COMP 307
Principles
of Web
Development

Contents

Intro to JS
Standard programs
Event-based

MCGILL UNIVERSITY

COMP 307

Principles of Web Development

Lecture 5

Unit 2 – Frontend Internet Languages

JavaScript 1

COMP 307
Principles of Web Development

Contents

Intro to JS
Standard programs
Event-based

Class Outline

- Introduction to JS
- Using JS as a standard language
- Using JS as an event-based language

Readings

- Internet and World Wide Web textbook
  - Chapters: 6 to 10
- The Full Stack Developer
  - Chapter 8
- Web resources
  - https://www.w3schools.com/js/
  - https://www.tutorialspoint.com/javascript/index.htm

COMP 307
Principles
of Web
Development

Introduction to JS

Java Script 1

Contents

Intro to JS
Standard programs
Event-based

COMP 307
Principles of Web Development

Two ways to program JS

Standard Language

`A main() that calls functions()`

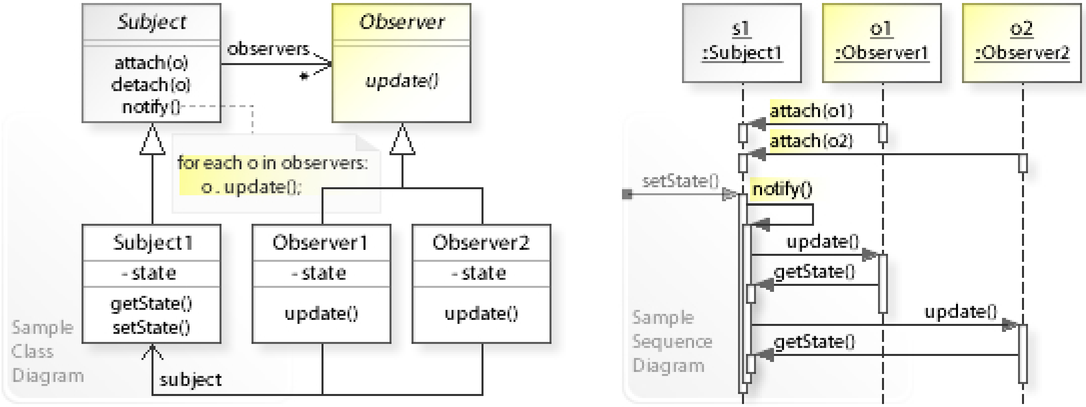
This form of programming permits you to create standard algorithms.

Event-based

`Just functions() with triggers`

This form of programming responds to the user’s interaction with the webpage.

clicking, mousing over



Observer Design Pattern

event-based language

[Contents](#)

