MEDS2007: WEBSITE

LECTURE 4

Dave Everitt deveritt@dmu.ac.uk

THE TOPICS

- CSS animations
- Web Typography exercise
- introducing JavaScript

CSS ANIMATION: 1

COMPONENTS

- **transitions** (things to change, over *time*)
- **transformations** (shape-shifting, *no* timing)
- **animation** (composed *timeline* keyframes)

You can use these separately or together.

CSS ANIMATION: 2

TRANSITION: EXAMPLE 1

- transition goes in the **element's style** (the `div`)
- things to change go in the element's `:hover` style

```
div {
        width: 100px;
        background: red;
        transition: all 2s;
}
div:hover {
        width: 400px;
        background: blue;
}
```

W3Schools: CSS3 transition Property

CSS ANIMATION: 3

TRANSITION: EXAMPLE 2

The **trigger** can be **something other** than :hover:

```
input[type=text] {
    width: 100px;
    transition: all .35s ease-in-out;
}
input[type=text]:focus {
    width: 250px;
    border: 2px solid #0f0;
}
```

Early article on `transition`, Dan Cederholm, 2010

CSS ANIMATION: 4

TRANSITION

Changes CSS properties over a **given time** - three examples:

```
transition: all 2s;
transition: color 2s 1s; /* add 1s delay */
transition: color, width 2s ease-in-out 1s;
```

- **property**: the CSS to transition (comma-separated, all)
- **duration**: transition length (secs/m'secs `s`|`ms`, `infinite`)
- **timing-function**: (optional) 'speed curve' of the transition
- **delay**: (optional) time before transition starts

timing-function easing examples and demo

CSS ANIMATION: 5

TRANSFORM

```
transform: rotate(70deg);
```

Can be used alone (not animated) or with `transition` on an element, then `transform` on :hover/another action.

The **values** of the `transform` property are:

`translate(540px,-200px);` x, y
`rotate(360deg);`
`scale(2, 0.5);` x, y
`skew(10deg);` x axis
`matrix(1, -0.3, 0, 1, 0, 0)`
scaleX, skewY, skewX, scaleY, translateX, translateY

CSS ANIMATION: 6

TRANSFORM REFERENCES, TUTORIALS

- W3Schools reference: CSS3 transform Property
- W3Schools tutorial: CSS3 2D Transforms
- W3Schools: CSS3 3D Transforms

CSS ANIMATION: 7

ANIMATION

The CSS `animation` property binds a custom `animation` to an element identified by it's `selector` e.g.

```
.see-this {
  animation: change-stuff 5s infinite;
}
```

(animation name, duration, how many times to run)
`infinite` runs forever, seconds/milliseconds limit time

The changes are user-defined in a separate `@keyframes` code block…

CSS ANIMATION: 8

ANIMATION KEYFRAMES

```
@keyframes change-stuff {
    25% {
        styles for 1/4 way through;
    }
    50% {
        styles for 1/2 way through;
    }
    75% {
        styles for 3/4 way through;
    }
}
```

Download the code for this demo animation or view it here
(See W3Schools CSS3 Animations)

CSS ANIMATION: 9

ANIMATION WHAT IS (NOT) ANIMATABLE?

not `display: none` or `display: block` etc.

Animatable CSS properties (W3Schools)

triggered by focus/hover/etc.... that can be styled e.g.:

```
:focus :blur :hover :active
input[type=checkbox]:checked …
```

CSS ANIMATION: 10

SOME EXAMPLES

- A usable CSS-only slideshow
- Example: animated movie posters
- Original Hover Effects
- Chessboard vacuum (Ana Tudor)
- Bending effect - page flip (Fabrizio Bianchi)
- Rotating colour box (CodePen)

WEB TYPOGRAPHY

A TUTORIAL

This will help you understand how to control type on the web.

You can download the typography tutorial files here.

JAVASCRIPT: 01

HOW IS JAVASCRIPT USED?

- form **validation** of user input
- handling **data** (e.g. JSON)
- interactive **user interfaces** and feedback
- web-based **apps** (mobile and desktop)
- web and mobile **frameworks**
- **applications** built on web technologies
- **games** with HTML5 canvas (HTML5 Games website)
- **3D animation** with canvas and WebGL (F1 car, Aquarium)

JAVASCRIPT: 02

JAVASCRIPT HISTORY

- **1995:** JavaScript created by Brendan Eich as Mocha/LiveScript
- **1997:** ECMAScript standard established (see ECMA)
- **1999:** ES3 - Microsoft discover www. IE5 everywhere.
- **2000–2005:** XMLHttpRequest (AJAX): Outlook Web Access (2000), Gmail (2004), Google Maps (2005)
- **2009:** ES5 - established JavaScript, and standard JSON
- **2015:** ES6/ECMAScript2015 - mostly syntactic sugar, because no firm agreement on anything more radical

