CSY1018
Web Programming

Topic 5

thomas.butler@northampton.ac.uk

Topic 5

- Mouse over effects with :hover
- Creating animations with transitions

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 on, several interactive elements were added that allowed *state changes*
- One of these state changes is the :target selector we saw 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="#">Click here</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 mouse-over menu
  - (Not to be confused with the navigation menu for mobile devices)

Nesting with :hover

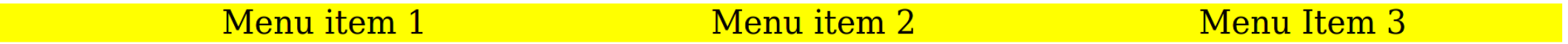
- You can use this to create a navigation bar with mouse over drop downs
- Firstly, you need some HTML for the menu

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

Menu bar

- Next, some styling to make it go across the page:

```
nav {
  background-color: yellow;
}
nav ul {
  list-style-type: none;
  display: grid;
  grid-template-columns: 33% 33% 33%;
  text-align: center;
}
```



:hover for submenus

- A submenu can be added with a secondary <ul> tag for each menu
- Note the location in the HTML code, it is inside the primary menu's <li> tag

```
<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</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 1.1 Submenu item 1.2 Submenu item 1.3 Submenu item 2.1 Submenu item 2.2 Submenu item 2.3

Positioning the submenus

- 1. Set the containing list item to position: relative

```
nav > ul > li {
  position: relative;
}
```

  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!

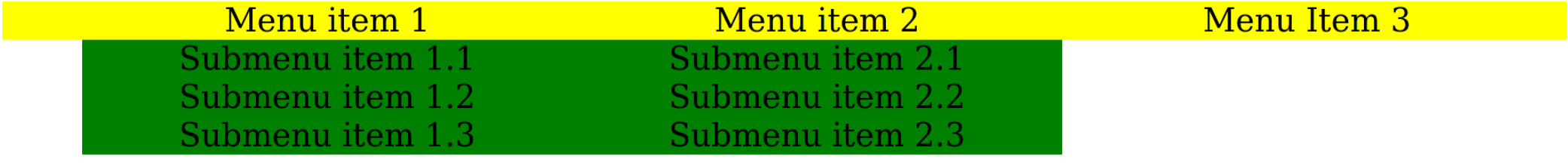
- Note the direct descendant operator. This applies the style to only the first level of <li> elements

Style the submenus

- 2. 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;
  padding: 0;
  width: 100%;

}
```



Using :hover to toggle the menu

- At the moment 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

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

```
nav ul ul {
  position: absolute;
  display: none;
  background-color: green;
  padding: 0;
  width: 100%;

}
```

Menu item 1 Menu item 2 Menu Item 3

Add the hover effect

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

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

Exercise 1

- 1. Try adding submenus to the page you created last week. They should only be visible when you hover your mouse over them.
  - Be sure to use :hover
- Note: You *do not * need Javascript to do this
- 2. See what happens on the mobile version of your site
  - Make sure you use developer tools to preview it as if it was a phone!

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

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

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

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

Fade effect

- 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

```
nav ul ul {
  position: absolute;
  background-color: green;
  padding: 0;
  display: block;
  width: 100%;
  opacity: 0;
  transition: opacity 0.5s ease-in;
}

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

Transitions

- 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

- You can use the css pointer-events property to stop the mouse interacting with an element
  - Pointer-events: none stops the element being clickable/hoverable 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

```css
nav ul ul {
  position: absolute;
  background-color: green;
  padding: 0;
  display: block;
  width: 100%;
  opacity: 0;
  transition: opacity 0.5s ease-in;
  pointer-events: none;
}

nav > ul > li:hover ul {
  opacity: 1;
  pointer-events: auto;
}
```

Slide effect

- Another option for a menu animation is a slide effect
- The first thing we need is the *before* state
- To move the menu up, the *transform* property can be used
  - **Note: 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

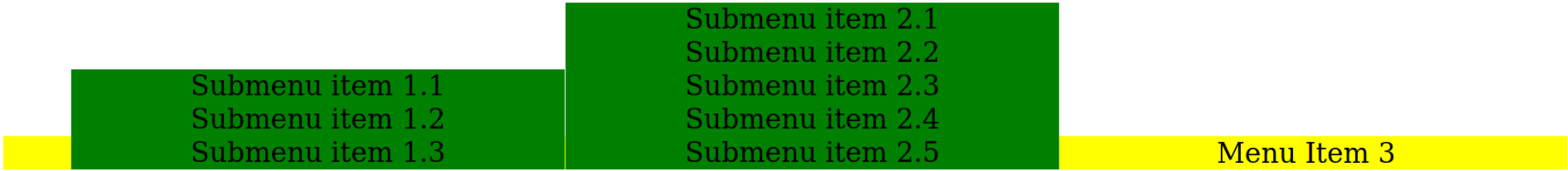
Translate

- 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
- (Remember the container has position: relative, so all these are relative to the containing menu!)

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

Translate

- But wait.. there’s a problem:



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>
        <li>Submenu item 2.2</li>
        <li>Submenu item 2.3</li>
      </ul>
    </li>
    <li>Menu Item 3</li>
  </ul>
</nav>
```

