Lecture 13

Dynamic UI Implementation

UNIVERSITY OF AUCKLAND

COMPSCI 345 / SOFTENG 350

Dr. Gerald Weber

Zoom:
https://auckland.zoom.us/j/98929090157?pwd=NnRuWEZzcVhFMGt3cE9LR2VoVGhJQT09

Learning Objectives

- Understand the role of JavaScript in dynamic UI implementation.
- Being able to perform simple dynamic adaptations in JavaScript and access the HTML page content.
- Understand how one can write JavaScript code that is bounded in execution time in a way that reduces performance issues.

JavaScript

- A client-side interpreted scripting language since 1995.
- Has a high degree of standardization as ECMAscript, which is a proper ISO standard (ISO/IEC 22275:2018)
- Great efficiency gains through better just–in-time compilation approaches.
- Remains a language that is fully integrated in web technology.
- C-style syntax
- Strong foothold.

JavaScript in the HTML page

- JavaScript can be written directly in HTML within the <script> element.
- Everything within this element is code and is interpreted as a sequence of commands which are executed immediately.
- Programs are commands that store code for later.

```
<body>
  <div>
    <div>1</div>
    <div>2</div>
    <div>3</div>
  </div>
  <div>
    <div>5</div>
    <div>6</div>
  </div>
  <script
   console.log('hithere');
  </script>
</body>
```

JavaScript Object model

- In JavaScript, values of variables are objects, some immutable, but usually changeable.
- A JavaScript object has an opaque identity (expected to be realized as pointer, can be compared but opaque means the address cannot be obtained as number).
- A powerful feature of JavaScript objects is that fields can be added at any time.

```
<body>
  <script
    var myobj = {};
    myobj.name = 'Pat';
    myobj.code = 123;
  </script>
</body>
```

Developing JavaScript

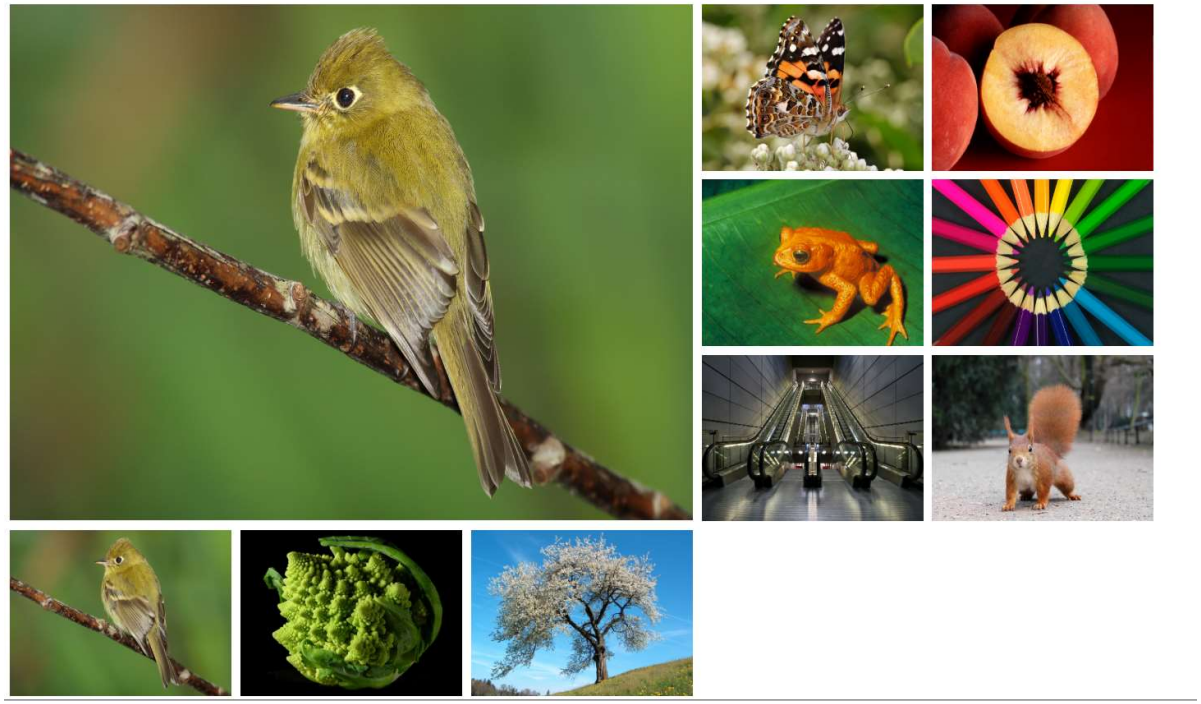
- JavaScript can be executed directly in the browser
- Modern browsers offers rich features comparable to a development environment as developer tools

