# CSY1063
# Web Development
# Week 5

Chris.Rafferty@northampton.ac.uk

# Learning Objectives

- This week we will be covering
  - :hover effects
    - Drop-down navigation
  - CSS transitions
    - Fade effect
    - Slide effect
  - Advanced animation
    - Keyframes

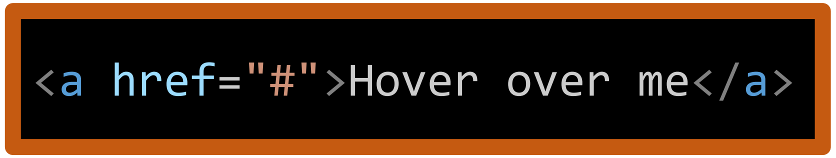
# Hover effects

- In the early days of CSS, the styles were applied during page load and that was it
- There was no interactivity, one the page was drawn on the screen that was it

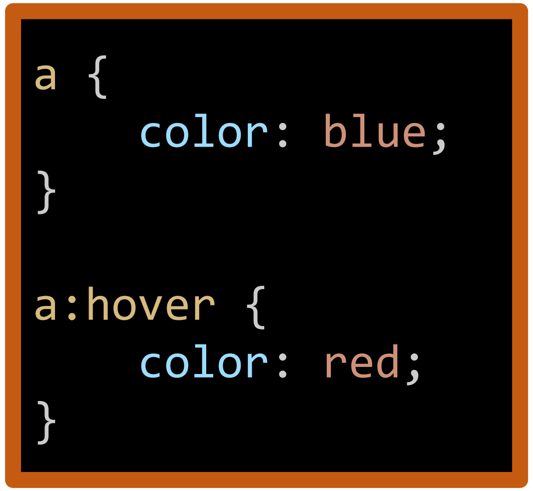
- Later, several interactive elements were added that allowed state changes
- One of these state changes is the :target selector
    - Last week

# :hover

- Another CSS state that exists is :hover
- Originally it only worked on links to allow hover effects when placing the mouse over a link



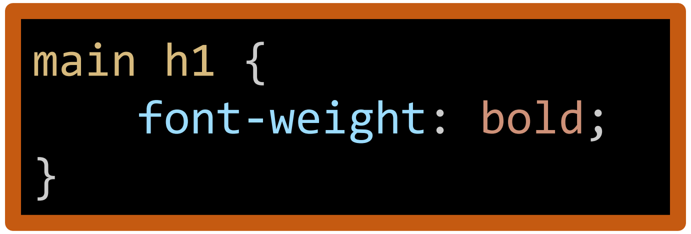
```
<a href="#">Hover over me</a>
```



```
a {
    color: blue;
}

a:hover {
    color: red;
}
```

# Nesting with :hover

- Remember how CSS selectors can target nested elements?

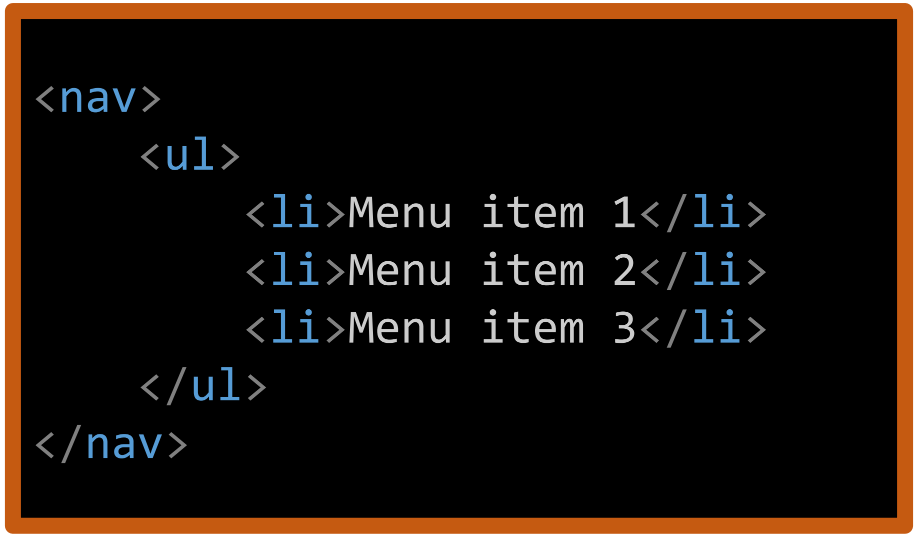


```
main h1 {
    font-weight: bold;
}
```

- This will target any h1 element inside a main element
- You can also nest elements inside :hover
- This is very useful for displaying a drop-down menu

# Nesting with :hover pt2

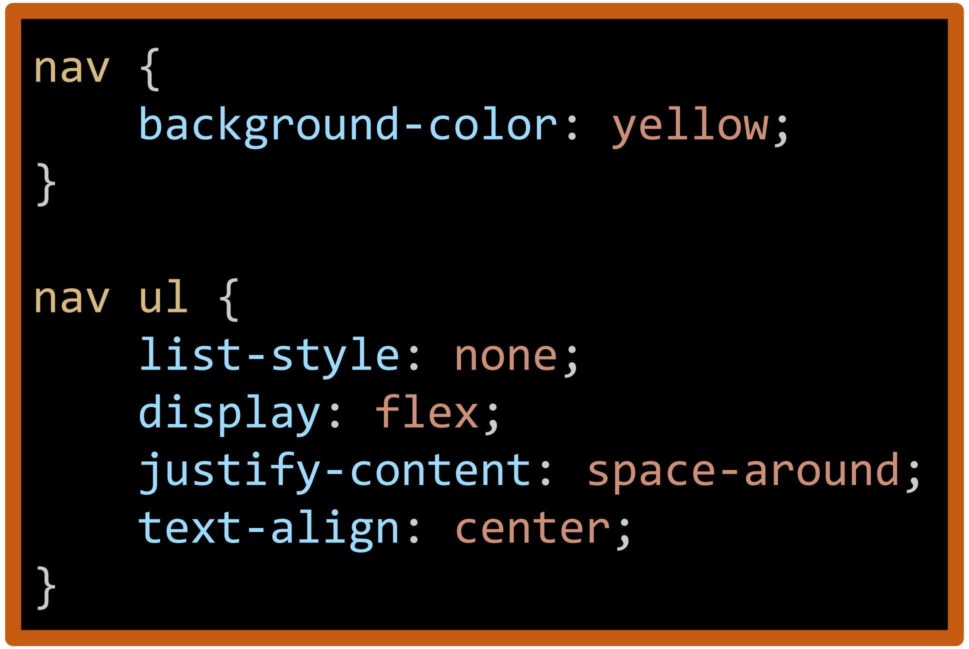
- You can use this to create a navigation bar with mouse over drop downs
- Firstly, you need some HTML for the menu



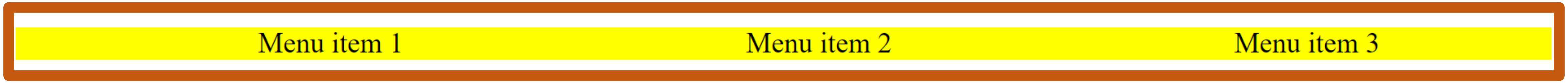
```
<nav>
    <ul>
        <li>Menu item 1</li>
        <li>Menu item 2</li>
        <li>Menu item 3</li>
    </ul>
</nav>
```

