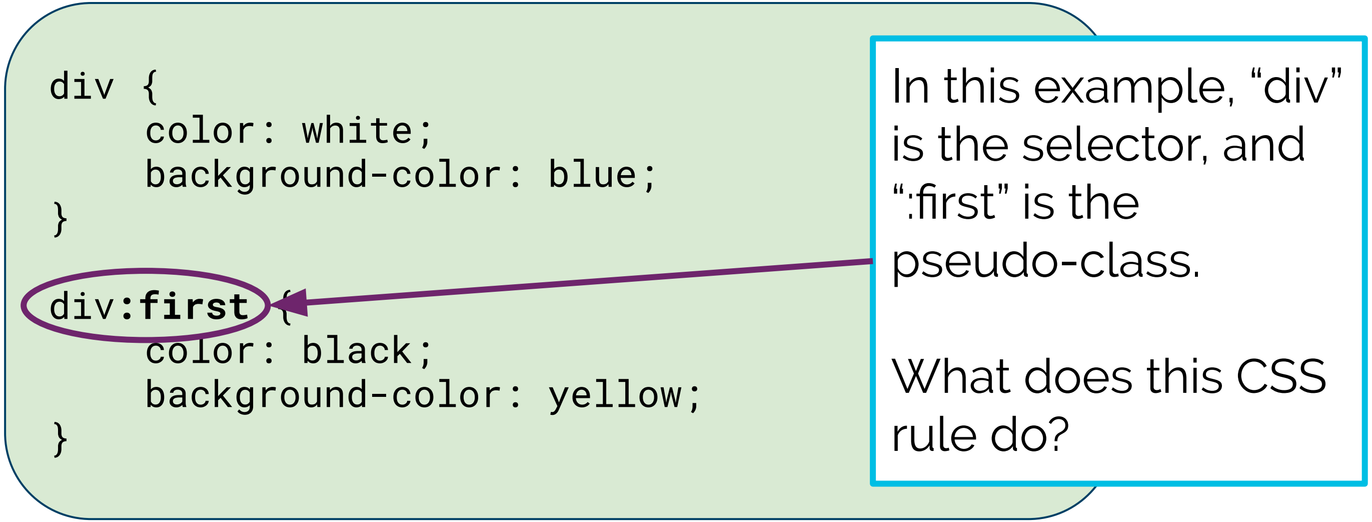
```
a {
    color: black;
    text-decoration: none;
}

a:hover {
    color: red;
    text-decoration: underline;
}
```

In this example, “**a**” is the selector, and “**:hover**” is the pseudo-class.

What does this CSS rule do?

Pseudo-classes



```
div {
     color: white;
     background-color: blue;
}

div:first {
     color: black;
     background-color: yellow;
}
```

In this example, “div” is the selector, and “:first” is the pseudo-class.

What does this CSS rule do?

Transitions

When using changing CSS properties on a web page, **transitions** are useful for controlling the speed of those changes and making them appear animated.

Transitions

Transitions have four properties:

- Transition-property: the CSS property that will be “animated”
- Transition-duration: how long the transition will last
- Transition-timing-function: changes the speed of the transition
- Transition-delay: how long to wait before the transition starts

They are usually written in shorthand:

```
transition: <property> <duration> <timing-function> <delay>;
```

Transitions

```
div {
    transition-property: height;
    transition-duration: 1s;
    transition-timing-function: ease-in;
    transition-delay: 0s;
}

div {
    transition: height 1s ease-in 0s;
}
```

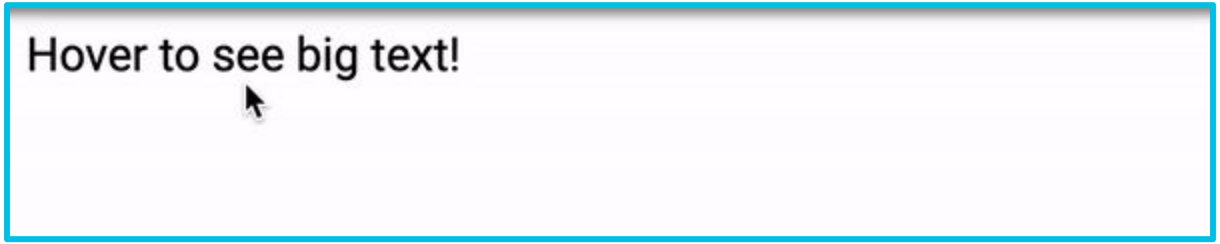
These CSS rules do the same thing.