  - Move to compiled approaches such as TypeScript in a project should be carefully considered.

- Many online development environments that also allow to showcase work: JSFiddle
- Some more oriented to online sharing of small solutions
- Some full development environments.

Resizeable Gallery

Resizeable Gallery



Resizeable Gallery

Resizeable Gallery

- Changing the main image in a gallery view.

Changing the gallery main image

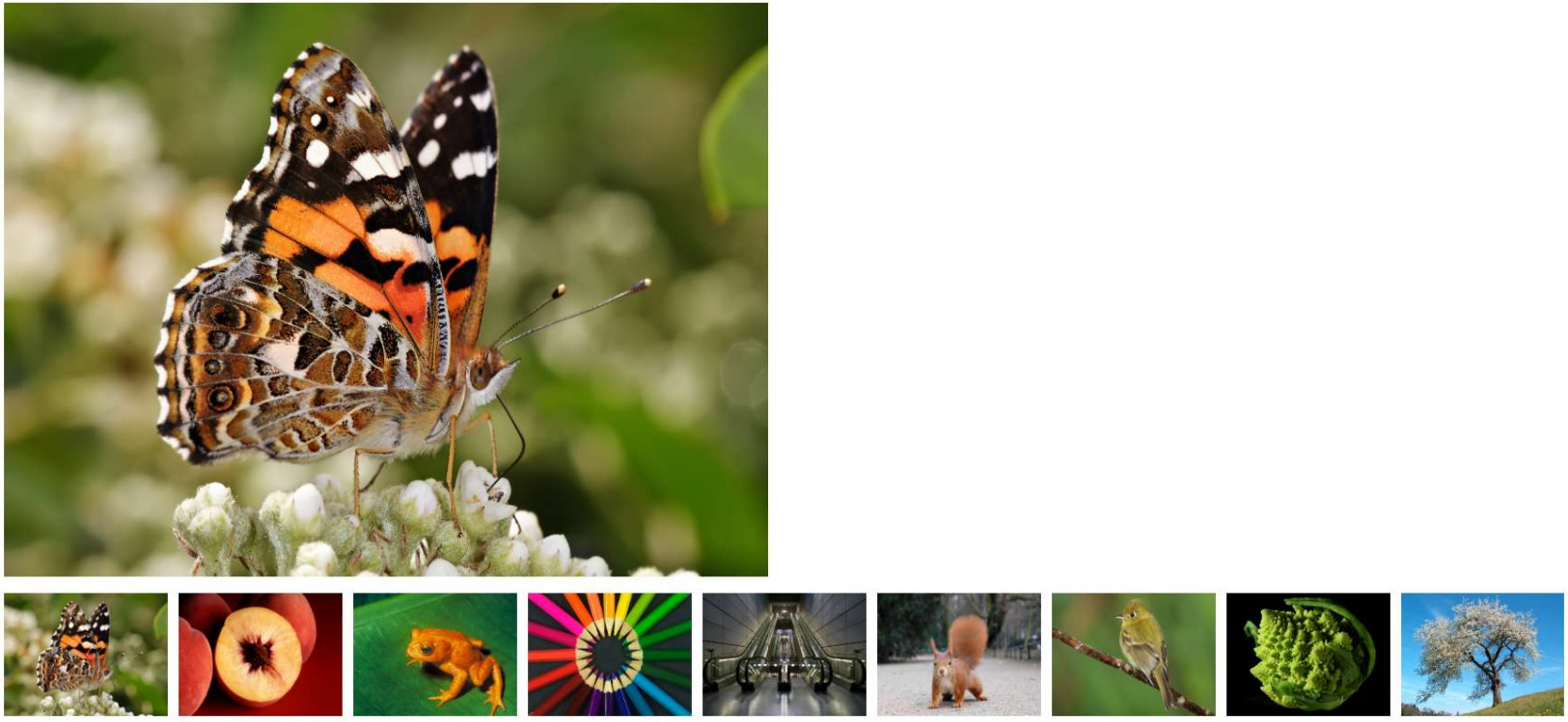
```
img {
  height:465px;
  width:616px;
  border: solid
     4px
    transparent;
  float: left;
}

img.thumb {
  height:150px;
  width:200px;
}
```

```
<link rel=stylesheet type="text/css"
href="main01.css">
<body>
<img id="fullview"  src="full/01.jpg">
<img class="thumb"
onclick='changeMainImg(this);'
src="full/01.jpg">
. . .
<img class="thumb"
onclick='changeMainImg(this);'
src="full/09.jpg">

<script >
   function changeMainImg(that) {
    document.getElementById("fullview").src=
   that.src;
}
</script>
```