# Styling the navigation

- Some styling to get it across the page



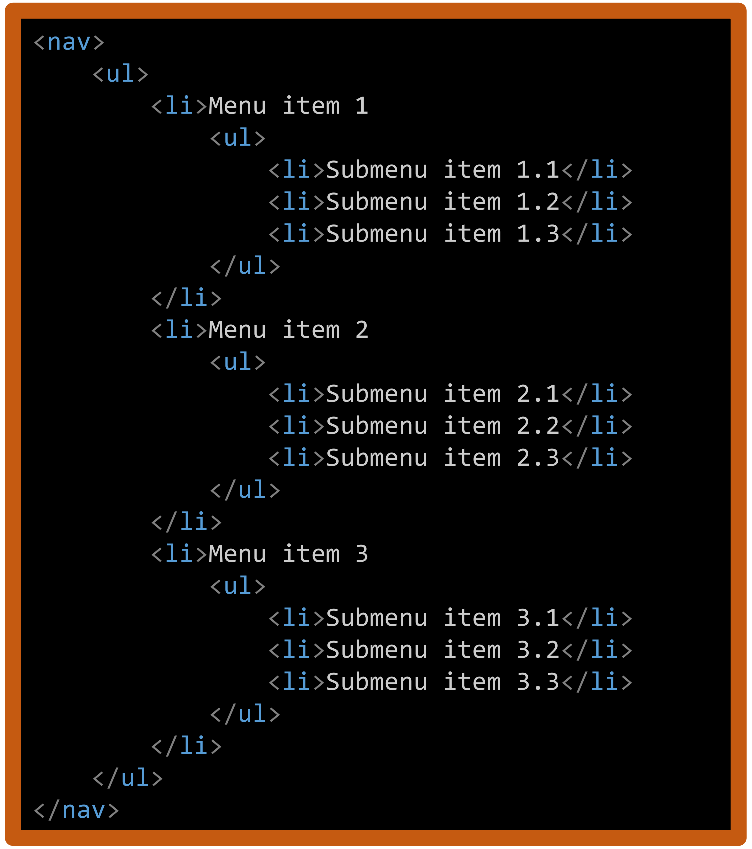
```
nav {
    background-color: yellow;
}

nav ul {
    list-style: none;
    display: flex;
    justify-content: space-around;
    text-align: center;
}
```



Menu item 1 Menu item 2 Menu item 3

# :hover for submenus

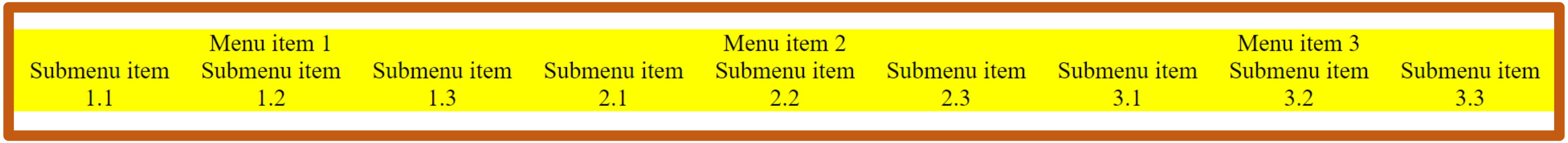
- A submenu can be added with a secondary tag for each menu



```
<nav>
    <ul>
        <li>Menu item 1
            <ul>
                <li>Submenu item 1.1</li>
                <li>Submenu item 1.2</li>
                <li>Submenu item 1.3</li>
            </ul>
        </li>
        <li>Menu item 2
            <ul>
                <li>Submenu item 2.1</li>
                <li>Submenu item 2.2</li>
                <li>Submenu item 2.3</li>
            </ul>
        </li>
        <li>Menu item 3
            <ul>
                <li>Submenu item 3.1</li>
                <li>Submenu item 3.2</li>
                <li>Submenu item 3.3</li>
            </ul>
        </li>
    </ul>
</nav>
```

# Submenu

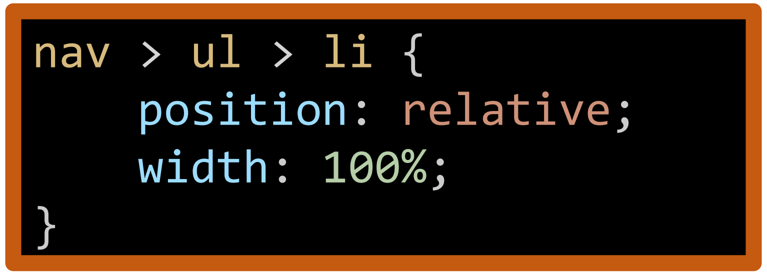
- If you run this code, it won't look very nice



```
                 Menu item 1                                   Menu item 2                                   Menu item 3
Submenu item   Submenu item   Submenu item   Submenu item   Submenu item   Submenu item   Submenu item   Submenu item   Submenu item
    1.1            1.2            1.3            2.1            2.2            2.3            3.1            3.2            3.3
```

# Positioning the submenus

- Set the containing list item to position: relative and the width to 100%

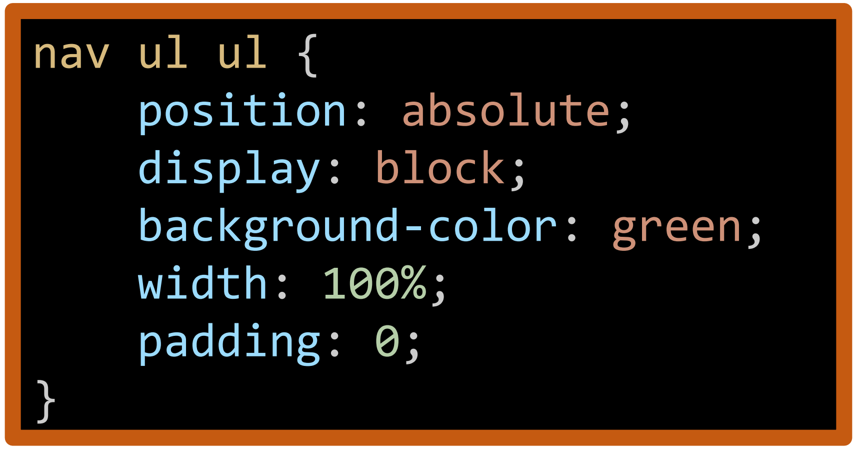


```css
nav > ul > li {
    position: relative;
    width: 100%;
}
```

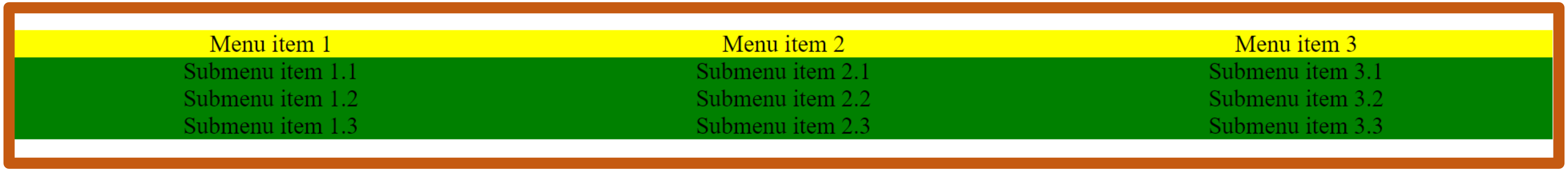
- This makes the elements inside the li (the <ul> for the submenu) positioned relative to this element. Normally elements are positioned relative to the <body> element!
  - Note the direct descendant operator. This applies the style to only the first level of <li> elements (more on this next week)

# Style the submenus

- Add some CSS to the submenus. Choose any colours you like, but position: absolute, display: block, width: 100% and padding: 0 are necessary to position it correctly!



```
nav ul ul {
    position: absolute;
    display: block;
    background-color: green;
    width: 100%;
    padding: 0;
}
```



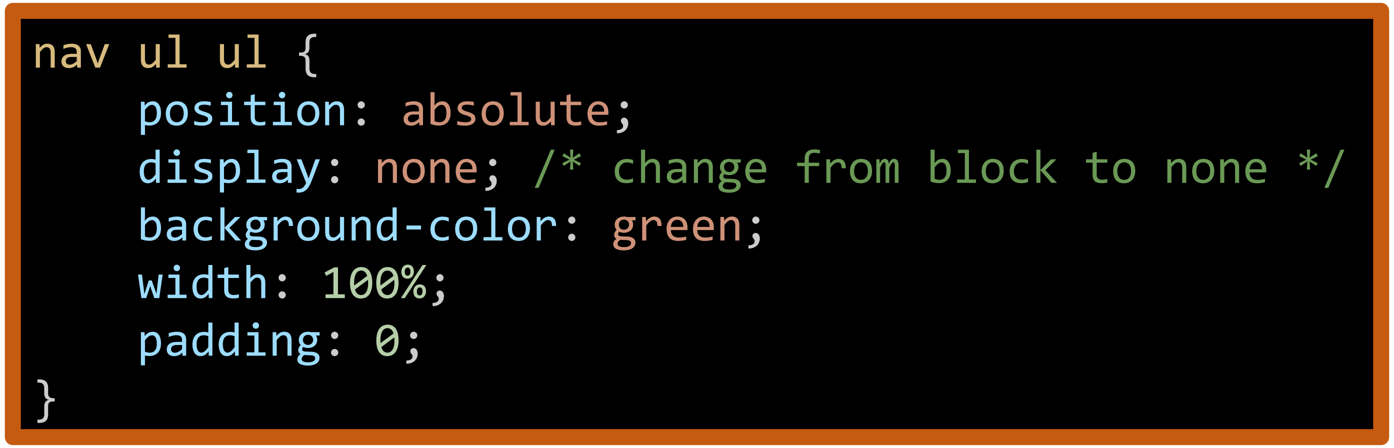
| Menu item 1 | Menu item 2 | Menu item 3 |
|---|---|---|
| Submenu item 1.1 | Submenu item 2.1 | Submenu item 3.1 |
| Submenu item 1.2 | Submenu item 2.2 | Submenu item 3.2 |
| Submenu item 1.3 | Submenu item 2.3 | Submenu item 3.3 |