Z Index

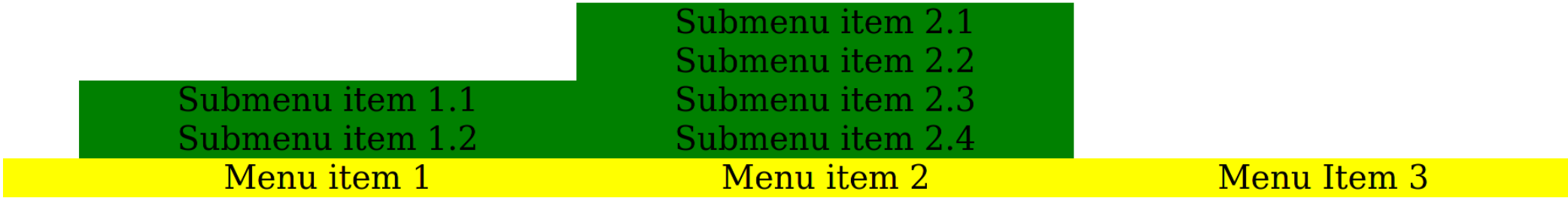
- Because the yellow menu (top level <ul>) is *before* the green submenus in the HTML file, the submenus are drawn *above* the yellow
- Normally this isn’t a problem or even something we’d ever notice because they wouldn’t overlap
- To fix this, you can use z-index

z-index

- 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

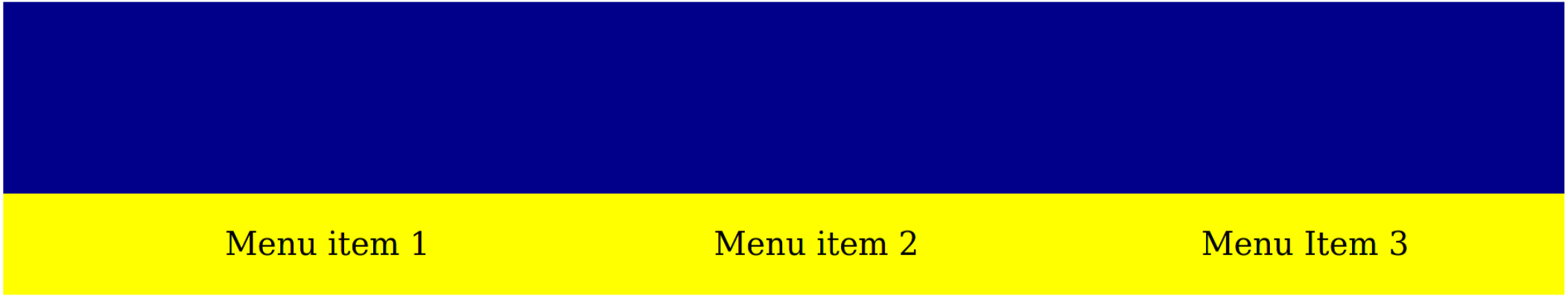
z-index

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



z-index

- The submenus have been move 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



Menu hover effect

- 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);
}
```

Animating

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

```
nav ul ul {
  position: absolute;
  background-color: green;
  padding: 0;
  display: block;
  width: 100%;
  transform: translate(0, -100%);
  transition: transform 0.5s ease-in;
  z-index: -1;
}

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

Exercise 2

- 1. Try adding a fading menu submenu to your work from the previous week
- 2. Amend the fade to a slide
- 3. Test 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 2

- Add transform-origin: bottom right;
- Before hover: transform: rotate(90deg) translate(-50vw);
- After hover: transform: none

Exercise 3

- Play around with different transform effects
-
- https://developer.mozilla.org/en-US/docs/Web/CSS/transform

See what interesting menu animations you can come up with

Exercise 4

- Use transitions to slide in your mobile menu from last week 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!