Transitions

The **transition-timing-function** property is an *acceleration curve so that the speed of the transition can vary over its duration.*

Transitions

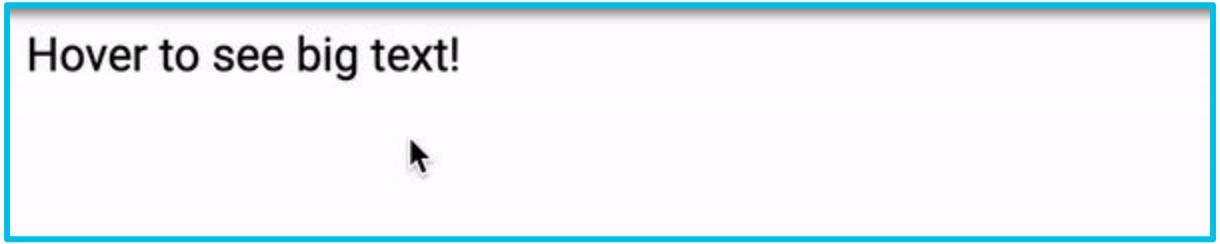
Using pseudo-class hover, no transition.



Hover to see big text!

Transitions

Using pseudo-class hover with transition (ease-in timing function).



Hover to see big text!

Transitions

Try it yourself:

- Open Big Text CodePen
- Add a transition to the .big-text class to make text size change smoothly
- Use the shorthand transition property:
    - transition: <property> <duration> <timing function> <delay>;

Maydm

Animations

Transitions are useful when you are going between two states -- for example, a link is hovered over, or it’s not.

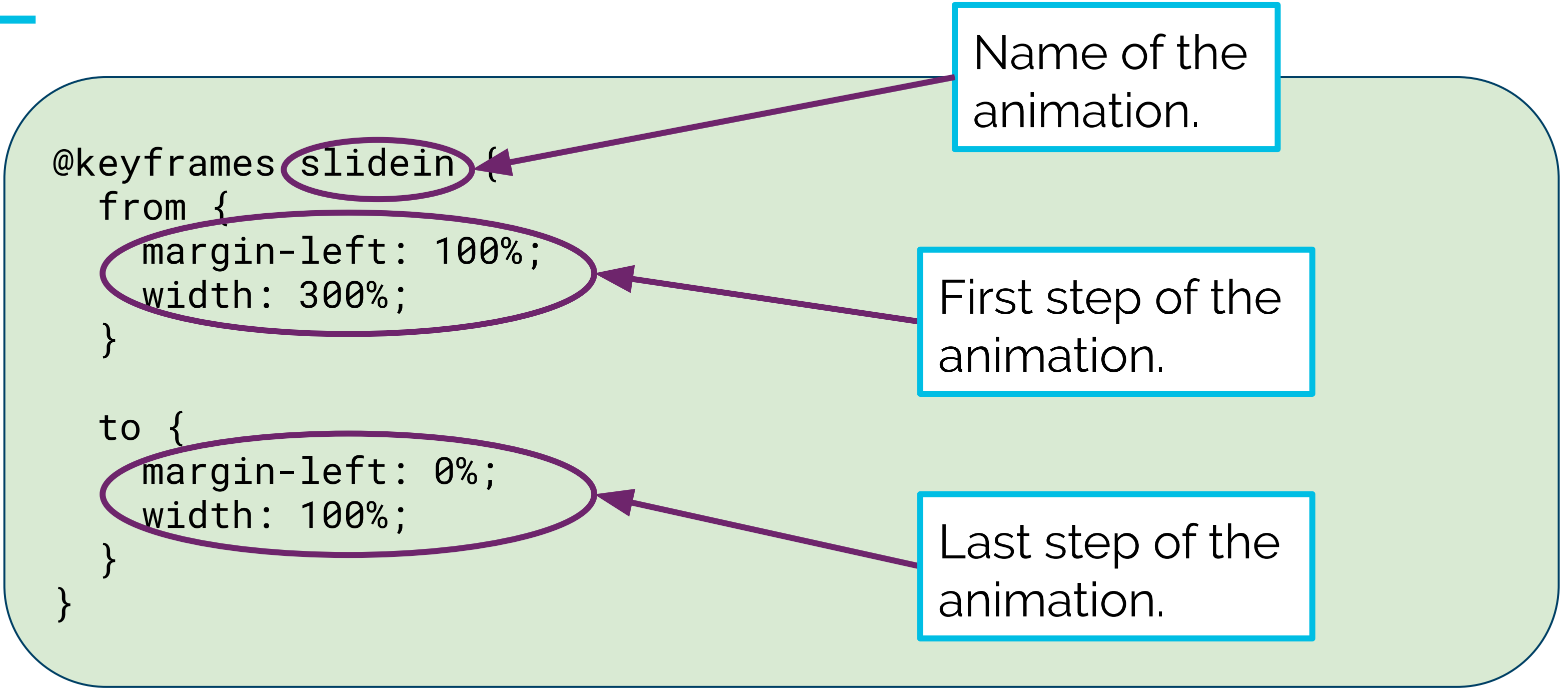
Animations are good for more complex changes, where there are many states, or if the animation begins on page load.