# Use :hover to toggle the menu

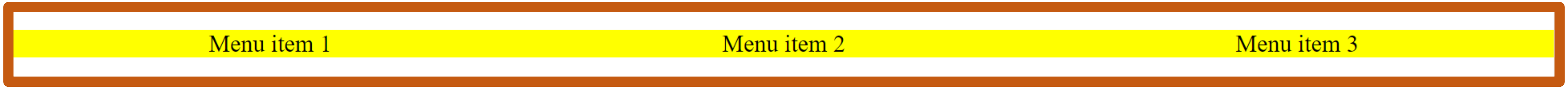
- Now the menu is showing all the time
- The goal is to make it so that when the mouse is hovered over the top-level menu, the secondary level menu is displayed
- This can be done with :hover
- You can apply :hover to any element in a nested CSS selector!

# Hide the menu

- Hide the submenus by default by setting them to display: none



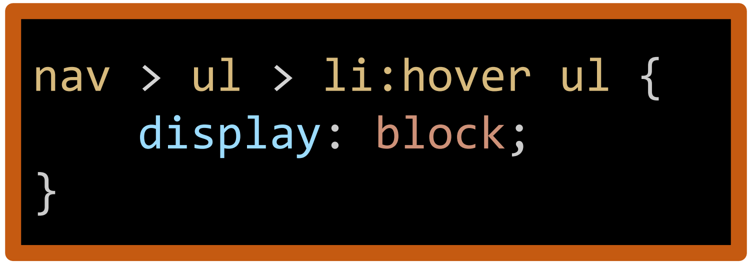
```
nav ul ul {
    position: absolute;
    display: none; /* change from block to none */
    background-color: green;
    width: 100%;
    padding: 0;
}
```



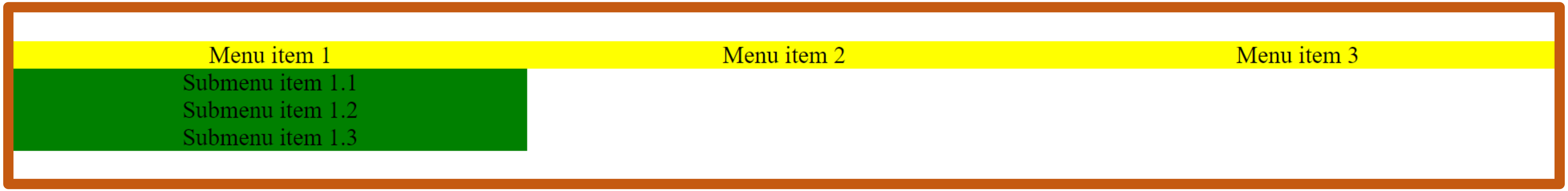
```
Menu item 1                    Menu item 2                    Menu item 3
```

# Add the hover effect

- When the top level <li> is hovered over, show the containing <ul>



```css
nav > ul > li:hover ul {
    display: block;
}
```



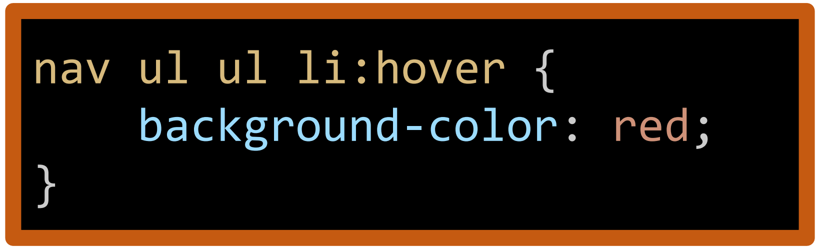
Menu item 1
Submenu item 1.1
Submenu item 1.2
Submenu item 1.3

Menu item 2

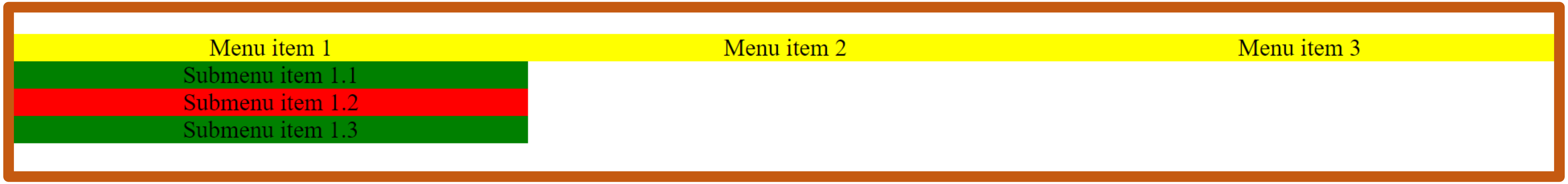
Menu item 3

# Highlight the menu

- You can also use hover to highlight which submenu the user has selected



```
nav ul ul li:hover {
    background-color: red;
}
```



Menu item 1

Submenu item 1.1

Submenu item 1.2

Submenu item 1.3

Menu item 2

Menu item 3

# Animations

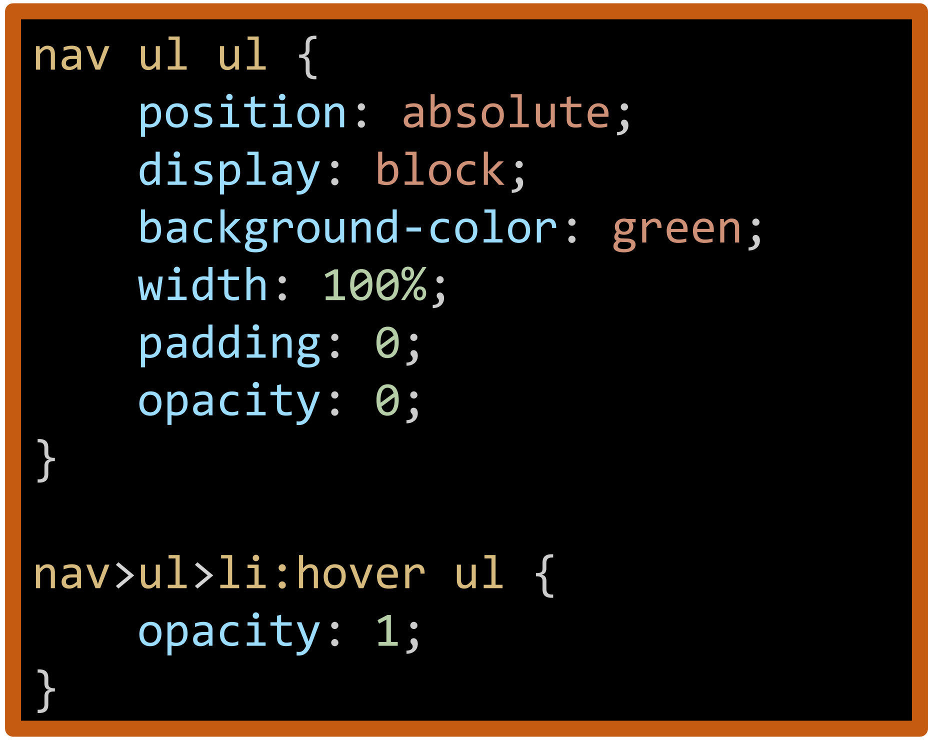
- The next stage will be animating the menu coming in rather than just having it instantly toggle on and off
- There are lots of different ways to animate it
- An animation needs two states
    - Before
    - After

