CSS Positioning, Transforms, Transitions, & Animations

Web Development & Consulting Club
Workshop One, Session Two

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

Auckland
ICT Graduate School

CSS Positioning

CSS positioning

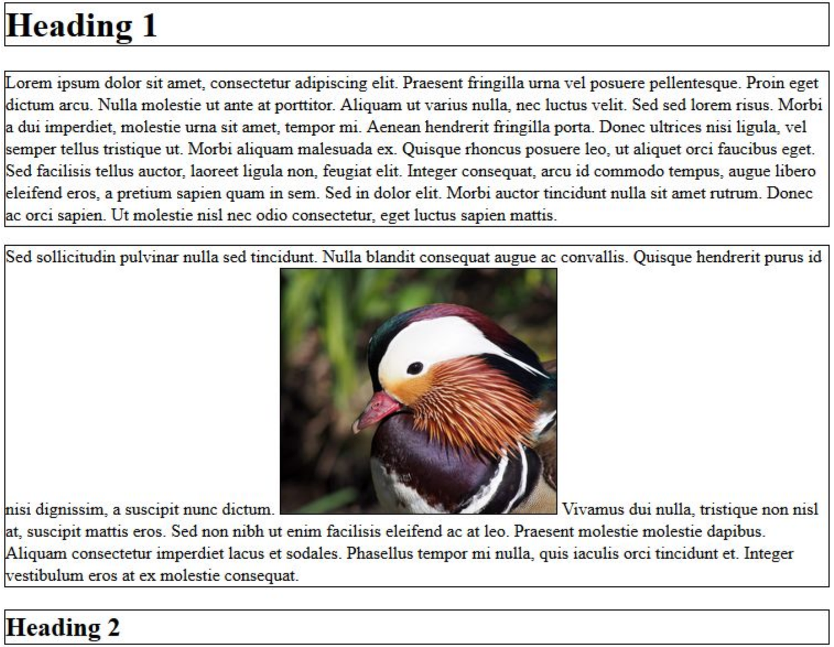
Up until now, all of our CSS has been positioned in **normal flow**

What this means for **block** elements is that they have been laid out vertically as boxes placed one after the other, top-to-bottom. Distance between other box is based on the margin properties of each, and the left edge of each box touches the left edge of its parent

For our **inline** elements, they have been laid out horizontally, left-to-right, separated by their margins, padding and borders. Vertical alignment can be by the element tops, bottoms or text baseline

CSS positioning - Normal flow

```
<h1>Heading 1</h1>
  <p>
     ...
  </p>
  <p>
    ...
    <img src="duck.png">
    ...
  </p>
<h2>Heading 2</h2>
```



Heading 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent fringilla urna vel posuere pellentesque. Proin eget dictum arcu. Nulla molestie ut ante at porttitor. Aliquam ut varius nulla, nec luctus velit. Sed sed lorem risus. Morbi a dui imperdiet, molestie urna sit amet, tempor mi. Aenean hendrerit fringilla porta. Donec ultrices nisi ligula, vel semper tellus tristique ut. Morbi aliquam malesuada ex. Quisque rhoncus posuere leo, ut aliquet orci faucibus eget. Sed facilisis tellus auctor, laoreet ligula non, feugiat elit. Integer consequat, arcu id commodo tempus, augue libero eleifend eros, a pretium sapien quam in sem. Sed in dolor elit. Morbi auctor tincidunt nulla sit amet rutrum. Donec ac orci sapien. Ut molestie nisl nec odio consectetur, eget luctus sapien mattis.

Sed sollicitudin pulvinar nulla sed tincidunt. Nulla blandit consequat augue ac convallis. Quisque hendrerit purus id nisi dignissim, a suscipit nunc dictum. Vivamus dui nulla, tristique non nisl at, suscipit mattis eros. Sed non nibh ut enim facilisis eleifend ac at leo. Praesent molestie molestie dapibus. Aliquam consectetur imperdiet lacus et sodales. Phasellus tempor mi nulla, quis iaculis orci tincidunt et. Integer vestibulum eros at ex molestie consequat.

