

# CSS III

## (Transitions & Animations)

Programming with  
Web Technologies

# Last time

---

- The DOM
- Attribute, DOM & Pseudoclass selectors
- CSS box model, margins & padding
- CSS positioning

# Today

---

- Filters
- Transforms
- Transitions
- Animations

# Filters



Auckland  
ICT Graduate School

# CSS filters

---

- Graphical effects, applied to elements after they have been placed on the page
  - that change their appearance in a number of ways
- Most commonly used on images due to the supported operations, which include blurring, contrast shifting, brightness adjustment and greyscaling
- Controlled with the `filter` property. More than one effect can be specified in the same statement, and will be evaluated left-to-right

# CSS filters

```
filter: brightness(40%);
```

Decrease the brightness of an image to a percentage of the original

```
filter: contrast(200%);
```

Tweak the contrast to over-saturate an image

```
filter: blur(2px) sepia(70%);
```

Slightly blur an image, then apply a strong sepia filter to it



# Transformations

# CSS transformations

- Transformations allow us to perform certain operations - such as rotation or scaling - of HTML elements
- 2D & 3D transformations are supported
- Controlled with the `transform` property.
- Multiple transformations can be specified at once
- If built-in transform functions are insufficient, we can use the `matrix()` or `matrix3d()` functions to specify arbitrary mathematical transformations



# CSS transformations

```
transform: rotate3d(1, 1, 1, 75deg);
```

Rotate 75 degrees on all 3 axis



```
transform: scale(0.9, 0.2);
```

Scale the image to 90% width, 20% height



```
transform: rotate(5deg);
```

Rotate the image 5 degrees clockwise



Can combine transformations: e.g:

```
transform: rotate(5deg) scale(0.9, 0.2);
```

# Transitions



Auckland  
ICT Graduate School

# CSS transitions

- Using transitions, we can change CSS properties over a given time period
  - Gives a smoother feeling to a user experience.
- Simple to specify
  - Just indicate which property to watch, and how long the change should take.
- Transitions don't run immediately, they run once the property they are watching has changed.
  - Typically used with pseudo class selectors like :[hover](#)
- Controlled with the **transition** property. As with filters, more than one property can be watched in the same statement

# CSS transitions

```
#input[type=text] {  
  width: 100px;  
  transition-property: width;  
  transition-duration: 2s;  
  transition-timing-function: linear;  
  transition-delay: 0.5s;  
}
```

```
#input[type=text]:hover {  
  width: 200px;  
}
```

0.5 seconds after a text input gets focus, it will expand to 200px wide over 2 seconds

```
#identifier {  
  width: 50px;  
  transition: width 2s ease;  
}  
  
#identifier:hover {  
  width: 200px;  
}
```

When hovered over, this element will widen by 150px over 2 seconds, with the change happening slightly faster at the start and end of the transition.

# Animations



Auckland  
ICT Graduate School

# CSS animations

---

- Allow us to specify more complex visual effects compared with transitions
- Introduced to save animators from needing to write complex JavaScript code
  - Can still augment animations using JavaScript if CSS animations themselves aren't enough
- Animations are made up of a `@keyframes` block, as well as a collection of `animation-*` properties to control the behaviour of the defined animation

# CSS animations

```
h1 {  
  animation-duration: 3s;  
  animation-name: slide-right;  
}  
  
@keyframes slide-right {  
  from {  
    margin-left: 100%;  
    width: 300%  
  }  
  
  to {  
    margin-left: 0%;  
    width: 100%;  
  }  
}
```

This animation starts with its target off screen, accomplished by setting its left margin to 100%, then slides towards the left of the screen until its left side is touching the left of its container.

The h1 element in this case uses this animation and executes it over the course of 3 seconds

# CSS animations

```
@keyframes hrmm {  
  from {  
    width: 50px;  
    margin-left: 0px;  
    background-color: blue;  
  } 25% {  
    width: 150px;  
    margin-left: 0px;  
  } 50% {  
    width: 50px;  
    margin-left: 100px;  
    background-color: red;  
  } 75% {  
    width: 150px;  
    margin-left: 100px;  
  } to {  
    width: 50px;  
    margin-left: 200px;  
    background-color: green;  
  }  
}
```

```
#blue {  
  width: 50px;  
  height: 50px;  
  background-color: blue;  
  animation-duration: 3s;  
  animation-name: hrmm;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```

```
<div id="blue"></div>
```

Any guesses as to what this will do?



# Further Reading

---

- [MDN CSS Transitions](#)
- [MDN CSS Transforms](#)
- ['Just Add Water' CSS Animations](#)
- [MDN CSS Animations](#)
- [MDN Animatable CSS Properties](#)