# Fade effect

- The CSS property opacity controls how opaque and element is
- opacity: 1 means the element is completely visible
- opacity: 0 means the element is completely see-through

# Fade effect pt2

- Let’s change our CSS to use opacity instead of display to toggle the menu



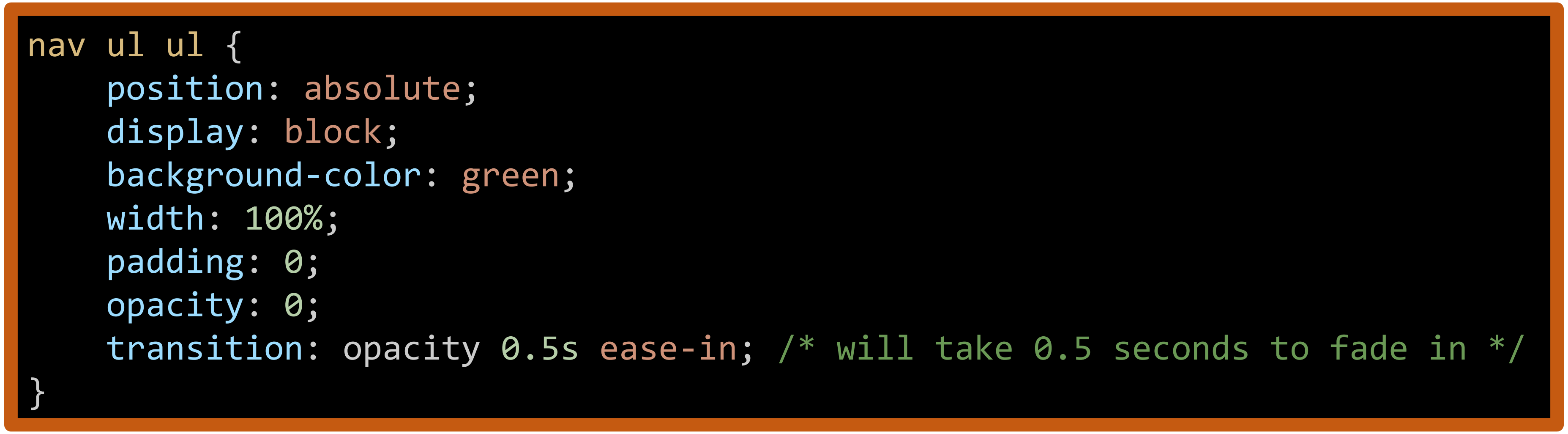
```
nav ul ul {
    position: absolute;
    display: block;
    background-color: green;
    width: 100%;
    padding: 0;
    opacity: 0;
}

nav>ul>li:hover ul {
    opacity: 1;
}
```

# Fade effect pt3

- This appears to have the exact same effect as using display: none and display: block
- The menu instantly appears and disappears on mouse over

# Transitions

- Opacity can support any value from 0-1
- 0.5 means semi-transparent
- CSS can be used to transition an property
  - You can control how long something goes on for (fade in for 0.5 seconds)



```
nav ul ul {
    position: absolute;
    display: block;
    background-color: green;
    width: 100%;
    padding: 0;
    opacity: 0;
    transition: opacity 0.5s ease-in; /* will take 0.5 seconds to fade in */
}
```

# Fade effect issue

- If you test this in your browser the menu will fade in and out as you hover over it!
- But there is a problem…

# Pointer-events

- Because the submenu is just invisible rather than removed from the page using display: none, if you hover over where the menu should be, it appears
- This would be a problem if someone wanted to click on something underneath the menu

- This can be solved with the pointer-events CSS property

# Pointer-events pt2

- You can use the css pointer-events property to stop the mouse interacting with an element
  - Pointer-events: none stops the element being clickable/hover able with the mouse
  - Pointer-events: auto makes the element act like normal

- This can be applied to the menu so that it is not interactive while hidden, but interactive while shown

# Pointer-events example

```css
nav ul ul {
    position: absolute;
    display: block;
    background-color: green;
    width: 100%;
    padding: 0;
    opacity: 0;
    transition: opacity 0.5s ease-in;
    pointer-events: none; /* Sub menus cannot be interacted with */
}

nav>ul>li:hover ul {
    opacity: 1;
    pointer-events: auto; /* Sub menus can be interacted with */
}
```

# Slide effect

- Another option for a menu animation is a slide effect
- The first thing we need is a the before state
- To move the menu up, the transform property can be used
  - This is also possible using one or more of the following properties
    - margin-top
    - Top

- **If you find a tutorial using anything other than transform for this it is very out of date!**

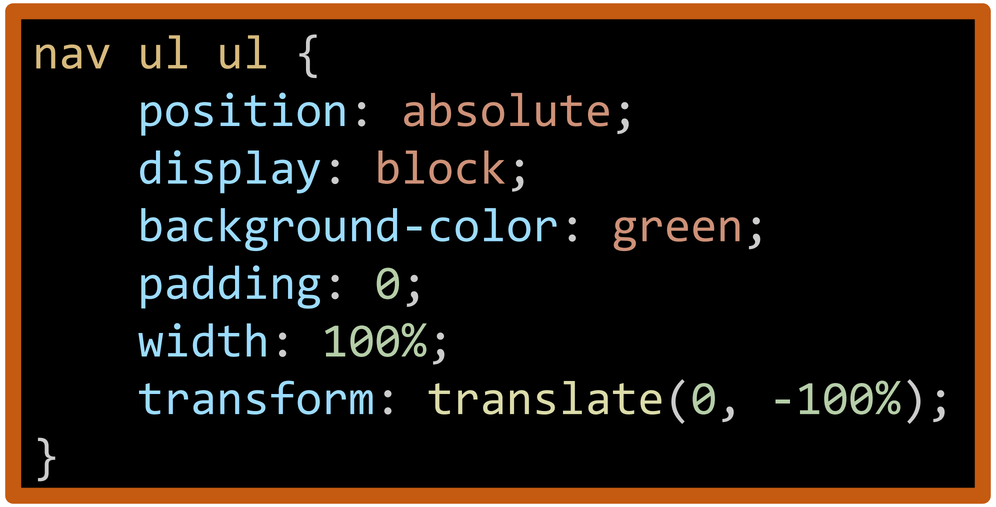
# Transform

- The transform property can be used to perform a two dimensional transformation on the menu
- There are lots of properties available
  - Rotate
  - Skew
  - Scale
  - Translate

- The one we are interested in is translate which is a way of moving where the element is drawn on the screen

# Translate

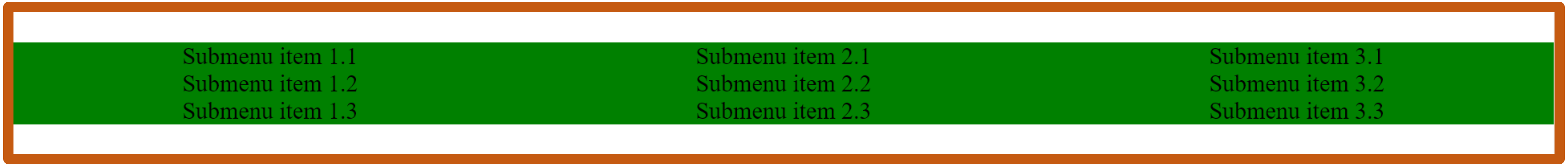
- You can move the submenus up so they are displayed on the same row as the main menu using this code
- The translate(0, -100%) tells the browser to draw the element 0px from its original left position and -100% from its original y position



```
nav ul ul {
    position: absolute;
    display: block;
    background-color: green;
    padding: 0;
    width: 100%;
    transform: translate(0, -100%);
}
```