Intro to JS
Standard programs
Event-based

COMP 307 Principles of Web Development

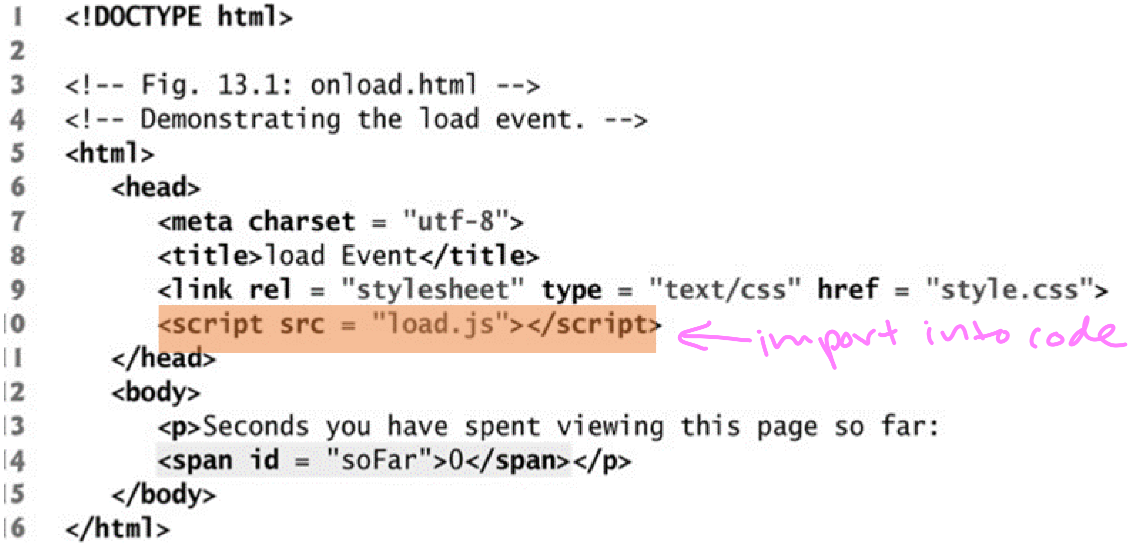
Where to write Code

Internally

```
<script>

  write you code

</script>
```

Same

Externally



```
<!DOCTYPE html>

<!-- Fig. 13.1: onload.html -->
<!-- Demonstrating the load event. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>load Event</title>
      <link rel = "stylesheet" type = "text/css" href = "style.css">
      <script src = "load.js"></script>
   </head>
   <body>
      <p>Seconds you have spent viewing this page so far:
      <span id = "soFar">0</span></p>
   </body>
</html>
```

← import into code

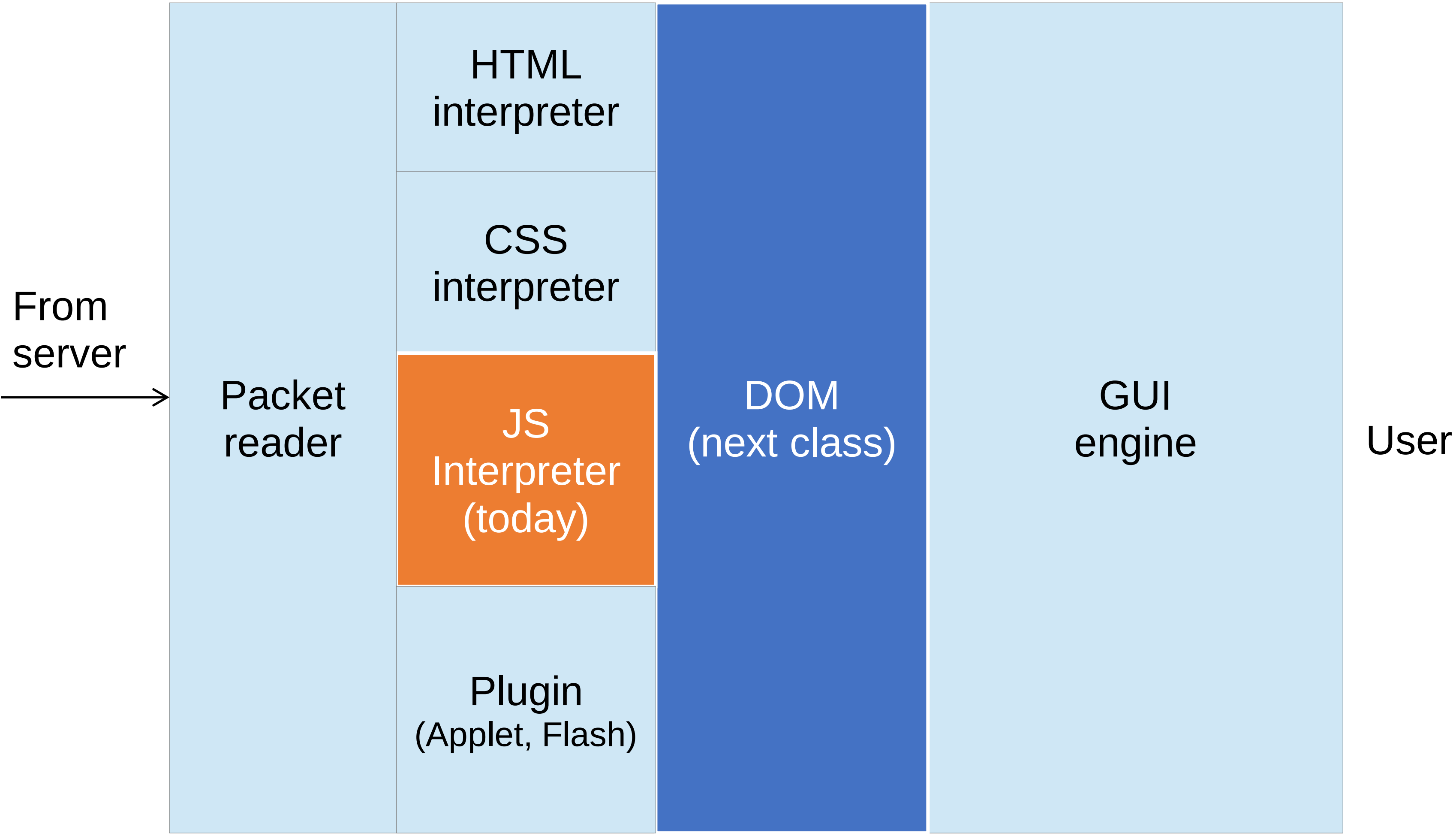
nice to have in different document to edit it and think about it separately } and no code repitition