Condensed from benmccormick.org

JAVASCRIPT: 03

THE BROWSER EVENT LOOP

In a browser, everything runs in an **event loop** so you use **event handlers** to **listen** for **events** triggered by the **user** or other **processes**:

```
myElement.addEventListener("click", do_stuff);
```

While CSS can be triggered by focus/hover/etc., JavaScript can listen for **hundreds** of events. See:
Web Event reference (MDN)

JAVASCRIPT: 04

SOME EXAMPLES

The simplest possible example:

```
<p id="my-id">Click me!</p>
```

```
const clickThing = document.getElementById("my-id");

clickThing.addEventListener("click", do_stuff);

function do_stuff() {
  alert("You clicked me!");
}
```

JAVASCRIPT: 05

What's the date?

I've no idea. Click the button.

RESOURCES:

W3Schools: JavaScript HTML DOM EventListener
W3Schools: JavaScript toDateString() Method

JAVASCRIPT: 05A

The code that showed the date:

HTML:

```
<button id="myBtn">What's the date?</button>

<p id="demo">I've no idea. Click the button.</p>
```

JavaScript:

```
document.getElementById("myBtn").addEventListener("click",

function displayDate() {
  let d = new Date();
  document.getElementById("demo").innerHTML = `Oh, it's: $
}
```

JAVASCRIPT: 06

Hello, anonymous user! What's your name?

Enter name: name here please :-) Submit

ADVANCED RESOURCES:

(Code is too long to show here)

A simple introduction to JavaScript and the DOM
A simple introduction to JavaScript and Local Storage

JAVASCRIPT: 07

JAVASCRIPT OBJECTS: 1

A JavaScript object contains a series of **colon-separated name-value pairs** (no comma after the last one!!) enclosed in curly braces {}:

```
var car = {
  make: "Tesla",
  model: "Model S 100D",
  range: "335"
};
```

You can have an **array of objects**, which is similar to a common data format called **JSON**, often used for online data sources.

JAVASCRIPT: 08

JAVASCRIPT OBJECTS: 2

How many miles will the Model S travel?

JAVASCRIPT: 08 CODE

HTML and JavaScript:

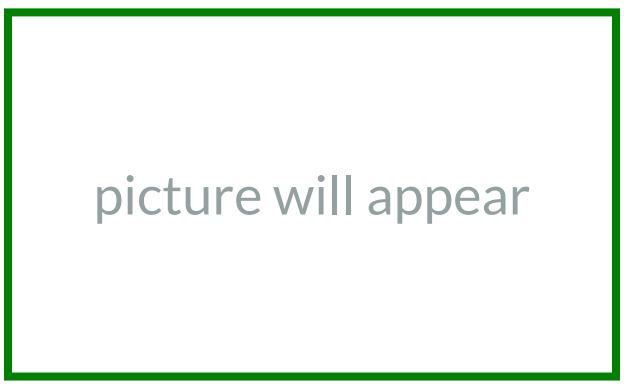
```
<p id="question">How many miles will the Model S travel?</p>

<p id="answer"></p>
```

```
<script>
  const question = document.getElementById("question");
  const answer = document.getElementById("answer");

  let info = function() {
    answer.innerHTML = `The <strong>${car.make} ${car.model}</strong> ha
  }

  question.addEventListener("click", info);
</script>
```

JAVASCRIPT: 09

JAVASCRIPT OBJECTS: 3

Show me a photo of: the Tesla Model S OR a kitten!



JAVASCRIPT: 09 CODE

CSS:

```
#picture {
  font-size: 75%;
  text-align: center;
  line-height: 178px;
  height: 178px;
  width: 300px;
  background-repeat: no-repeat;
  border: solid green 4px;
  transition: all 1s;
}
#picture.car {
  border-color: silver;
  background-image: url("images/tesla-model-s.jpg");
}
#picture.cat {
  border-color: pink;
  background-image: url("images/kitten.jpg");
}
```

JAVASCRIPT: 09 CODE

HTML and JavaScript:

```
<p>Show me a photo of: <span id="photo1">the Tesla Model S</span> OR <sp

<p id="picture" class="text">picture will appear</p>
```

```
const picture = document.getElementById("picture");
const photo1 = document.getElementById("photo1");
const photo2 = document.getElementById("photo2");

function showImage(whichPic) {
  picture.innerHTML = "";
  if (whichPic === "photo1") {
    picture.classList.remove("cat");
    picture.classList.add("car");
  } else {
    picture.classList.remove("car");
    picture.classList.add("cat");
  }
}

photo1.addEventListener("click", function(){ showImage(this.id) });
photo2.addEventListener("click", showImage);
```

FRONT-END WEB TECHNOLOGIES

There are only **three**:

HTML – CSS – JAVASCRIPT