# Issue with translate

- But wait.. there’s a problem

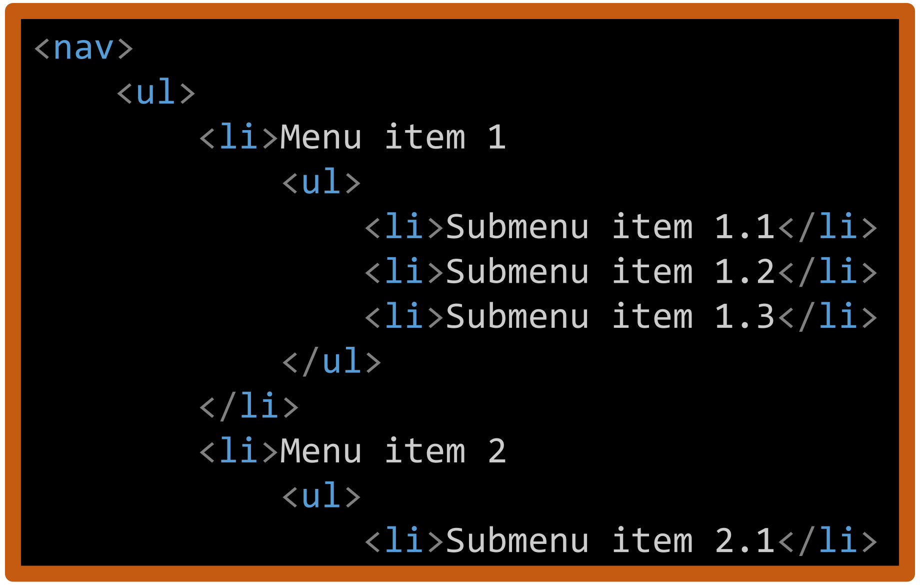


| | | |
|---|---|---|
| Submenu item 1.1 | Submenu item 2.1 | Submenu item 3.1 |
| Submenu item 1.2 | Submenu item 2.2 | Submenu item 3.2 |
| Submenu item 1.3 | Submenu item 2.3 | Submenu item 3.3 |

- Assuming we wanted to slide the submenus down to display them, they are being drawn in the correct place but over the top of the yellow menu bar
- We only want them visible on mouse over

# Z index

- Normally elements are stacked on top of each other based on the order they appear in the HTML file
- Elements lower down the HTML file will appear on top of elements higher up



```
<nav>
    <ul>
        <li>Menu item 1
            <ul>
                <li>Submenu item 1.1</li>
                <li>Submenu item 1.2</li>
                <li>Submenu item 1.3</li>
            </ul>
        </li>
        <li>Menu item 2
            <ul>
                <li>Submenu item 2.1</li>
```

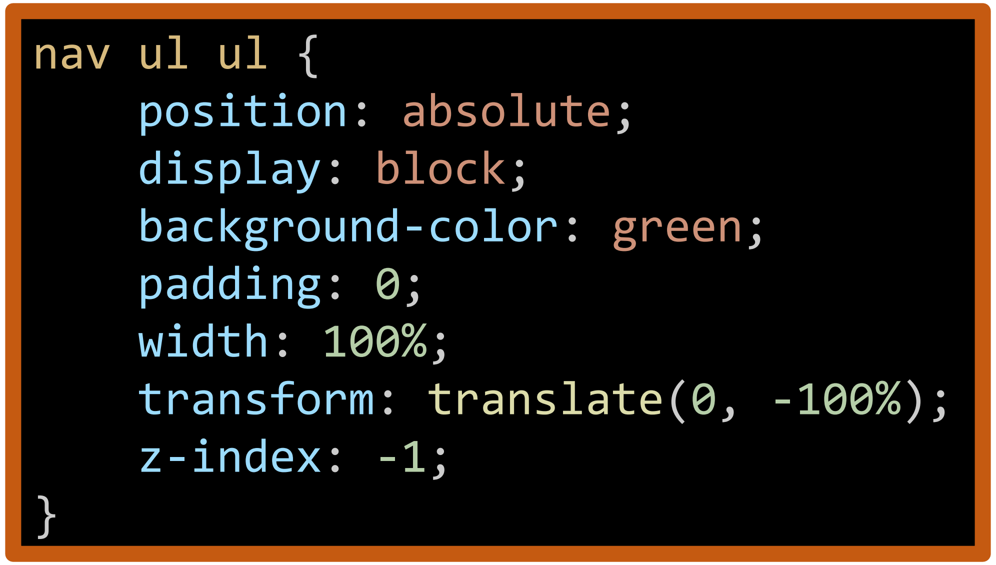
# Z index pt2

- Because the yellow menu (top level <ul>) is before the green submenus in the HTML file, the submenus are drawn above the yellow <ul>
- Normally this isn’t a problem or even something we’d ever notice because they wouldn’t overlap
- To fix this, you can use z-index

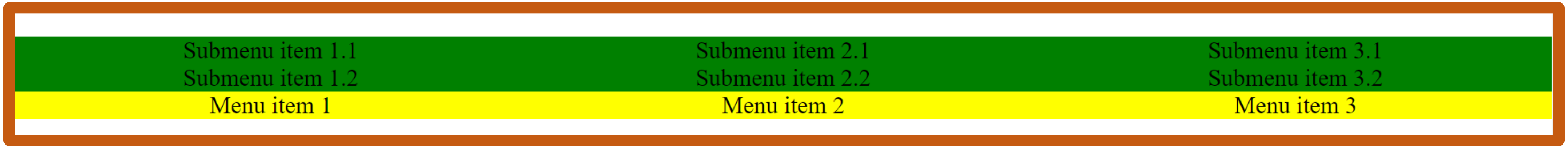
# Z index pt3

- The z-index attribute can be used to describe the order in which elements will stack in front of one another
- You can set specific values, however, if you just wish to move an element behind its parent you can use
  - z-index: -1

# Z index example



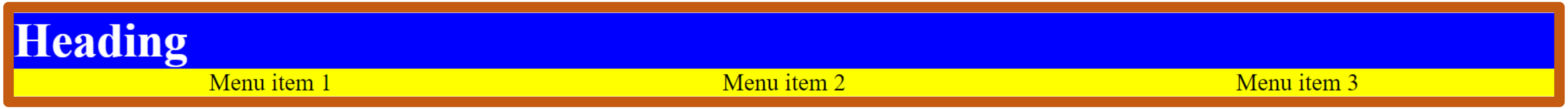
```
nav ul ul {
    position: absolute;
    display: block;
    background-color: green;
    padding: 0;
    width: 100%;
    transform: translate(0, -100%);
    z-index: -1;
}
```



```
Submenu item 1.1        Submenu item 2.1        Submenu item 3.1
Submenu item 1.2        Submenu item 2.2        Submenu item 3.2
  Menu item 1             Menu item 2             Menu item 3
```

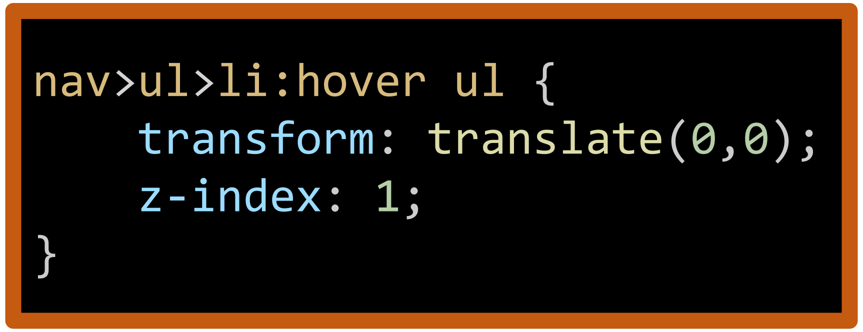
# Z index pt4

- The submenus have been moved behind the yellow menu bar
- However, they are still appearing
- Normally there would be content on the screen at that point so let’s add a <header> element like on our test page
- The submenus are still there, they’re just behind the yellow and blue elements now