Contents

Intro to JS
Standard programs
Event-based

COMP 307 Principles of Web Development

The Browser System



Js and dom intimately connected

Contents

Intro to JS
Standard programs
Event-based

What is JavaScript

- Written in the HTML file
- Executed on the client-side
- Browser **interpreted** scripting language
- Code can be viewed by the user ! ← ☹
  - Privacy and proprietary issues
- JS can access and modify the DOM } power where JS can re-edit webpage live

COMP 307
Principles
of Web
Development

As a Standard Language

Java Script 1

Contents

Intro to JS
Standard programs
Event-based

COMP 307
Principles of Web Development

I/O

document object = access to DOM

Note: document object

```
<!DOCTYPE html>
<html>
 <body>

   <h2>My First Web Page</h2>
   <p>My first paragraph.</p>

   <p>Never call document.write after the document has finished loading. It will overwrite the document.</p>

   <script>
     document.write(5 + 6);
   </script>

 </body>
</html>
```

← programming print

standard

```
<!DOCTYPE html>
<html>
 <body>

   <h2>My First Web Page</h2>
   <p>My first paragraph.</p>

   <button type="button" onclick="document.write(5 + 6)">Try it</button>

 </body>
</html>
```

trigger

renders button and stores code to be executed onclick

Creates whole new render of just 11

event

Contents

Intro to JS
Standard programs
Event-based

COMP 307
Principles of Web Development

Note: window and console objects

I/O

```
<!DOCTYPE html>
<html>
 <body>

  <h2>My First Web Page</h2>
  <p>My first paragraph.</p>

  <script>
    window.alert(5 + 6);
  </script>

 </body>
</html>
```

popup, then webpage loads

⇒ popup

popup

```
<!DOCTYPE html>
<html><body>

  <h2>Activate Debugging</h2>

  <p>F12 on your keyboard will activate debugging.</p>
  <p>Then select "Console" in the debugger menu.</p>
  <p>Then click Run again.</p>

  <script>
   console.log(5 + 6);
  </script>

</body></html>
```

⋮ ⇒ more tools ⇒ developer tools
↓
console

prints to console, not webpage
good for debugging

console

Contents

Intro to JS
Standard programs
Event-based

Variables & Statements

```
<!DOCTYPE html>
<html>
 <body>

   <h2>JavaScript Statements</h2>

   <p>A <b>JavaScript program</b> is a list of <b>statements</b> to be executed by a computer.</p>

   <p id="demo"></p>

   <script>
     let x, y, z;       // Statement 1 – declares a block variable, untyped
     const w=10; // Constant
     x = 5;            // Statement 2 – assignment
     y = 6;            // Statement 3 – assignment
     z = x + y;       // Statement 4 – expression

     document.getElementById("demo").innerHTML = "The value of z is " + z + ".";  // assignment
   </script>

 </body>
</html>
```

DOM

This program uses DOM

COMP 307
Principles
of Web
Development

Contents

Intro to JS
Standard programs
Event-based

Basic JS

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Functions</h2>

<p>This example calls a function to convert from Fahrenheit to Celsius:</p>
<p id="demo"></p>

