COMP4021
Internet Computing

jQuery

Gibson Lam and David Rossiter

jQuery

- jQuery is a JavaScript library that makes your JavaScript code writing easier and more powerful

- jQuery code is typically more concise and cleaner than most JavaScript code
- jQuery hides various issues with different browsers from the programmer

Using jQuery

- It is **not necessary** to use jQuery to do things in a web page
- However, it usually helps a lot if you use it
- According to the page below, 78% of all websites use jQuery

https://w3techs.com/technologies/overview/javascript_library/all

- Let's start by looking at how to include it in your web page

Adding jQuery to Your Page 1/2

- You can use one of the two approaches to add jQuery in your web page
- **Approach 1:**
  - Download jQuery from http://jquery.com/ and then add a link to the library from your HTML page, i.e.:

```
<script src="jquery-3.6.0.js"></script>
```

The jQuery library you have downloaded to the folder containing your webpage

Adding jQuery to Your Page 2/2

- **Approach 2:**
  - You use the jQuery file from somewhere else
  - There are lots of copies of the jQuery library on the web
  - Some organisations make a CDN (Content Delivery Network) which means the jQuery library is distributed around the world, and you will automatically receive the file from the closest server

jQuery CDNs

- For example, to link to the Google jQuery CDN, you would do something like this:

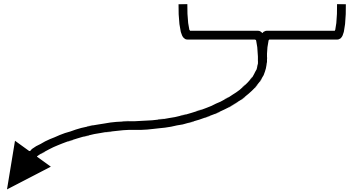
```
<head>
...
<script src="https://ajax.googleapis.com/ajax/ ⇨ to next line
from prev line ⇨ libs/jquery/3.6.0/jquery.min.js"></script>
...
</head>
```

- To link to the jquery.com CDN, use this instead:

```
<head>
...
<script src="https://code.jquery.com/ ⇨ to next line
from prev line ⇨ jquery-3.6.0.min.js"></script>
...
</head>
```

The ‘Minified’ jQuery

```
<script src="https://code.jquery.com/
                jquery-3.6.0.min.js"></script>
```



- You are using the ‘minimum’ version, i.e. and sometimes it is called the ‘minified’ version
- This version uses clever tricks to make the file much smaller e.g. no spaces unless necessary, variables names that use just 1 or 2 letters, etc
- The ‘minimum’ version is about 87KB
- The regular version is about 281KB

Basic jQuery Use

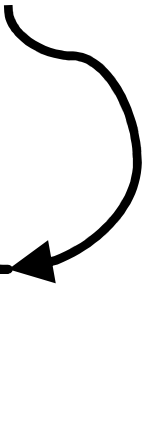
- You write code in jQuery similar to what you are doing using DOM functions, i.e.:
  - Access some elements from the DOM
  - Then do something with those elements
  - Often jQuery code is triggered by an event
- We will briefly look at the first two ideas in this presentation and events later

An Example HTML File 1/2

- We will use this HTML file in the examples in the following slides

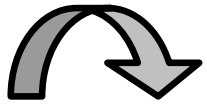
```
<!DOCTYPE html>
<html>
<head><title>Mac and Cheese</title></head>
<body>
  <h1 id="name">Mac and Cheese</h1>
  <h2>Ingredients</h2>
  <ul id="ingredients">
    <li>1 box of <a href="...">macaroni</a></li>
    <li>1/4 cup of <a href="...">butter</a></li>
    <li>1/4 cup of <a href="...">flour</a></li>
    <li>1/2 tsp of <a href="...">salt</a></li>
    <li>2 cups of <a href="...">milk</a></li>
    <li>2 cups of <a href="...">cheddar cheese</a></li>
  </ul>
```

