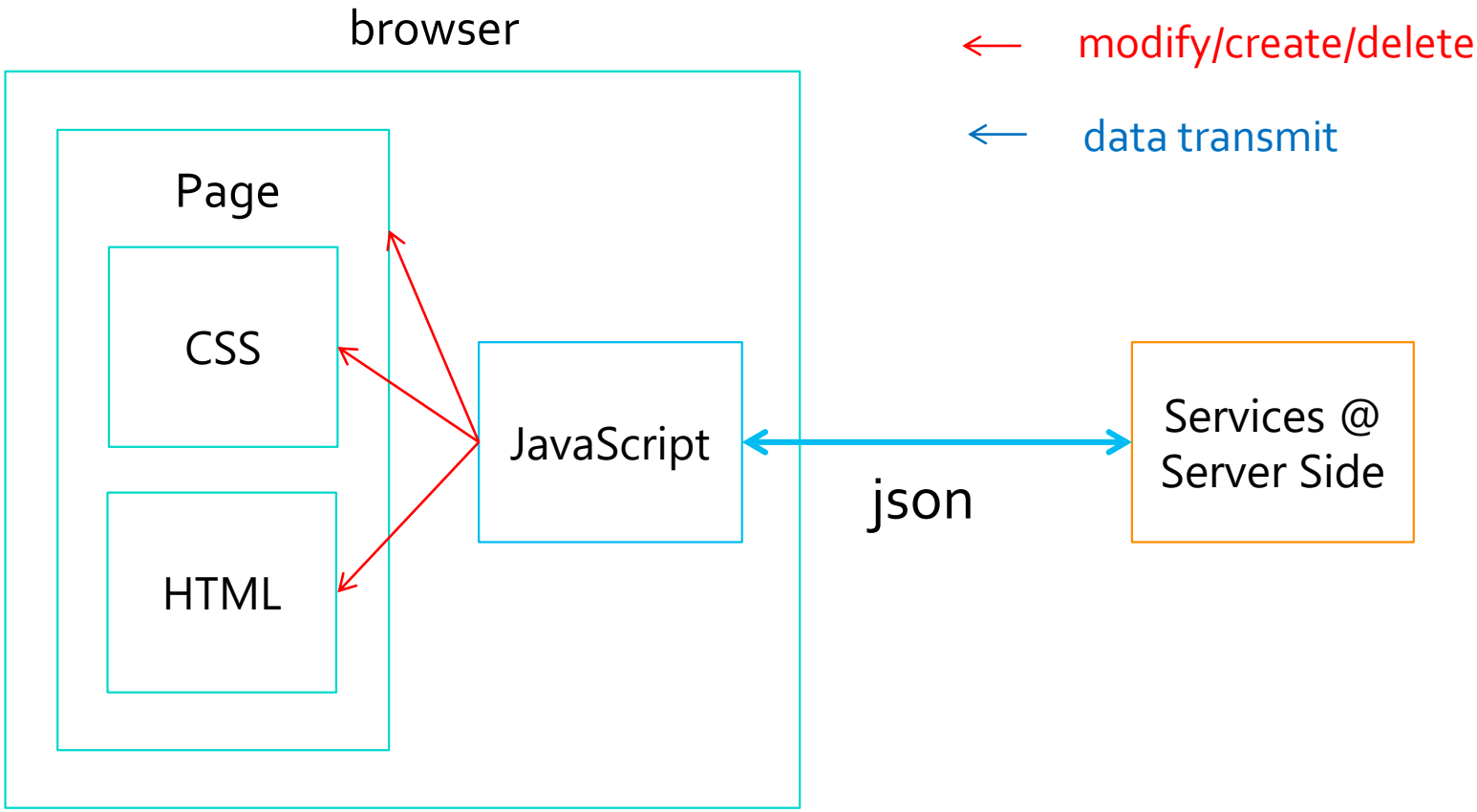