Heading 2

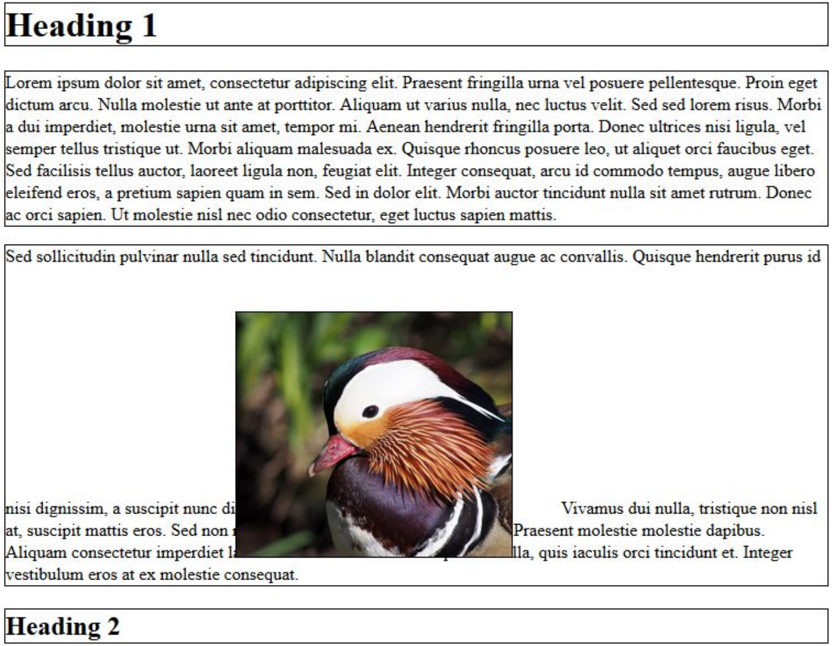
CSS positioning

The way an element is positioned is controlled by the `position` property, and this can be set to one of 5 values

`static`	(default) Positioned using normal flow
`relative`	Positioned relative to where it would be using normal flow by providing an offset
`absolute`	Removed from normal flow and positioned by coordinates relative to the containing block Note: you may need to add position relative to the containing/parent element for this to work
`fixed`	A constant position relative (normally) to the window i.e. not affected by scrolling
`sticky`	Toggles between relative and fixed depending on scroll position

CSS positioning - Relative

```
img {
  position: relative;
  left: -40px;
  top: 40px;
  /* Could use right or bottom */
}
```



Heading 1

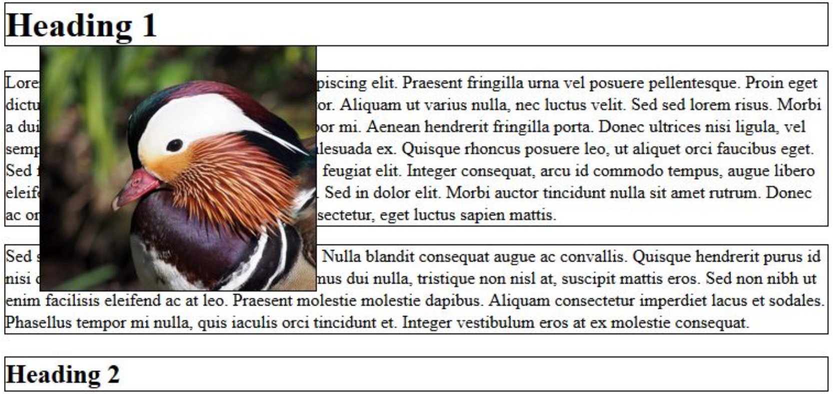
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent fringilla urna vel posuere pellentesque. Proin eget dictum arcu. Nulla molestie ut ante at porttitor. Aliquam ut varius nulla, nec luctus velit. Sed sed lorem risus. Morbi a dui imperdiet, molestie urna sit amet, tempor mi. Aenean hendrerit fringilla porta. Donec ultrices nisi ligula, vel semper tellus tristique ut. Morbi aliquam malesuada ex. Quisque rhoncus posuere leo, ut aliquet orci faucibus eget. Sed facilisis tellus auctor, laoreet ligula non, feugiat elit. Integer consequat, arcu id commodo tempus, augue libero eleifend eros, a pretium sapien quam in sem. Sed in dolor elit. Morbi auctor tincidunt nulla sit amet rutrum. Donec ac orci sapien. Ut molestie nisl nec odio consectetur, eget luctus sapien mattis.

