

COMP4021
Internet Computing

CSS Animations

Gibson Lam

CSS Animations

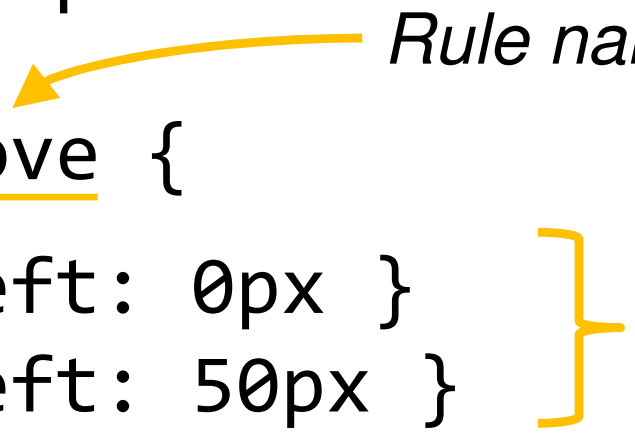
- You know that CSS can change visual styles of HTML
- It is interesting that CSS can also make things move inside HTML, e.g. moving some text from left to right:



- In this presentation, we will look at using CSS animations to do that

The Keyframes Rule

- You use a new style rule, *the keyframes rule*, to make animations
- Here is an example:



```
@keyframes move {  
    from { left: 0px }  
    to   { left: 50px }  
}
```

Rule name

The animation sequence (keyframes)

In this example, this animation rule changes the left property from 0px to 50px

Applying the Rule

- You have created the animation, let's use it on some HTML elements
- For example, you can use it on a header:

`<h1 style="position: relative;
animation: move 2s">`

I am bouncing!

`</h1>`

The keyframes rule to be used

The duration of the animation

Make its position adjustable

A diagram illustrating the application of a CSS animation rule. It shows an HTML header tag with a style attribute containing a CSS rule. The rule has two parts: a keyframe rule name and a duration. Yellow arrows point from descriptive text to these parts: one from 'The keyframes rule to be used' to 'move', and another from 'The duration of the animation' to '2s'. A third yellow arrow points from 'Make its position adjustable' to the 'position: relative;' part of the style attribute. The text 'I am bouncing!' is placed between the opening and closing header tags.

Moving Text Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
  <title>Moving Text</title>
```

```
  <style>
```

```
    @keyframes move {
```

```
      from { left: 0px }
```

```
      to   { left: 50px }
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1 style="position: relative;
    animation: move 2s">
```

```
    I am bouncing!
```

```
  </h1>
```

```
</body>
```

```
</html>
```

Left = 0 at 0 second

I am bouncing!



I am bouncing!



I am bouncing!

Left = 50 at 2 seconds

Freezing the Animation

- In the previous example, you can see the text ‘jumps’ back to the start after the animation finishes
- You may have expected the text to remain on the right hand side in the end
- To do that in CSS, you need an additional property, shown below:

`animation-fill-mode: forwards`

*Telling the browser to ‘freeze’
at the end of the animation*






Animating Multiple Elements

- You can apply a keyframes rule to multiple elements
- For example, you can apply it to two separate `<div>` using different durations:

```
<div id="box1" class="box"></div>  
<div id="box2" class="box"></div>
```

This is box1   

This is box2   

```
.box {  
    position: relative;  
    left: 0;  
    width: 50px;  
    height: 50px;  
}  
#box1 {  
    top: 0;  
    background: red;  
    animation: move 2s;  
}  
#box2 {  
    top: 80px;  
    background: blue;  
    animation: move 4s;  
}
```