# Slide hover effect – after state

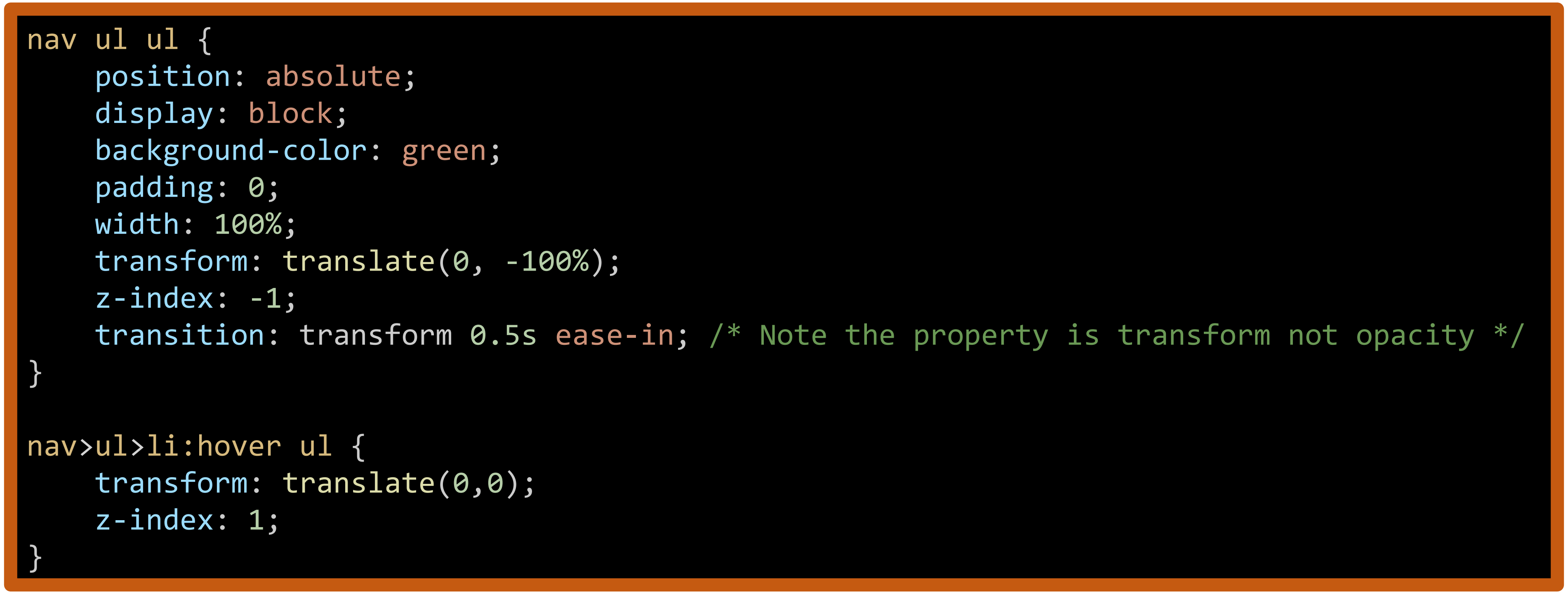
- To add the after state the menus can be translated back to 0,0 (their starting position) when the hover effect runs



```
nav>ul>li:hover ul {
    transform: translate(0,0);
    z-index: 1;
}
```

# Slide hover effect

- To animate the effect, you apply the transition property to transition transform changes



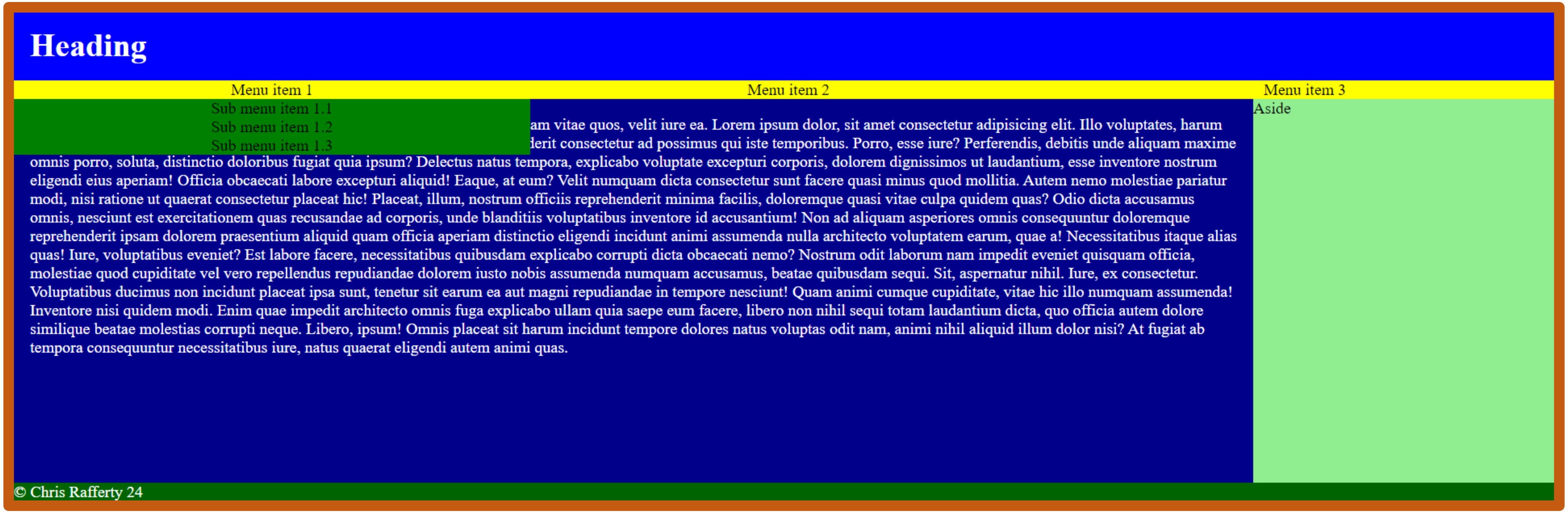
```
nav ul ul {
    position: absolute;
    display: block;
    background-color: green;
    padding: 0;
    width: 100%;
    transform: translate(0, -100%);
    z-index: -1;
    transition: transform 0.5s ease-in; /* Note the property is transform not opacity */
}

nav>ul>li:hover ul {
    transform: translate(0,0);
    z-index: 1;
}
```

# Exercise

- Try adding the submenus to your grid websites. They should only be visible when you hover your mouse over them Slide 8
  - Be sure to use :hover

- **Don’t use JavaScript for this in the assignment**

# Exercise 1 example



am vitae quos, velit iure ea. Lorem ipsum dolor, sit amet consectetur adipisicing elit. Illo voluptates, harum
derit consectetur ad possimus qui iste temporibus. Porro, esse iure? Perferendis, debitis unde aliquam maxime omnis porro, soluta, distinctio doloribus fugiat quia ipsum? Delectus natus tempora, explicabo voluptate excepturi corporis, dolorem dignissimos ut laudantium, esse inventore nostrum eligendi eius aperiam! Officia obcaecati labore excepturi aliquid! Eaque, at eum? Velit numquam dicta consectetur sunt facere quasi minus quod mollitia. Autem nemo molestiae pariatur modi, nisi ratione ut quaerat consectetur placeat hic! Placeat, illum, nostrum officiis reprehenderit minima facilis, doloremque quasi vitae culpa quidem quas? Odio dicta accusamus omnis, nesciunt est exercitationem quas recusandae ad corporis, unde blanditiis voluptatibus inventore id accusantium! Non ad aliquam asperiores omnis consequuntur doloremque reprehenderit ipsam dolorem praesentium aliquid quam officia aperiam distinctio eligendi incidunt animi assumenda nulla architecto voluptatem earum, quae a! Necessitatibus itaque alias quas! Iure, voluptatibus eveniet? Est labore facere, necessitatibus quibusdam explicabo corrupti dicta obcaecati nemo? Nostrum odit laborum nam impedit eveniet quisquam officia, molestiae quod cupiditate vel vero repellendus repudiandae dolorem iusto nobis assumenda numquam accusamus, beatae quibusdam sequi. Sit, aspernatur nihil. Iure, ex consectetur. Voluptatibus ducimus non incidunt placeat ipsa sunt, tenetur sit earum ea aut magni repudiandae in tempore nesciunt! Quam animi cumque cupiditate, vitae hic illo numquam assumenda! Inventore nisi quidem modi. Enim quae impedit architecto omnis fuga explicabo ullam quia saepe eum facere, libero non nihil sequi totam laudantium dicta, quo officia autem dolore similique beatae molestias corrupti neque. Libero, ipsum! Omnis placeat sit harum incidunt tempore dolores natus voluptas odit nam, animi nihil aliquid illum dolor nisi? At fugiat ab tempora consequuntur necessitatibus iure, natus quaerat eligendi autem animi quas.