<script>
  function toCelsius(f) {
    return (5/9) * (f-32);
  }

  document.getElementById("demo").innerHTML = toCelsius(77);
</script>

</body>
</html>
```

find demo, replace with →

Placed anywhere

Like Bash:
- Main
- Fn()

Inserting result into the DOM

COMP 307 Principles of Web Development

Iteration

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Do/While Loop</h2>

<p id="demo"></p>

<script>
  var text = ""
  var i = 0;

  do {
    text += "<br>The number is " + I +”</br>;
    i++;
  }
  while (i < 10);

  document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

apending html

Old tyle variables

Building a string of HTML

Inserting it into the DOM

Looks somewhat like Java

Contents

Intro to JS
Standard programs
Event-based

In-line JS

Add this to any tag to make it click-able

```
<!DOCTYPE html>
<html>
<body>

<button onclick="document.getElementById('demo').innerHTML=Date()">The
time is?</button>

<p id="demo"></p>

</body>
</html>
```

Treated as a function call

Library function

COMP 307 Principles of Web Development

In-line JS (2)

Calling it like a function

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to display the date.</p>

<button onclick="displayDate()">The time is?</button>

<script>
  function displayDate() {
    document.getElementById("demo").innerHTML = Date();
  }
</script>

<p id="demo"></p>

</body>
</html>
```

Could go in here

Contents

Intro to JS
Standard programs
Event-based

COMP 307 Principles of Web Development

Dynamic Web Pages

```
<!DOCTYPE html>
<html>
<body>

<p id="p1">
This is a text.
This is a text.
This is a text.
</p>

<input type="button" value="Hide text"
onclick="document.getElementById('p1').style.visibility='hidden'">

<input type="button" value="Show text"
onclick="document.getElementById('p1').style.visibility='visible'">

</body>
</html>
```

FORM element

change the page

Contents

Intro to JS
Standard programs
Event-based

Cookies

- http://www.w3schools.com/js/tryit.asp?filename=tryjs_doc_cookie
- http://www.w3schools.com/js/tryit.asp?filename=tryjs_cookie_username

Contents

Intro to JS
Standard programs
Event-based

COMP 307
Principles
of Web
Development

Programming Events

Java Script 1

Contents

Intro to JS
Standard programs
Event-based

COMP 307
Principles
of Web
Development

System Events

```
<!DOCTYPE html>
<html>
  <body>

    <h1>The Document Object</h1>
    <h2>The addEventListener() Method</h2>

    <p>This example adds many events on the document.</p>

    <p id="demo"></p>

    <script>
      document.addEventListener("mouseover", myFunction);
      document.addEventListener("click", mySecondFunction);
      document.addEventListener("mouseout", myThirdFunction);

      function myFunction() {
        document.getElementById("demo").innerHTML = "Moused over!"
      }

      function mySecondFunction() {
        document.getElementById("demo").innerHTML = "Clicked!<br>"
      }

      function myThirdFunction() {
        document.getElementById("demo").innerHTML = "Moused out!<br>"
      }
    </script>

  </body>
</html>
```

event added to DOM, not particular tag, eg mousing over whole webpage

Contents

Intro to JS
Standard programs
Event-based

COMP 307
Principles
of Web
Development

More Mouse Examples

- https://www.w3schools.com/js/tryit.asp?filename=tryjs_events_onmouse
- https://www.w3schools.com/js/tryit.asp?filename=tryjs_events_onmousedown