Sed sollicitudin pulvinar nulla sed tincidunt. Nulla blandit consequat augue ac convallis. Quisque hendrerit purus id
nisi dignissim, a suscipit nunc di Vivamus dui nulla, tristique non nisl
at, suscipit mattis eros. Sed non r Praesent molestie molestie dapibus.
Aliquam consectetur imperdiet la lla, quis iaculis orci tincidunt et. Integer
vestibulum eros at ex molestie consequat.

Heading 2

CSS positioning - Absolute

```css
img {
  position: absolute;
  left: 40px;
  top: 60px;
  /* Could use right or bottom */
}
```



Heading 1

Lore piscing elit. Praesent fringilla urna vel posuere pellentesque. Proin eget
dictu or. Aliquam ut varius nulla, nec luctus velit. Sed sed lorem risus. Morbi
a dui or mi. Aenean hendrerit fringilla porta. Donec ultrices nisi ligula, vel
semp lesuada ex. Quisque rhoncus posuere leo, ut aliquet orci faucibus eget.
Sed feugiat elit. Integer consequat, arcu id commodo tempus, augue libero
eleif . Sed in dolor elit. Morbi auctor tincidunt nulla sit amet rutrum. Donec
ac or sectetur, eget luctus sapien mattis.

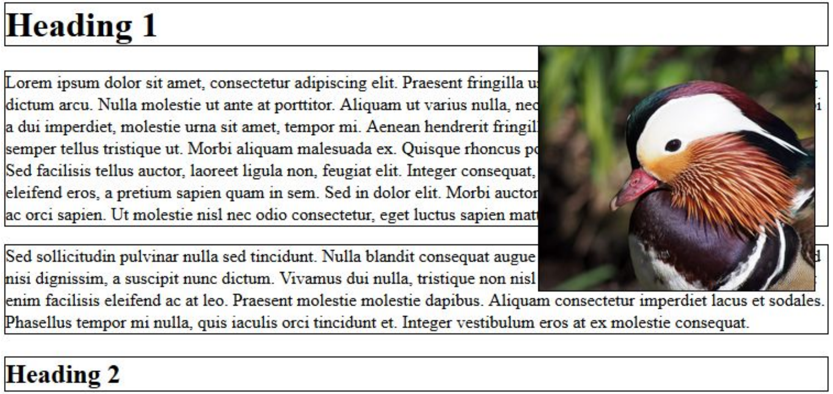
Sed Nulla blandit consequat augue ac convallis. Quisque hendrerit purus id
nisi mus dui nulla, tristique non nisl at, suscipit mattis eros. Sed non nibh ut
enim facilisis eleifend ac at leo. Praesent molestie molestie dapibus. Aliquam consectetur imperdiet lacus et sodales.
Phasellus tempor mi nulla, quis iaculis orci tincidunt et. Integer vestibulum eros at ex molestie consequat.

Heading 2

Absolute positioning positions within the first containing element that is not in **normal flow**. In this case, the paragraph where the image is placed is in the normal flow. The closest container that would make sense would be the body element

CSS positioning - Fixed

```
img {
  position: fixed;
  right: 20px;
  top: 60px;
}
```



Heading 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent fringilla u
dictum arcu. Nulla molestie ut ante at porttitor. Aliquam ut varius nulla, nec
a dui imperdiet, molestie urna sit amet, tempor mi. Aenean hendrerit fringil
semper tellus tristique ut. Morbi aliquam malesuada ex. Quisque rhoncus po
Sed facilisis tellus auctor, laoreet ligula non, feugiat elit. Integer consequat,
eleifend eros, a pretium sapien quam in sem. Sed in dolor elit. Morbi auctor
ac orci sapien. Ut molestie nisl nec odio consectetur, eget luctus sapien mat

Sed sollicitudin pulvinar nulla sed tincidunt. Nulla blandit consequat augue
nisi dignissim, a suscipit nunc dictum. Vivamus dui nulla, tristique non nisl
enim facilisis eleifend ac at leo. Praesent molestie molestie dapibus. Aliquam consectetur imperdiet lacus et sodales.
Phasellus tempor mi nulla, quis iaculis orci tincidunt et. Integer vestibulum eros at ex molestie consequat.