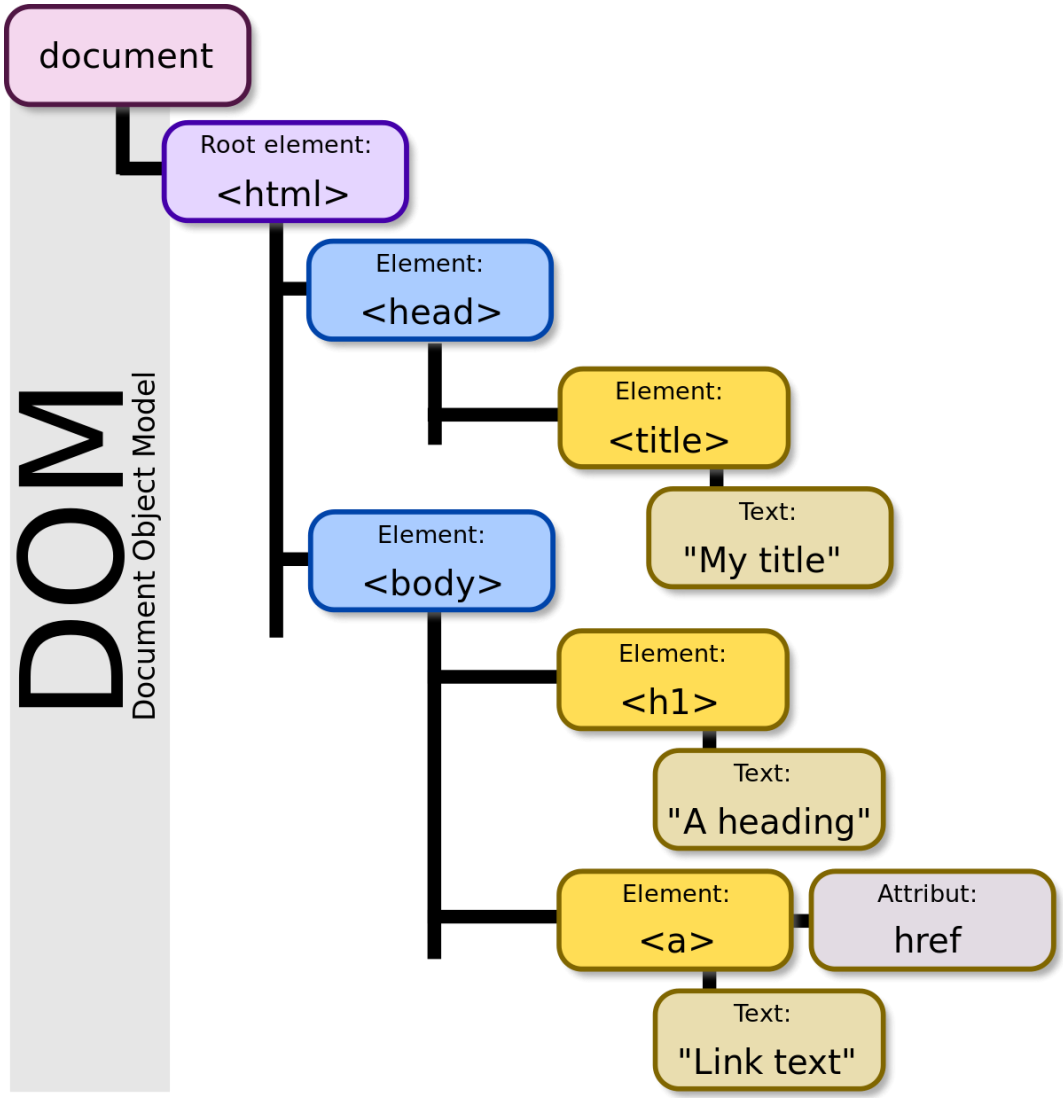
DOM & jQuery