Flexbox



Flexbox









Flexbox

Flexbox

main.css

```
.flex-container {
  display:
        flex;
}
```

```
<link rel=stylesheet type="text/css"
href="main.css">
<body>
<img id="fullview"  src="full/01.jpg">
<div class="flex-container">
<img class="thumb"
onclick='changeMainImg(this);'
src="full/01.jpg">
. . .
<img class="thumb"
onclick='changeMainImg(this);'
src="full/09.jpg">
</div>
<script >
   function changeMainImg(that) {
    document.getElementById("fullview").src=
   that.src;
}
</script>
```

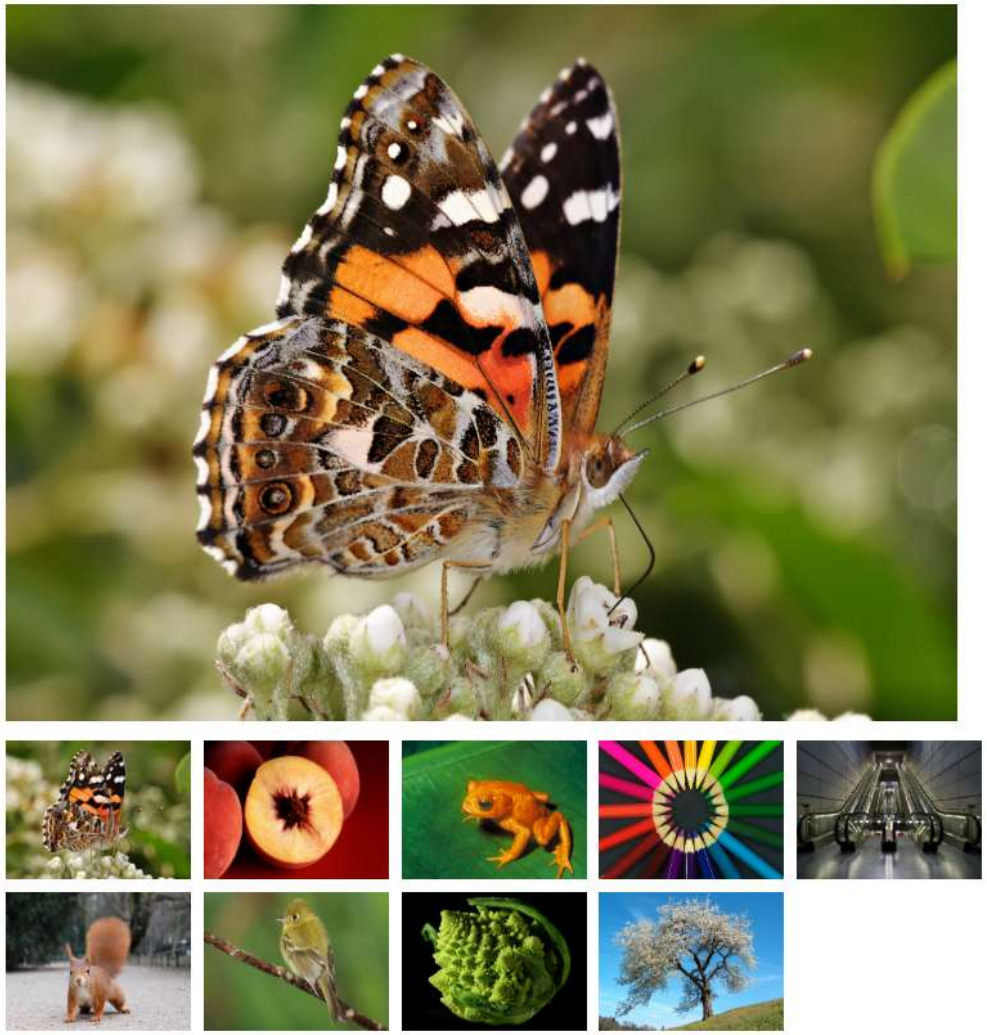
Flexbox

main.css

```
.flex-container {
  display:
        flex;
  flex-wrap:
      wrap;
}
```

```
<link rel=stylesheet type="text/css"
href="main.css">
<body>
<img id="fullview"  src="full/01.jpg">
<div class="flex-container">
<img class="thumb"
onclick='changeMainImg(this);'
src="full/01.jpg">
. . .
<img class="thumb"
onclick='changeMainImg(this);'
src="full/09.jpg">
</div>
<script >
   function changeMainImg(that) {
    document.getElementById("fullview").src=
   that.src;
}
</script>
```

Wrapped Flexbox



Customizing the order

Customizing the order

Customizing the order

Customizing the order

Customizing the order

```
<img id="fullview"  onclick='makeMax();'
src="full/01.jpg">
<div class="flex-container">
. . .
<img class="thumb" onclick='changeMainImg(this);'
src="full/09.jpg">
</div>
<script >
   var current;
   function changeMainImg(that) {
      document.getElementById("fullview").src = that.src;
      current = that;  }
   var maxorder = -100;
    function makeMax() {
      maxorder--;
      if (current) { current.style.order =
                        maxorder; }    }
</script>
```

State variables

```
<img id="fullview"  onclick='makeMax();'
src="full/01.jpg">
<div class="flex-container">