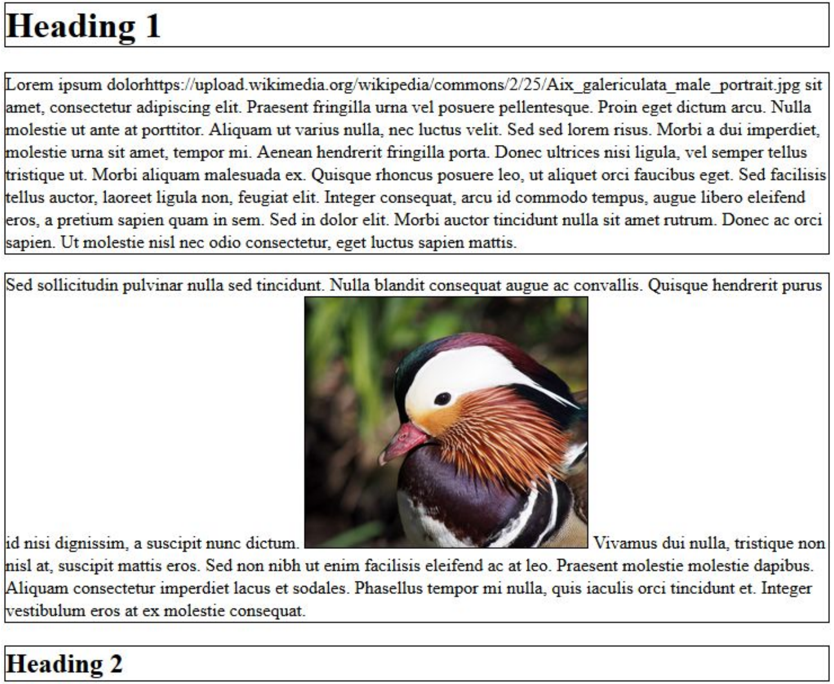
Heading 2

Fixed positioning is used when you want something to always stay at the same position on the screen. A fixed position element will not scroll with the page

CSS positioning - Sticky

```
img {
  position: -webkit-sticky;
  position: sticky;
  top: 0px;
}
```

Sticky is difficult to visualize with a static image, but when the image reaches the top of the page due to scrolling, it will 'stick' to it, acting like a fixed positioned element



Heading 1

Lorem ipsum dolorhttps://upload.wikimedia.org/wikipedia/commons/2/25/Aix_galericulata_male_portrait.jpg sit amet, consectetur adipiscing elit. Praesent fringilla urna vel posuere pellentesque. Proin eget dictum arcu. Nulla molestie ut ante at porttitor. Aliquam ut varius nulla, nec luctus velit. Sed sed lorem risus. Morbi a dui imperdiet, molestie urna sit amet, tempor mi. Aenean hendrerit fringilla porta. Donec ultrices nisi ligula, vel semper tellus tristique ut. Morbi aliquam malesuada ex. Quisque rhoncus posuere leo, ut aliquet orci faucibus eget. Sed facilisis tellus auctor, laoreet ligula non, feugiat elit. Integer consequat, arcu id commodo tempus, augue libero eleifend eros, a pretium sapien quam in sem. Sed in dolor elit. Morbi auctor tincidunt nulla sit amet rutrum. Donec ac orci sapien. Ut molestie nisl nec odio consectetur, eget luctus sapien mattis.

Sed sollicitudin pulvinar nulla sed tincidunt. Nulla blandit consequat augue ac convallis. Quisque hendrerit purus id nisi dignissim, a suscipit nunc dictum. Vivamus dui nulla, tristique non nisl at, suscipit mattis eros. Sed non nibh ut enim facilisis eleifend ac at leo. Praesent molestie molestie dapibus. Aliquam consectetur imperdiet lacus et sodales. Phasellus tempor mi nulla, quis iaculis orci tincidunt et. Integer vestibulum eros at ex molestie consequat.