© Chris Rafferty 24

# Exercise 2

- Try adding a fading submenu to your work
- Change the fading menu to a sliding menu
- Try out the following transforms
    - Before hover: transform: skewX(90deg);
    - After hover: transform: skewX(0deg);

    - Before hover: transform: rotateY(90deg)
    - After hover: transform: rotateY(0deg)

# Exercise 3

- Add transform-origin: bottom right;
- Before hover: transform: rotate(90deg) translate(-50vw);
- After hover: transform: none;
- Play around with different transform effects
  - [https://developer.mozilla.org/en-US/docs/Web/CSS/transform](https://developer.mozilla.org/en-US/docs/Web/CSS/transform)
- See what interesting menu animations you can produce

# Exercise 4

- Use transitions to slide in your mobile menu in an interesting way when the hamburger icon is pressed
- Remember, you can apply the transform inside the :target selector instead of the :hover selector!

- The assignment is looking for at least a fade or slide animation somewhere in your work
  - Drop-down navigation
  - Mobile sliding navigation
  - Etc.

# Animation in CSS

- There are two types of animations available in CSS
  - Transitions
  - Keyframes

# Transitions

- We just used transitions for the sub menu navigation
- Transitions are the simple kind of animation

# Transitions explained

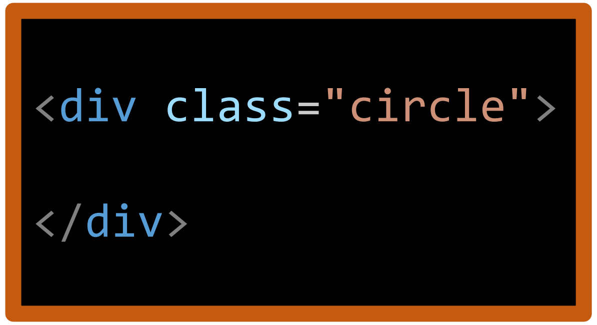
- Transitions allow you to create a two state animation
  - Before
  - After
- Transitions can be applied to any property that has a numerical range of values ( x,y screen co-ordinates, opacity, border size, etc)
- Transitions are limited to this use-case
- If your animation requires more than two states (before and after) you cannot achieve this with transitions

# Keyframe animations

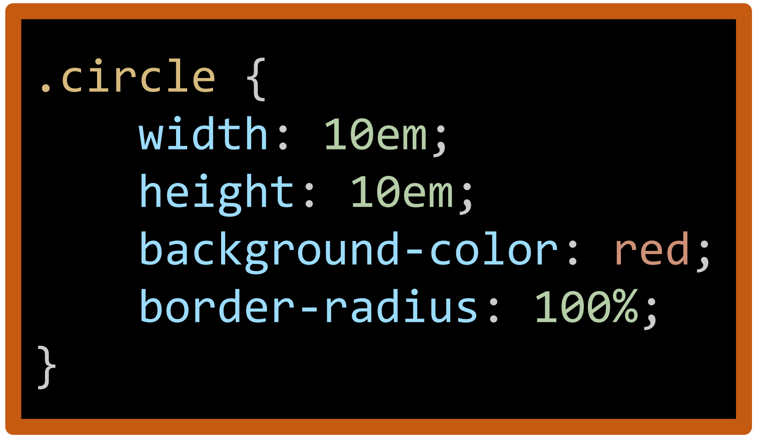
- Keyframe animations allow you to describe an animation with as many different steps as you like
- The syntax is more complex than transitions because there is more to do
- There are two parts to defining a keyframe animation
    - Describe the steps of the animation
    - Apply the animation to an element on the page

# Creating a circle

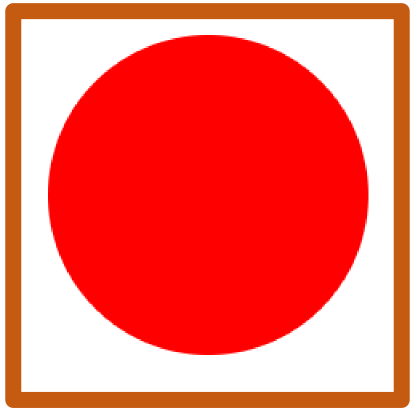
- Lets say we have a red ball that we want to move across the screen
  - First we need to create it in the HTML
  - And style it through CSS (you can use border-radius: 100% to make a circle)



```
<div class="circle">

</div>
```

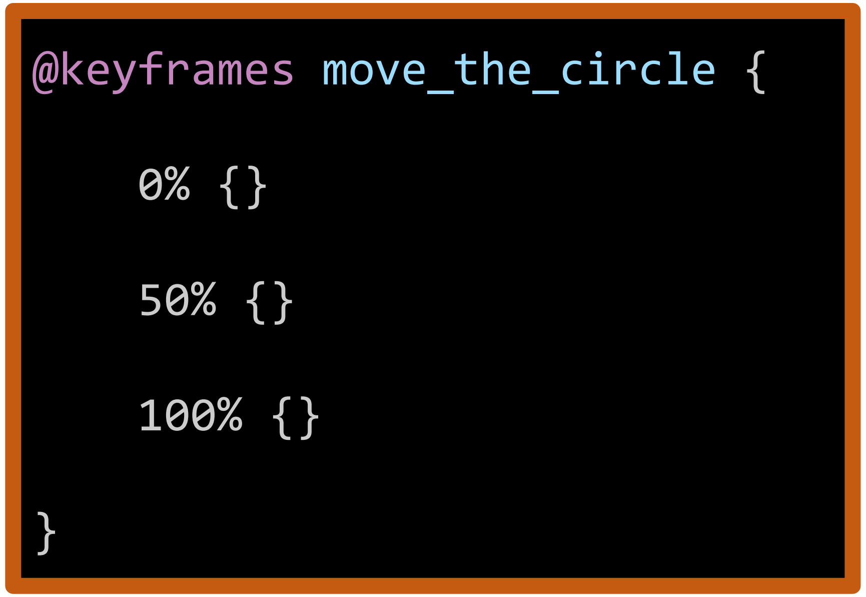


```
.circle {
    width: 10em;
    height: 10em;
    background-color: red;
    border-radius: 100%;
}
```



# Keyframe template

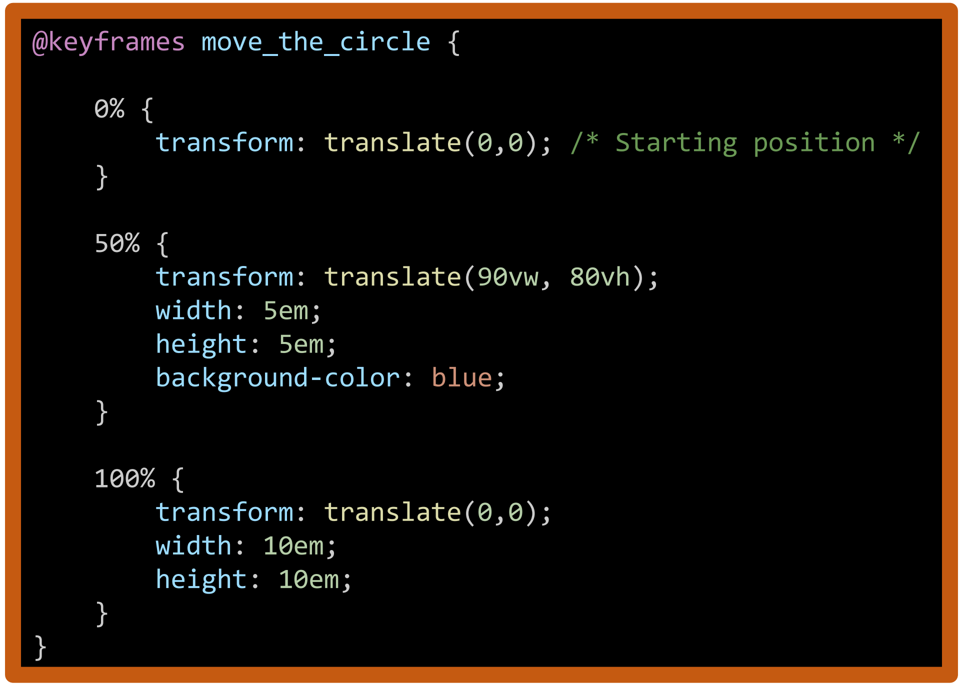
- Once we have the element created and styled, we need to animate it
- First we create the keyframe using @keyframes NAME {}
- We then need to specify the number of steps in the animation, in the example below we have
    - 0% (the start of the animation)
    - 50% (the middle of the animation)
    - 100% (the end of the animation)



```
@keyframes move_the_circle {

    0% {}

    50% {}

    100% {}

}
```