Contents

Intro to JS
Standard programs
Event-based

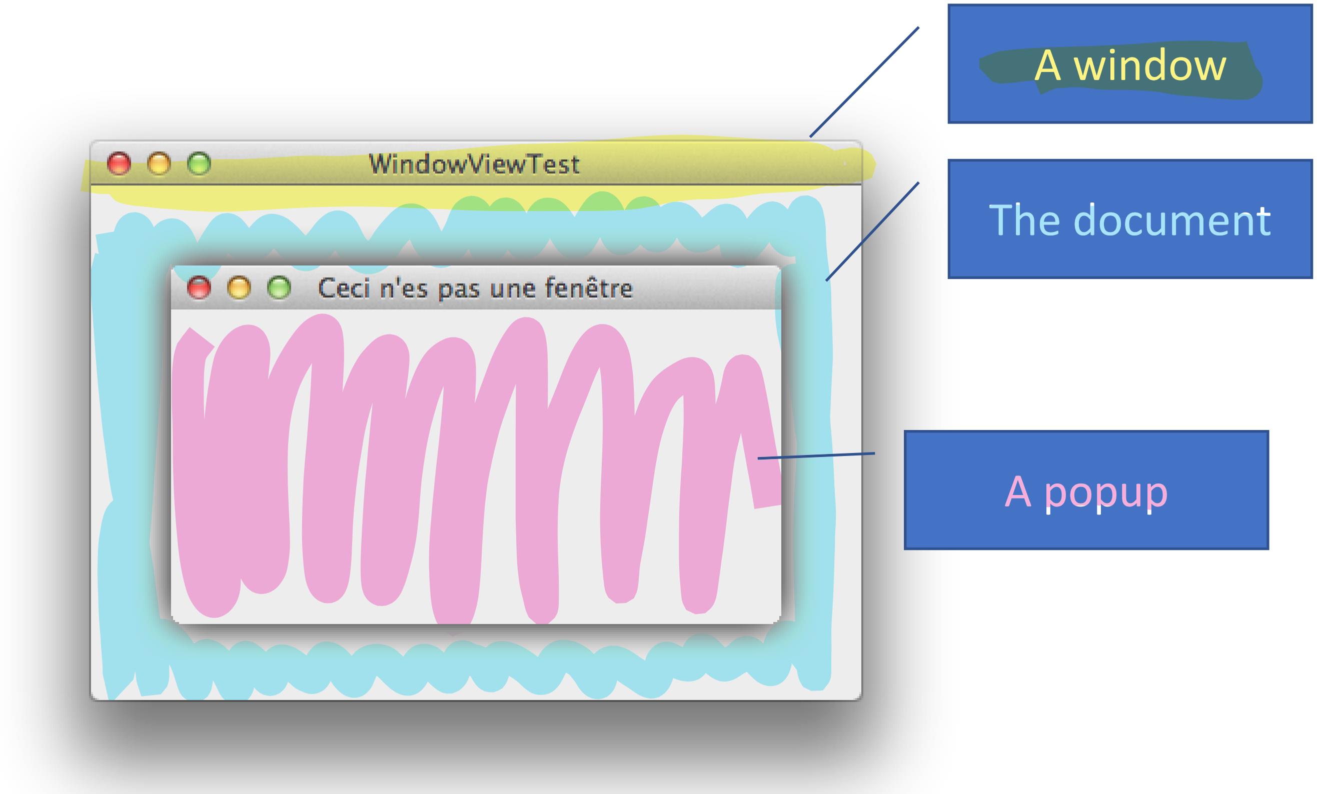
onload System Event

```
<html>
  <body>

    <script>
      function start() {
        document.body.setAttribute("style","background-color:red");
      }

      window.addEventListener("load",start);
    </script>

  </body>
</html>
```