Heading 2

CSS positioning - Float

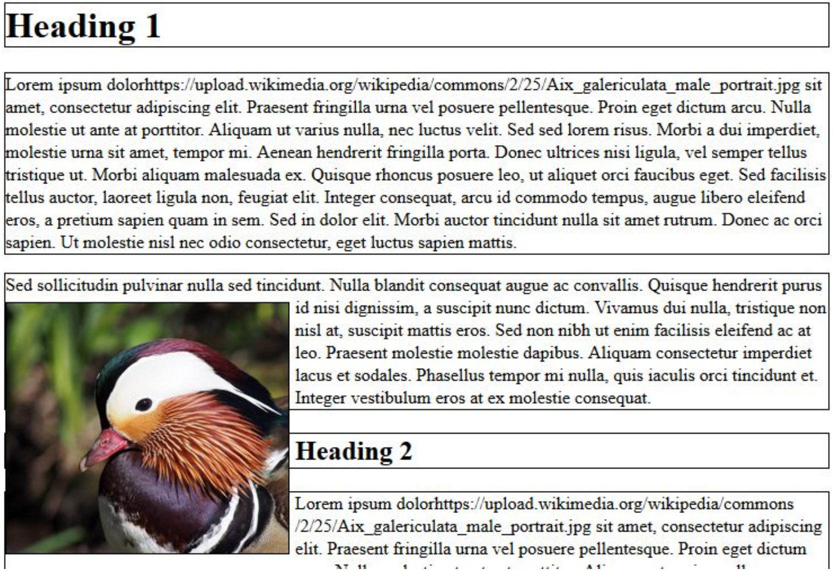
Sometimes, we want to be able to remove elements from the normal flow. This is known as **floating** an element. When an element is floated, it moves to the left or right side of its container, and all other content will flow around it. The `float` property controls the floating state of an element

After an element has been floated, all other content in the normal flow will flow around the element. If this is not what is wanted, you can correct this by clearing the float with the `clear` property

CSS positioning - Float

```
img {
  float: left;
  margin: 5px 5px 5px 0px;
}
```

Even though the image has been floated, we can still use box-model properties such as margin to affect how they interact with other elements. In this case a margin has been added to keep the wrapped text a short distance from the image



Heading 1

Lorem ipsum dolorhttps://upload.wikimedia.org/wikipedia/commons/2/25/Aix_galericulata_male_portrait.jpg sit amet, consectetur adipiscing elit. Praesent fringilla urna vel posuere pellentesque. Proin eget dictum arcu. Nulla molestie ut ante at porttitor. Aliquam ut varius nulla, nec luctus velit. Sed sed lorem risus. Morbi a dui imperdiet, molestie urna sit amet, tempor mi. Aenean hendrerit fringilla porta. Donec ultrices nisi ligula, vel semper tellus tristique ut. Morbi aliquam malesuada ex. Quisque rhoncus posuere leo, ut aliquet orci faucibus eget. Sed facilisis tellus auctor, laoreet ligula non, feugiat elit. Integer consequat, arcu id commodo tempus, augue libero eleifend eros, a pretium sapien quam in sem. Sed in dolor elit. Morbi auctor tincidunt nulla sit amet rutrum. Donec ac orci sapien. Ut molestie nisl nec odio consectetur, eget luctus sapien mattis.

Sed sollicitudin pulvinar nulla sed tincidunt. Nulla blandit consequat augue ac convallis. Quisque hendrerit purus id nisi dignissim, a suscipit nunc dictum. Vivamus dui nulla, tristique non nisl at, suscipit mattis eros. Sed non nibh ut enim facilisis eleifend ac at leo. Praesent molestie molestie dapibus. Aliquam consectetur imperdiet lacus et sodales. Phasellus tempor mi nulla, quis iaculis orci tincidunt et. Integer vestibulum eros at ex molestie consequat.