How JavaScript fits in the big Picture



Document Object Model (DOM)



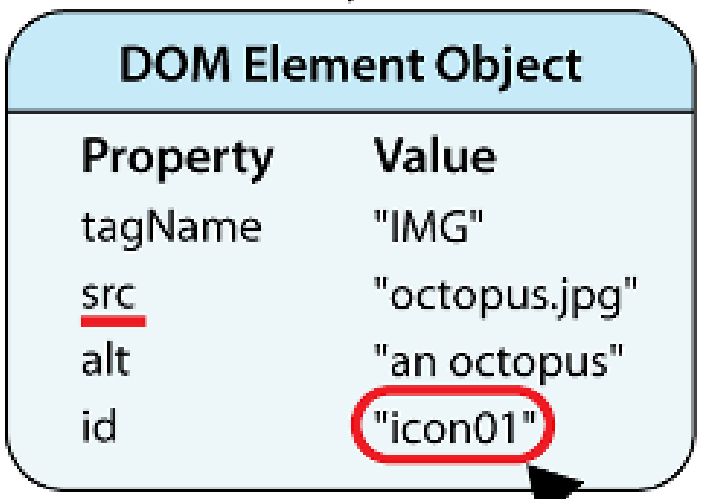
Document Object Model (DOM)

- Every HTML element is accessible via the JavaScript DOM API
- The **event model** lets a document to react when the user does something on the page
- Advantages
  - Create interactive pages
  - Updates the objects of a page without reloading it

Example

HTML

```
<p>
  Look at this octopus:
  <img src="octopus.jpg" alt="an octopus" id="icon01" />
  Cute, huh?
</p>
```



JavaScript

```
var icon = document.getElementById("icon01");
icon.src = "kitty.gif";
```

Accessing Elements

- Access elements via their ID attribute

```
let element = document.getElementById("some-id")
```

- Via the **name** attribute

```
let elArray = document.getElementsByName("some-name")
```

- Via tag name

```
let imgTags = document.getElementsByTagName("img")
```

  - Returns array of `<img>` elements

DOM Manipulation

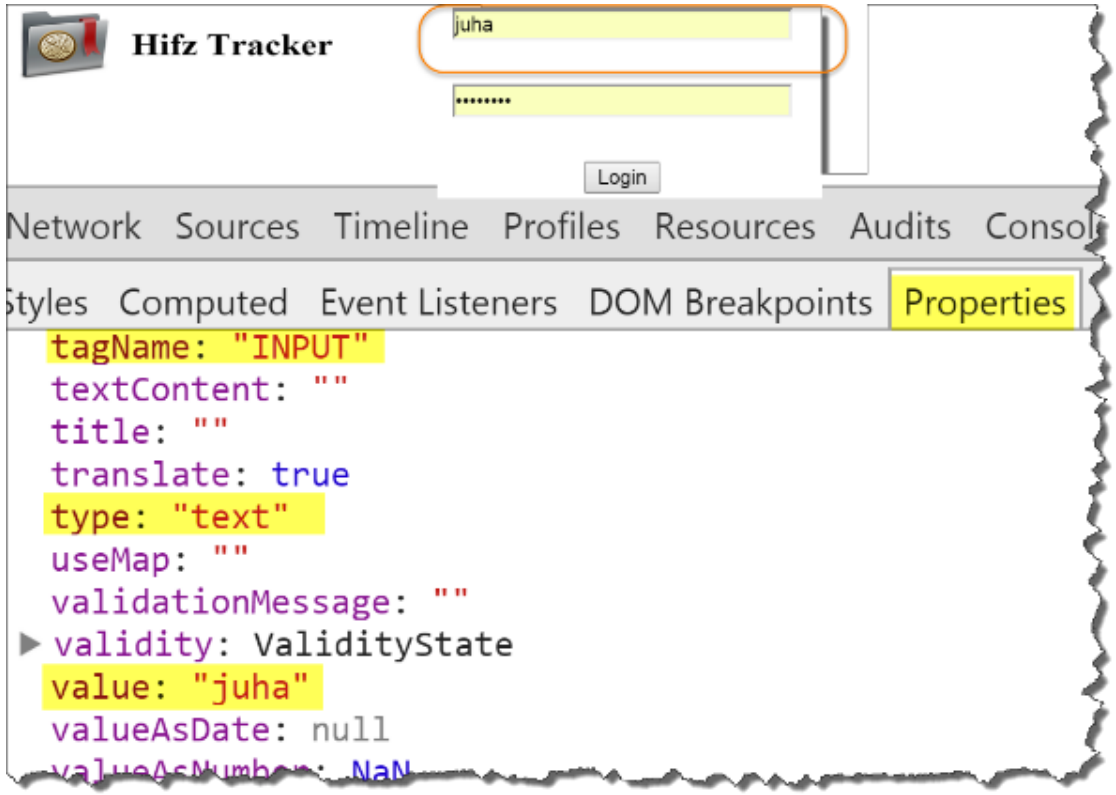
- Once we access an element, we can read and write its attributes

```
function change(state) {
  let lampImg = document.getElementById("lamp");
  lampImg.src = "lamp_" + state + ".png";
  let statusDiv =
    document.getElementById("statusDiv");
  statusDiv.innerHTML = "The lamp is " + state";
}
…
<img src="test_on.gif"  id="lamp"
  onmouseover="change('off')"
  onmouseout="change('on')" />
```