# Adding styles to the keyframe

- The next step is to specify what happens at each stage of the animation
  - We can move the circle using transform: translate
  - You can apply most CSS properties like changing the background colour or size



```
@keyframes move_the_circle {

    0% {
        transform: translate(0,0); /* Starting position */
    }

    50% {
        transform: translate(90vw, 80vh);
        width: 5em;
        height: 5em;
        background-color: blue;
    }

    100% {
        transform: translate(0,0);
        width: 10em;
        height: 10em;
    }
}
```

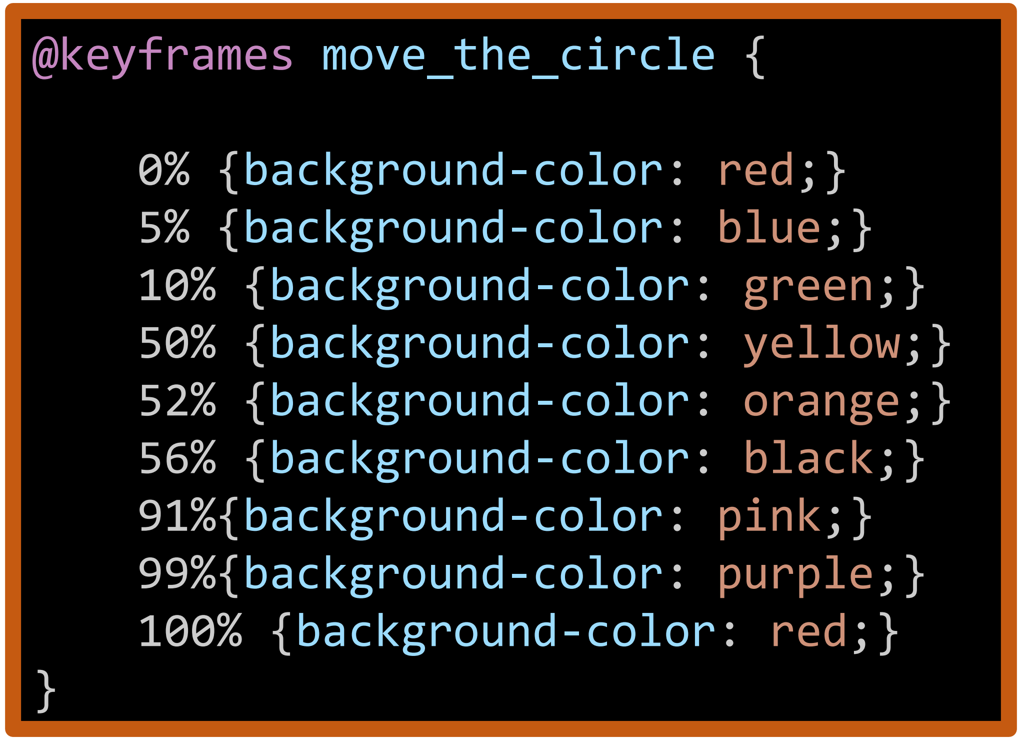
# Applying the animation

- To apply the animation to the circle we use the animation property
  - It is used in the same way as transition
- You need to include the name of the animation and the duration

```
.circle {
    width: 10em;
    height: 10em;
    background-color: red;
    border-radius: 100%;

    animation: move_the_circle 10s infinite;
}
```

# Multiple steps

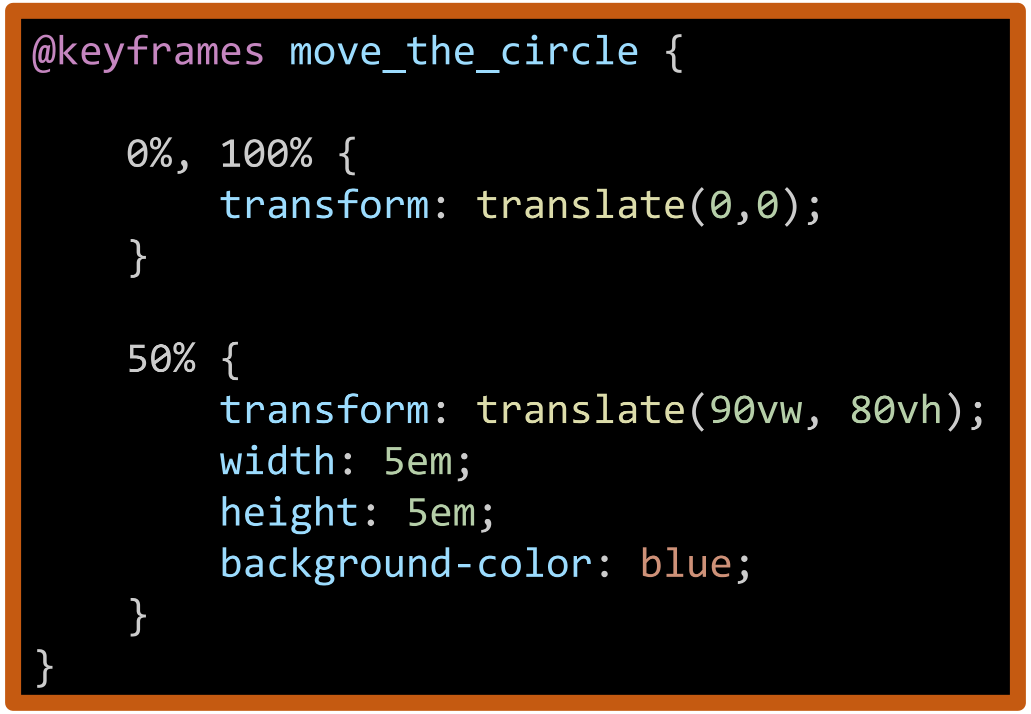
- You can have as many step as you like in the animation (0%, 10%, 25%, 50%, 70%, 80%, 100%)



```
@keyframes move_the_circle {

    0% {background-color: red;}
    5% {background-color: blue;}
    10% {background-color: green;}
    50% {background-color: yellow;}
    52% {background-color: orange;}
    56% {background-color: black;}
    91%{background-color: pink;}
    99%{background-color: purple;}
    100% {background-color: red;}
}
```

# Combining steps

- You can combine steps of the animation using a comma
- 0%,100% (the same CSS)



```
@keyframes move_the_circle {

    0%, 100% {
        transform: translate(0,0);
    }

    50% {
        transform: translate(90vw, 80vh);
        width: 5em;
        height: 5em;
        background-color: blue;
    }
}
```

# Keyframe uses

- You can apply keyframe animations to most HTML elements
  - You could change the background colour of the body
  - Have an image grow and shrink every 10 seconds
  - Create a matrix inspired background by moving tags around the screen

- Keyframe animations are useful as unlike transitions which only have a start and end we can specify each step of the animation (start, end and everything in-between)

# Exercise 5

- Create an animation that moves an element from the left of the screen to the right of the screen and back again repeatedly

- Create an animation that moves an element as follows
  - Top right
  - Bottom left (diagonally)
  - Bottom right
  - Top left (diagonally)
  - Top right

# Exercise 6

- There are many different effects which can be achieved with CSS
- Here are a few to look through
  - 3d flip https://webdevtrick.com/css-flip-effect-on-hover/
  - Animated text https://www.legendblogs.com/how-to-create-3d-animated-text-with-css
  - 3d cube: https://redstapler.co/pure-css-3d-cube-effect-tutorial/
- Non-square shapes
  - Triangles: https://medium.com/@codingdudecom/css-triangle-81eb02dffeee
  - Pentagons , octagons: https://coursesweb.net/css/polygons-css_cs

# Useful links

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_transitions/Using_CSS_transitions
- https://developer.mozilla.org/en-US/docs/Web/CSS/opacity
- https://developer.mozilla.org/en-US/docs/Web/CSS/transform
- https://developer.mozilla.org/en-US/docs/Web/CSS/z-index
- https://developer.mozilla.org/en-US/docs/Web/CSS/@keyframes