Heading 2

Lorem ipsum dolorhttps://upload.wikimedia.org/wikipedia/commons/2/25/Aix_galericulata_male_portrait.jpg sit amet, consectetur adipiscing elit. Praesent fringilla urna vel posuere pellentesque. Proin eget dictum

CSS units

Absolute units are useful assuming you know the characteristics of the device you are displaying on

in	Inches
cm	Centimetres
mm	Millimetres
pt	Points (1/72nd of an inch)
pc	Picas (1/6th of an inch, 12 points)
vw, vh, vmin	Percentage of viewport width, height, or min(width, height)

CSS units

Relative units are useful if you want your elements to be sized relative to some other length property in the document. Relative units scale better than absolute units when displaying across different devices

px	Pixels. Depends upon the resolution of the viewing device
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	x-height, relative to the size of a lower-case "x" in the font being used

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%);
```

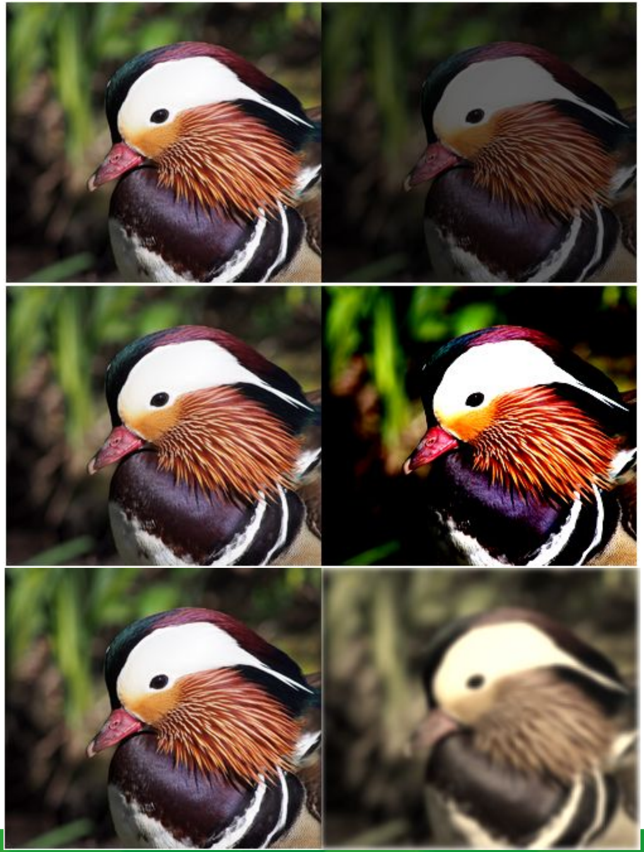
Decrease the brightness of an image to a percentage of the original

```
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

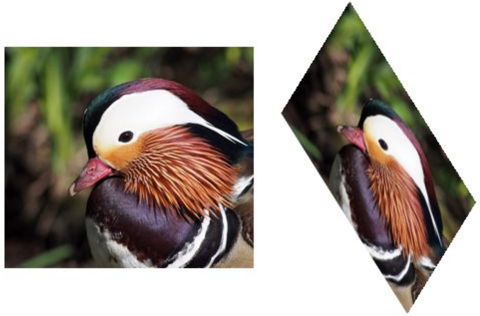
CSS transformations

- 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

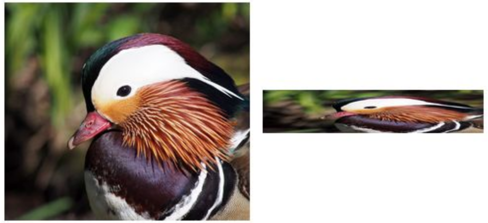
```
transform: rotate3d(1, 1, 1, 75deg);
```

Rotate 75 degrees on all 3 axis



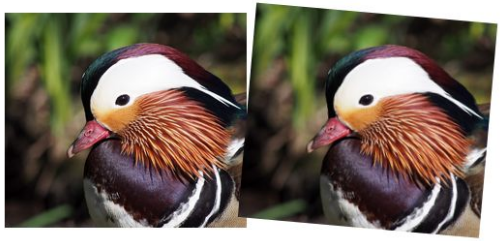
```
transform: scale(0.9, 0.2);
```

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

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

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

```css
@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;
   }
}
```

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

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

Any guesses as to what this will do?