COMP4021
Internet Computing

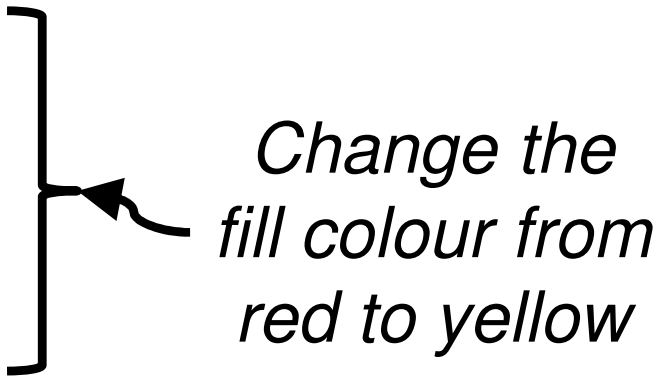
SVG and CSS Animations

Gibson Lam

SVG and CSS Animations

- You can use CSS animations to create animations for SVG elements
- Here is an example rule that can animate the `fill` colour of any SVG element:

```
@keyframes change-color {
    from { fill: red; }
    to   { fill: yellow; }
}
```

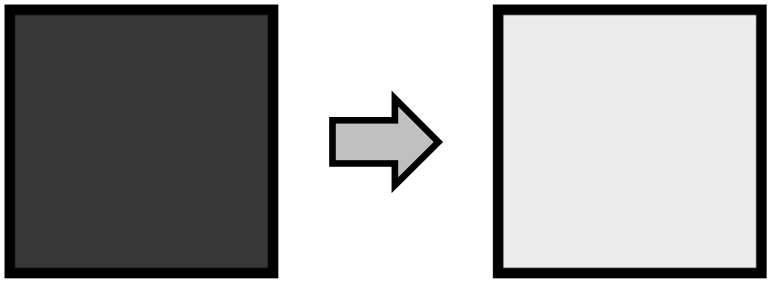


- An example is shown in the next slide

CSS Animation Example

- In this example, the rectangle changes from red to yellow in 4 seconds after loading the SVG

```
<style>
@keyframes change-color {
    from { fill: red; }
    to   { fill: yellow; }
}
rect {
    animation: change-color 4s;
}
</style>
```



The animation is applied to the `<rect>`

```
<rect x="50" y="50" width="100" height="100"
      stroke="black" stroke-width="4"
      fill="red"/>
```

Animating Movement

- Remember you have used the CSS `left` property to move HTML elements horizontally
- You cannot do that for SVG elements because they have no `left` property!

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

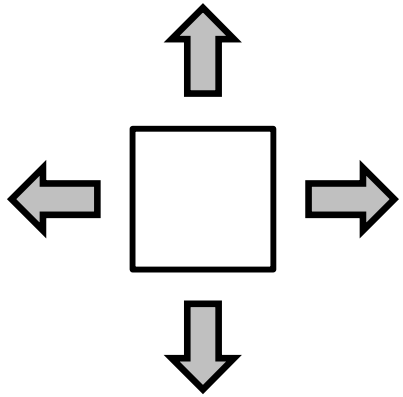
*This **DOES NOT** work for SVG elements*

- To move things in SVG, you need to use the `transform` property

The `transform` Property

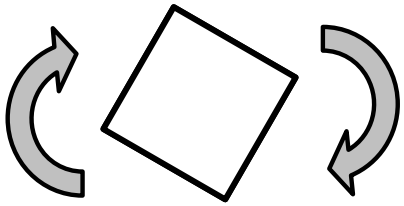
- Translation

  `transform: translate(x, y)`
  or
  `translateX(x)`
  `translateY(y)`



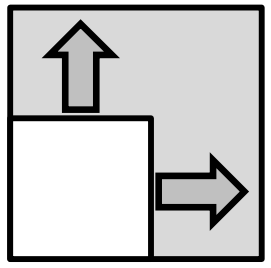
- Rotation

  `transform: rotate(angle)`



- Scaling

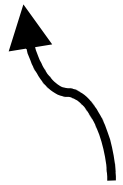
  `transform: scale(x, y)` *or*
  `scaleX(x)` *or* `scaleY(y)`