Links are not shown here to save space



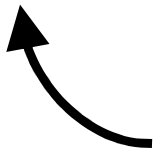
- *Continued on the following slide*

An Example HTML File 2/2



- This HTML in a browser is shown on the next slide

```
  <h2>Directions</h2>
  <ol id="directions">
    <li class="step">Cook the macaroni</li>
    <li class="step">Mix the butter, flour and
                     salt in a saucepan</li>
    <li class="step">Add and stir the milk
                     until thicken</li>
    <li class="step">Add slowly the cheese
                     until fully melt</li>
    <li class="step">Mix with the macaroni</li>
  </ol>
</body>
</html>
```



Using `class` *is not just for doing pretty CSS things, it is also used for controlling behaviour*

Showing the Example

- This is how the page looks like after loading it in a browser

Mac and Cheese

Ingredients

- 1 box of macaroni
- 1/4 cup of butter
- 1/4 cup of flour
- 1/2 tsp of salt
- 2 cups of milk
- 2 cups of cheddar cheese

Directions

1. Cook the macaroni
2. Mix the butter, flour and salt in a saucepan
3. Add and stir the milk until thicken
4. Add slowly the cheese until fully melt
5. Mix with the macaroni

Everything Starts From $

- In jQuery, everything that you write starts from the '$' symbol
- It can be used like a function, e.g.:

```
let myheader = $("h1");
```

- It can also be used to provide some useful functions by writing '$.', e.g.:

```
if ($.isNumeric("3.3")) ...
```

Selecting Elements in jQuery

- As you know, you use DOM functions `document.getElementById()` or `document.getElementsByTagName()` to access the DOM and get HTML elements
- In jQuery, you use the `$(...)` function to select elements using *CSS selectors*
- We have already learned some basic CSS selectors in the CSS discussion
- Let's see a few examples in the next slides

Using the id Attribute

- Let's select the `<h1>…</h1>` on the right using its `id`

```
<h1 id="name">
  Mac and Cheese
</h1>
```

- Using DOM functions:

```
let myname =
    document.getElementById("name");
```

- In jQuery:

```
let myname = $("#name");
```

A CSS selector referring to the id called 'name'

The jQuery Object

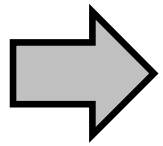
- Important notes about `$(...)`:
  - The result returned by `$(...)` is a *jQuery object*
  - It is **not** returned as a DOM element
- That means some things that you have done before with DOM elements, would have to be done differently in jQuery
- An example with `innerHTML` is shown in the next slide

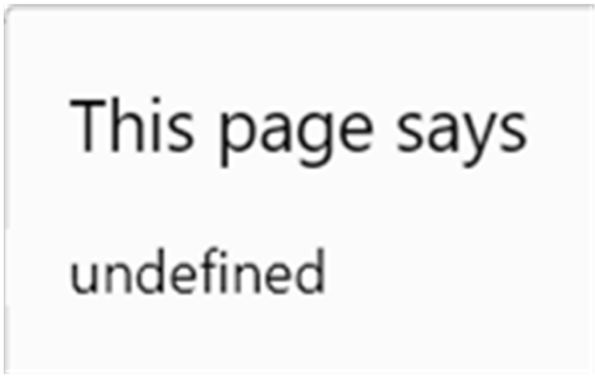
Using the jQuery Object

- It does not work this way using the jQuery result:

```
<h1 id="name">
  Mac and Cheese
</h1>
```

```
let myname = $("#name");
alert(myname.innerHTML);
```

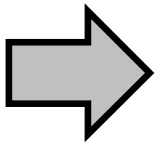


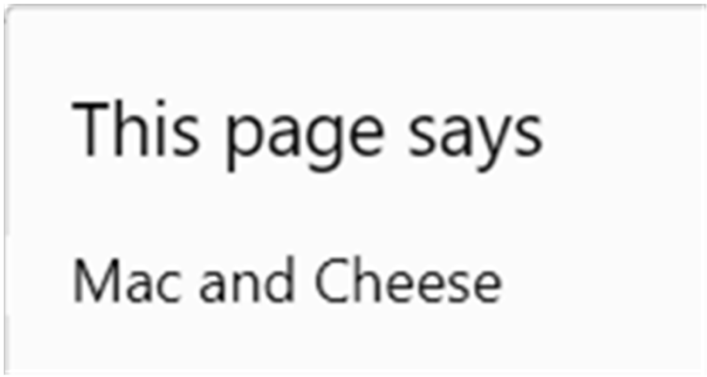


`innerHTML` *would not work for the jQuery object*

- In jQuery, you do it like this:

```
let myname = $("#name");
alert(myname.html());
```