Not in a function

when the page loads, call start function

document vs window addEventListener



A window

The document

A popup

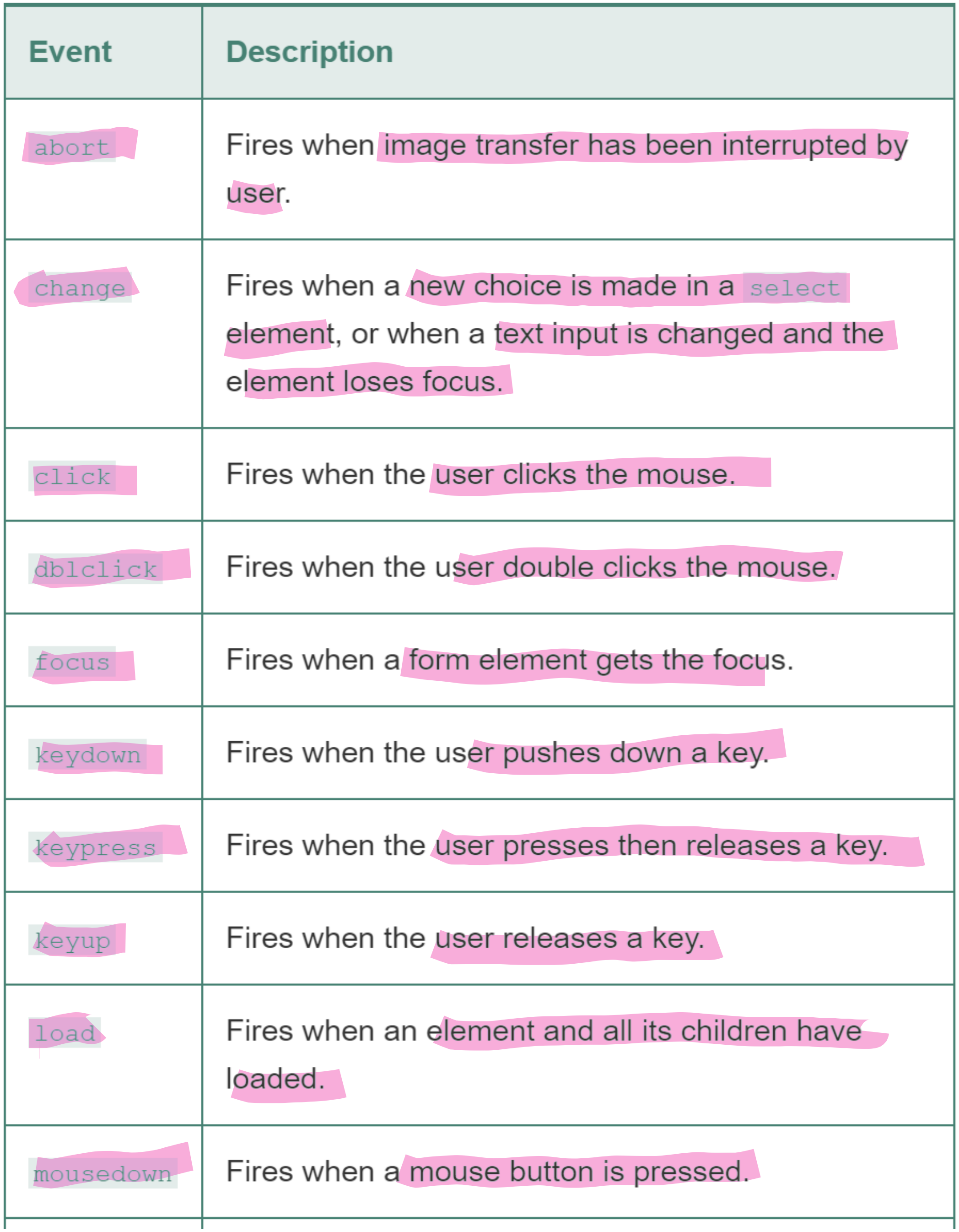
The “window” and “document” objects have different methods and attributes.

For example, you can “resize” a window, but the document inherits the window’s size.

Contents

Intro to JS
Standard programs
Event-based

Some other events



Event	Description
abort	Fires when image transfer has been interrupted by user.
change	Fires when a new choice is made in a select element, or when a text input is changed and the element loses focus.
click	Fires when the user clicks the mouse.
dblclick	Fires when the user double clicks the mouse.
focus	Fires when a form element gets the focus.
keydown	Fires when the user pushes down a key.
keypress	Fires when the user presses then releases a key.
keyup	Fires when the user releases a key.
load	Fires when an element and all its children have loaded.
mousedown	Fires when a mouse button is pressed.

Contents

Intro to JS
Standard programs
Event-based

COMP 307
Principles of Web Development

Timer Events

```
<!DOCTYPE html>
<html>
 <body>
  <h2>JavaScript Timing</h2>
  <p>Click "Try it". Wait 3 seconds, and the page will alert "Hello".</p>
  <button onclick="setTimeout(myFunction, 3000);">Try it</button>

  <script>
    function myFunction() {
      alert('Hello”);
    }
  </script>
 </body>
</html>
```

```
<!DOCTYPE html>
<html><body>
  <p>A script on this page starts this clock:</p>
  <p id="demo"></p>
  <button onclick="clearInterval(myVar)">Stop time</button>

  <script>
    let myVar = setInterval(myTimer ,1000); – calls myTimer every second
    function myTimer() {
       const d = new Date();
       document.getElementById("demo").innerHTML = d.toLocaleTimeString();
    }
  </script>
</body></html>
```