Common Element Properties

- `innerHTML` – holds all the entire HTML code inside the element
- `className` – the `class` attribute of the tag

User Chrome Dev Tool to see the Properties of Page element



The HTML DOM Event Model

- JavaScript can register event handlers
  - Events are fired by the Browser and are sent to the specified JavaScript event handler function
  - Can be set with HTML attributes:

```
<img src="test.gif" onclick="imageClicked()" />
```

  - Can be set through the DOM:

```
let img = document.getElementById("myImage");
img.addEventListener('click', imageClicked);
```

Common DOM Events

- Mouse events:
  - `onclick`, `onmousedown`, `onmouseup`
  - `onmouseover`, `onmouseout`, `onmousemove`
- Key events:
  - `onkeypress`, `onkeydown`, `onkeyup`
  - Only for input fields
- Interface events:
  - `onblur`, `onfocus`
  - `onscroll`

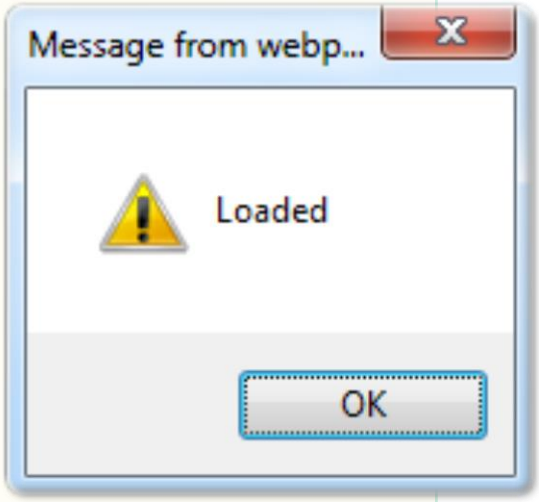
Common DOM Events (2)

- Form events
  - `onchange` – for input fields
  - `onsubmit`
    - Allows you to cancel a form submission
    - Useful for form validation
- Document events
  - `onload`
    - Allowed only for the `<body>` element
    - Fires when all content on the page was loaded

onload Event – Example

- onload event

```
<html>
<head>
  <script type="text/javascript">
    function greet() {
      alert('Loaded!');
    }
  </script>
</head>
<body onload="greet()" >
</body>
</html>
```



Event Handler

```
<script>
document.getElementById("myBtn").
    addEventListener("click", displayDate);

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

Try it @
http://www.w3schools.com/js/tryit.asp?filename=tryjs_addeventlistener_displaydate

https://jquery.com/

jQuery

- jQuery is a fast, small and feature-rich JavaScript library that works across a multitude of browsers
- Simplifies HTML document traversing, event handling and animation
- To include jQuery in your website, all you need is a script tag with its *src* pointed to the hosted location

```
<script
src="https://code.jquery.com/jquery-3.1.1.slim.min.js">
</script>
```

jQuery Syntax

- You can use the **$()** function to **select** HTML elements and perform some **action** on the element(s)
- Basic syntax is: **$(*selector*).*action*()**
  - A **$** sign to access jQuery
  - A (***selector***) to find HTML elements
  - A ***action***() to be performed on the element(s)

jQuery Selectors

jQuery supports CSS :

1. By element: `$("div")`
2. By id : `$("#id")`
3. By class: `$(".classname")`
4. By attribute: `$("a[href]")`
5. …

DOM method	jQuery equivalent
getElementById("id")	$("#id")
getElementsByTagName("tag")	$("tag")
getElementsByName("somename")	$("[name='somename']")

jQuery Syntax

- Examples:
  - $("p").hide() - hides all <p> elements
  - $(".test").hide() - hides all elements with class="test"
  - $("#test").hide() - hides the element with id="test“
  - $('div').css('background', 'blue'); - Make all DIVs blue

.ready() event

- jQuery provides a *ready* event that is fired when the document is ready to be manipulated

- You'll put most of your code in this method