. . .
<img class="thumb" onclick='changeMainImg(this);'
src="full/09.jpg">
</div>
<script >
   var current;
   function changeMainImg(that) {
      document.getElementById("fullview").src = that.src;
      current = that;  }
   var maxorder = -100;
    function makeMax() {
      maxorder--;
      if (current) { current.style.order =
                        maxorder; }    }
</script>
```

Resetting the order









Resetting the order

Resetting the order

```
<img id="fullview"  onclick='makeMax();'  ondblclick ='myReset();'
src="full/01.jpg">
<div class="flex-container">
<img class="thumb" data-ord="1"  onclick='changeMainImg(this);'
src="full/01.jpg">
. . .
</div>
<script >
const thumbs = document.querySelectorAll('.thumb');
    var current;
    function changeMainImg(that) {
        document.getElementById("fullview").src = that.src;
                  current = that;     }
         var maxorder = -100;
    function makeMax() {
             maxorder--;
                   if (current) { current.style.order = maxorder; }    }
    function myReset() {
          thumbs.forEach(thumb => {
              thumb.style.order = thumb.dataset.ord;});     }
</script>
```

Lecture notes

- JavaScript is a scripting language closely integrated with HTML and CSS.
- JavaScript is needed to provide functionality to a Web-technology based interface.
- The access of JavaScript to HTML gives us a powerful view of the overall HTML document.
- It is good to stick to bounded loops in the program because they cannot run into endless loops.
- Interface implementation with HTML/CSS/JavaScript can benefit from the robustness properties of the technology.
- JavaScript is readily executable in the Web browser, and web browsers even offer developer tools.