Clock in digits

prints time to demo every second

Contents

Intro to JS
Standard programs
Event-based

Event Listeners

JavaScript 1

```
<!DOCTYPE html>
<html><head>
  <title>JS Mouse Events - Button Demo</title>
</head>
<body>
  <button id="btn">Click me with any mouse button: left, right, middle, ...</button>
  <p id="message"></p>
  <script>
    let btn = document.querySelector('#btn');

    // disable context menu when right-mouse clicked
    btn.addEventListener('contextmenu', (e) => { e.preventDefault();  });

    // show the mouse event message
    btn.addEventListener('mouseup', (e) => {
      let msg = document.querySelector('#message');
      switch (e.button) {
        case 0:
          msg.textContent = 'Left mouse button clicked.';
          break;
        case 1:
          msg.textContent = 'Middle mouse button clicked.';
          break;
        case 2:
          msg.textContent = 'Right mouse button clicked.';
          break;
        default:
          msg.textContent = `Unknown mouse button code: ${event.button}`;
      }
    });
  </script>
</body></html>
```

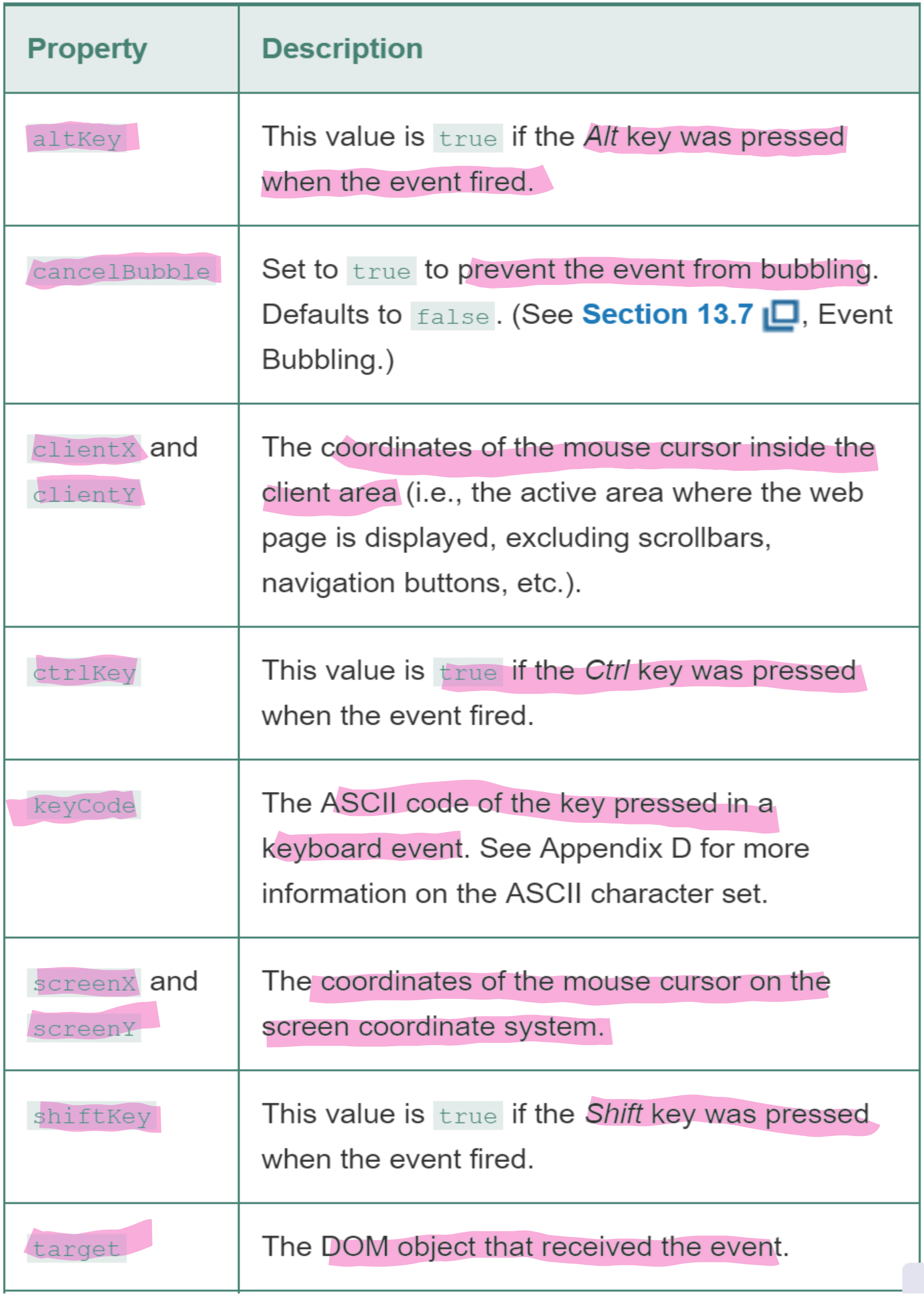
The event returns an object describing what happened