Animations

There are two parts to using CSS Animations:

- Creating the animation using the **@keyframes** keyword
- Assigning the animation to an element, which uses the following properties:
  - **animation-name**: the name of the animation
  - **animation-duration**: how long the animation lasts
  - **animation-iteration-count**: how many times the animation repeats
  - **animation-direction**: which direction the animation runs
- Just like transitions, there is a shorthand animation:
  animation: <name> <duration> <iteration-count> <direction>;

Animations



```
@keyframes slidein {
   from {
     margin-left: 100%;
     width: 300%;
   }

   to {
     margin-left: 0%;
     width: 100%;
   }
}
```

Name of the animation.

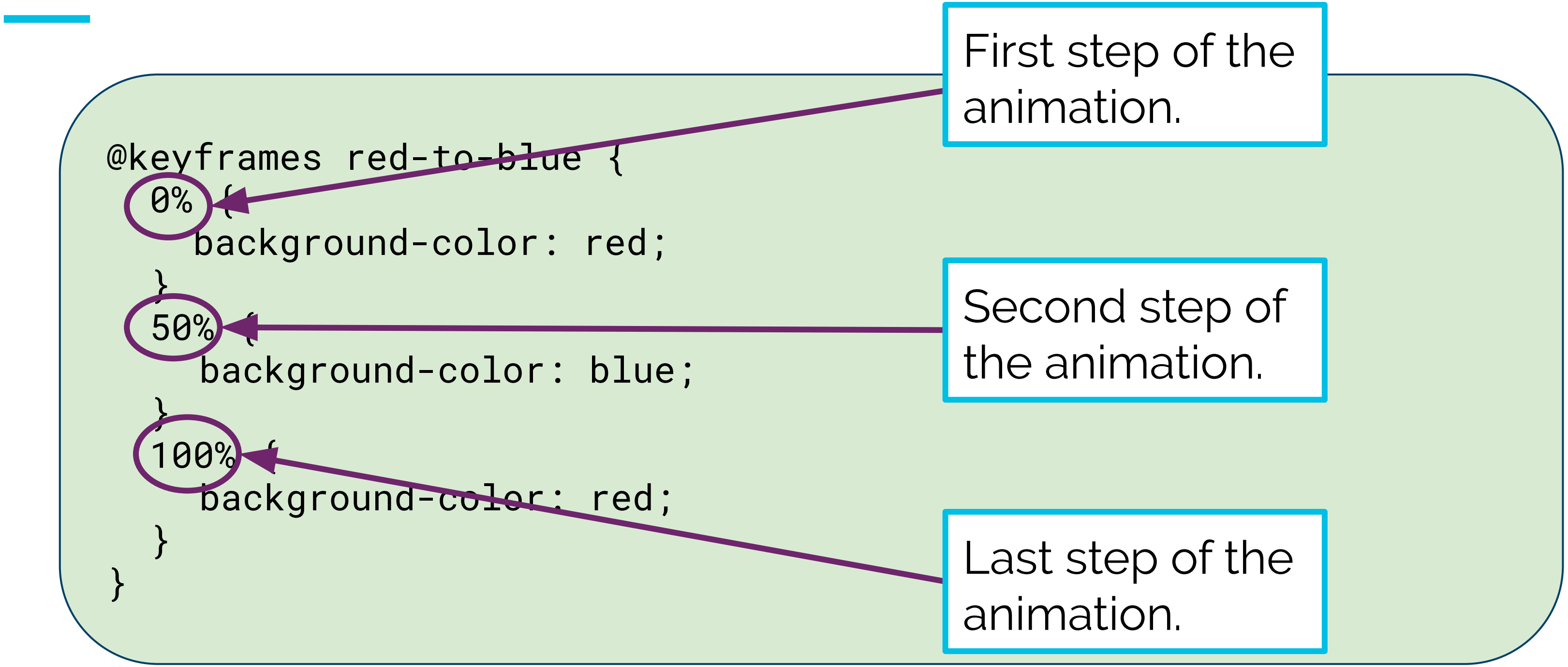
First step of the animation.

Last step of the animation.

Animations

Animations require at least a beginning and an ending state, but can take as many steps as necessary using percentages.

Animations



```
@keyframes red-to-blue {
  0% {
    background-color: red;
  }
  50% {
    background-color: blue;
  }
  100% {
    background-color: red;
  }
}
```

First step of the animation.

Second step of the animation.

Last step of the animation.

Animations

Assigning an animation to an element is similar to assigning a transition:

```
#circle {
    animation-name: red-to-blue;
    animation-duration: 4s;
    animation-iteration-count: infinite;
}

#circle {
    animation: red-to-blue 4s infinite;
}
```

These CSS rules do the same thing.

Animations

Try it yourself:

- Open Rainbow Circle CodePen