An Example Rule for Moving

- You can make an SVG element move to the right using this rule:

```
@keyframes move {
    from { transform: translateX(0px); }
    to   { transform: translateX(200px); }
}
```



- Note that you must provide a unit, e.g. ‘px’, and the value represents the movement away from the element’s original position

Using Percentages in Animations

- The previous example moves things from one place (`from`) to another (`to`)
- You can move things to various places in one single animation using percentages, i.e. `0%` represents '`from`' and `100%` represents '`to`'
- If you use percentages, you can create multiple 'keyframes', i.e. key points in the rule, as shown in the next example

An Example With Percentages

- Here is a rule with three percentage values:

This animation has two moving sequences

```
@keyframes move {
    0% {
        transform: translate(0px, 0px);
    }
    50% {
        transform: translate(0px, 200px);
    }
    100% {
        transform: translate(100px, 200px);
    }
}
```

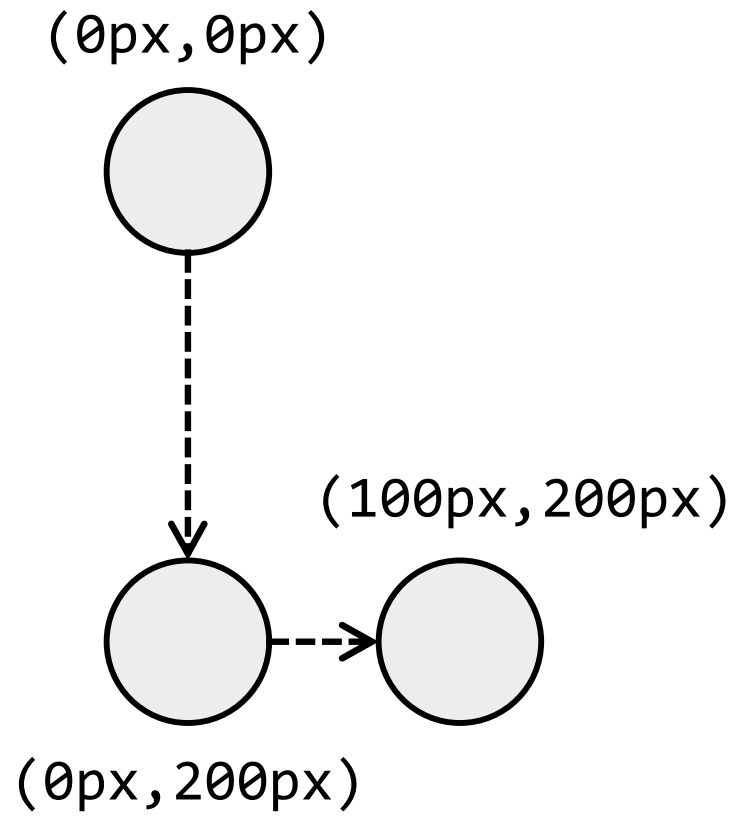
Move down ⇩

Move to the right ⇩

The Example Movement

- The animation moves things in an L shape
- Here is how a yellow circle moves by applying the rule to it:

```
@keyframes move {
  0% {
    transform: translate(0px, 0px);
  }
  50% {
    transform: translate(0px, 200px);
  }
  100% {
    transform: translate(100px, 200px);
  }
}
```



Animation Timing

- If you have more than two moving sequences, you can see the animation uses different speeds over the duration, i.e. sometimes it is faster and sometimes slower
- If you want to make the animation using the same timing, i.e. speed, you can use the additional CSS property on the SVG element

```
circle {
    animation: move 4s;
    animation-timing-function: linear;
}
```