using different keys

```
<!DOCTYPE html>
<html>
<head>
  <title>JS Modifier Keys Demo</title>
</head>
<body>
  <button id="btnKeys">Click me with Alt, Shift, Ctrl pressed</button>
  <p id="messageKeys"></p>

  <script>
    let btnKeys = document.querySelector('#btnKeys');

    btnKeys.addEventListener('click', (e) => {
      let keys = [];

      if (e.shiftKey) keys.push('shift');
      if (e.ctrlKey) keys.push('ctrl');
      if (e.altKey) keys.push('alt');
      if (e.metaKey) keys.push('meta'); ← windows key

      let msg = document.querySelector('#messageKeys');
      msg.textContent = `Keys: ${keys.join('+')}`;
    });
  </script>
</body>
</html>
```

COMP 307 Principles of Web Development

Some event object properties



Property	Description
`altKey`	This value is `true` if the *Alt* key was pressed when the event fired.
`cancelBubble`	Set to `true` to prevent the event from bubbling. Defaults to `false`. (See **Section 13.7**, Event Bubbling.)
`clientX` and `clientY`	The coordinates of the mouse cursor inside the client area (i.e., the active area where the web page is displayed, excluding scrollbars, navigation buttons, etc.).
`ctrlKey`	This value is `true` if the *Ctrl* key was pressed when the event fired.
`keyCode`	The ASCII code of the key pressed in a keyboard event. See Appendix D for more information on the ASCII character set.
`screenX` and `screenY`	The coordinates of the mouse cursor on the screen coordinate system.
`shiftKey`	This value is `true` if the *Shift* key was pressed when the event fired.
`target`	The DOM object that received the event.

Contents

Intro to JS
Standard programs
Event-based

COMP 307
Principles
of Web
Development

<u>Contents</u>

Intro to JS
Standard programs
Event-based

Input and JS

JavaScript 1

Focus & Blur on Inputs

- The focus and blur events can be useful when dealing with input elements that allow user input.
  - The focus event fires when an element gains the focus (i.e., when the user clicks an input field or uses the Tab key to move between input fields), and
  - The blur fires when an input field loses the focus

COMP 307
Principles of Web Development

```
<!DOCTYPE html>
<html>
<body>

<p>In this example, the text field gets focus immediately after the document window has been loaded.</p>

<input type="text" id="myText" value="A text field">

<script>
window.onload = function() {
  document.getElementById("myText").focus();
};
</script>

</body>
</html>
```

puts cursor into on page load

Contents

Intro to JS
Standard programs
Event-based

COMP 307 Principles of Web Development

```
<!DOCTYPE html>
<html>
<body>

<h1>The Element Object</h1>
<h2>The focus() and blur() Methods</h2>

<input type="text" id="myText" value="A text field">

<p>Click the buttons to give or remove focus from the text field.</p>

<button type="button" onclick="getFocus()">Get focus</button>
<button type="button" onclick="loseFocus()">Lose focus</button>

<script>
function getFocus() {
  document.getElementById("myText").focus();
}

function loseFocus() {
  document.getElementById("myText").blur();
}
</script>

</body>
</html>
```

Prepare for next class

- Assignments
  - How is mini 2 going?
- No labs this week
- On your own
  - Try out the code from class

Contents

Intro to JS
Standard programs
Event-based