- Add an animation to the .circle class to make it change colors using the given animation
- Use the shorthand animation property:
    - animation: <animation-name> <duration> <iteration-count> <delay>;

Project: Animated Solar System

Add the missing CSS to animate the Solar System

- The animation is given to you (@keyframes rotate)
- You will need to add the animation to only one CSS selector that has been given to you
  - Hint: it will apply to all of the planets and the moon
  - Another hint: the planets are all <li>s with different IDs but the moon is referenced by "li#earth span"
- All of the planets will have a different rotation speed so each of the <li>s will have a different animation-duration.

What is Programming?

Programming is the writing of instructions for a computer to complete. The instructions, or **code**, are written in one of many programming languages.

What is Programming?

Popular programming languages include:

- C++
- Java
- Python
- JavaScript
- Block

What is Programming?

HTML & CSS are not considered programming languages.

Why do you think that is?

What is Block Coding?

- A visual way to learn coding
- Use predefined chunks of code, called “blocks”
- Learn the logic of coding
  - No learning the “grammar” of a coding language
  - No memorizing commands or names
- Scratch is a very popular Block Coding platform

Reflection

Write in your journal about how you feel or what you learned today.

Prompts:

- Explain the difference between CSS transitions and animations.
- Was today your first time using block coding? What do you think of it?
- How did you like Scratch? What game would you program on your own?