```
$(document).ready( () => {
   // Your code here e.g.,
   alert("Ok document is ready…");
});
```

Creating Elements

- Creating new elements is also easy

```
let divElement = $('<div>');
let paragraph = $('<p>Some text</p>');
```

Adding Elements

- Adding elements can be done on the fly
  - **jQuery.appendTo() / jQuery.prependTo()**
  - **jQuery.append() / jQuery.prepend()**

```
<h2>QU Greetings</h2>
<div id="wrapper">
  <div>Hello, student</div>
  <div>Goodbye, student</div>
</div>
```

```
<h2>QU Greetings</h2>
<div id="wrapper">
  <div>Hello, student<p>Test</p></div>
  <div>Goodbye, student<p>Test</p></div>
</div>
```

```
$('#wrapper div').append('<p>Test</p>');
```

```
<div>First</div>
<h2>QU Greetings</h2>
<div id="wrapper">
  <div>Hello, studentdiv>
  <div>Goodbye, student<div>
</div>
```

```
$('<div>First</div>').prependTo('body');
```

Creating complex nodes in jQuery

- **The terrible way:**

```
$("<p id='myid' class='special'>My paragraph is awesome!</p>")
```

- **The bad way:**

```
$("<p>")
      .attr("id", "myid")
      .addClass("special")
      .text("My paragraph is awesome!");
```

- **The good way:**

```
$("<p>", {
      "id": "myid",
      "class": "special",
      "text": "My paragraph is awesome!"
});
```

Removing Elements

- You can also easily remove elements from the DOM

```
<div>
  <p>Red</p>
  <p>Green</p>
</div>

<script>
  $('p').remove(); // Remove all paragraphs
</script>
```

```
<!-- Result: -->
<div>
</div>
```

jQuery Events

- jQuery has a convenient way for attaching and detaching events Using methods **on()** and **off()**

```
function onButtonClick() {
  $(this).hide();
// "this" is the event source (the button clicked)
}

$('#button').on('click', onButtonClick);
```

Looping over the DOM

- Using the DOM

```
let elems = document.querySelectorAll("li");
for (let i = 0; i < elems.length; i++) {
      let e = elems[i];
      // do stuff with e
}
```

- Using jQuery

```
let elems = $("li");
for ( let e of elems )
      // do stuff with e
}
```

jQuery css method parameters

- **Getter:**

```
$("#myid").css(propertyName);
```

- **Setter:**

```
$("#myid").css(propertyName, value);
```

- **Multi-setter:**

```
$("#myid").css({
        'propertyName1': value1,
        'propertyName2': value2,
        ...
        });
```

More node manipulation with jQuery

jQuery method	functionality
.hide()	toggle CSS display: none on
.show()	toggle CSS display: none off
.empty()	remove everything inside the element, innerHTML = ""
.html()	get/set the innerHTML without escaping html tags
.text()	get/set the innerHTML, HTML escapes the text first
.val()	get/set the value of a form input, select, textarea, ...
.height()	get/set the height in pixels, returns a Number
.width()	get/set the width in pixels, return a Number

Summary

- jQuery – the most popular client-side JS library
- Select DOM elements with jQuery
  - **`$([selector])`**
- DOM Traversal:
  - **`$([selector]).next()`** / **`parent()`**
- Altering the DOM:
  - **`$([selector]).html(…)`** / **`append(…)`**
- jQuery Events
  - **`$([selector]).on([event], [callback]);`**

Resources

- W3C School:

  http://www.w3schools.com/jquery/

- Code School:

  http://www.codeschool.com/courses/jquery-air-first-flight