Selecting Elements By Tag Name

- Previously, you select elements of a certain HTML tag using this code, e.g.:

```
...
<h2>Ingredients</h2>
...
<h2>Directions</h2>
...
```

```
let headers =
    document.getElementsByTagName("h2");
```

- In jQuery, you do that by, e.g.:

```
let headers = $("h2");
```

The CSS selector referring to all `<h2>`

Having Multiple Elements

- Most of the operations under the jQuery object can work with multiple elements
- You can read the number of elements in a jQuery object using its `length` property, e.g.:

```
$("title").length  // return 1
```

```
<title>Mac and Cheese</title>
```

```
$("h2").length  // return 2
```

```
<h2>Ingredients</h2>
<h2>Directions</h2>
```

Selecting Elements By Class

- In jQuery you can also do something more with the CSS selector
- You can select elements by their class names
- Remember that you use class for applying visual styles before
- jQuery can use it for selection, e.g.:

```
<li class="step">Cook...
<li class="step">Mix ...
<li class="step">Add ...
<li class="step">Add ...
<li class="step">Mix ...
```

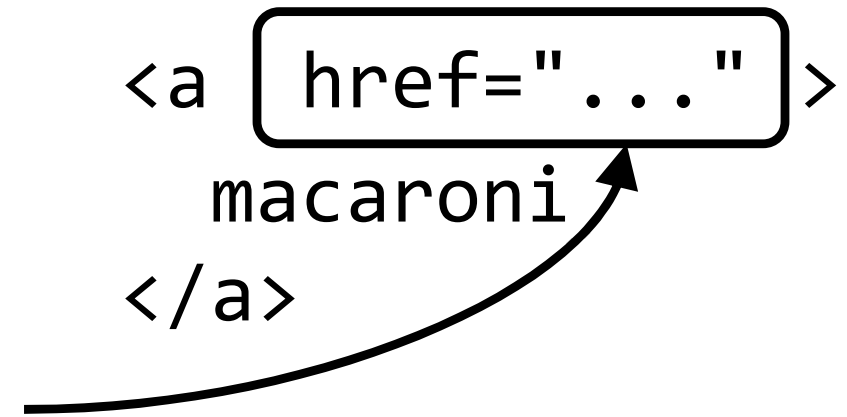
```
let steps = $(".step");
```

Selecting five `<li>`*s*

Reading Attributes

- Now you know how to get elements in jQuery, let's see what it can do, starting from attributes
- If you want to read the attribute of an element, simply do this, e.g.:

```
$("a").attr("href")
```

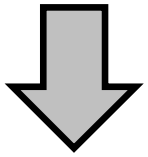


- However, even if the result contains multiple elements, jQuery always read the attribute of the **first** element only!

Writing to Attributes

- You can change the attributes of **all** elements in the jQuery result, like this:

```
$("a").attr("href",
            "https://en.wikipedia.org")
```



```
<a href="https://en.wikipedia.org">macaroni</a>
<a href="https://en.wikipedia.org">butter</a>
<a href="https://en.wikipedia.org">flour</a>
<a href="https://en.wikipedia.org">salt</a>
<a href="https://en.wikipedia.org">milk</a>
<a href="https://en.wikipedia.org">
   cheddar cheese</a>
```

Reading CSS Properties

- You can read and write CSS properties using `.css()` (not the `style` attribute!)
- Similar to `.attr()`, you can read CSS properties for the **first** element only, e.g.:

```
let liColor = $("li.step").css("color");
```

- The above code reads the `color` property of the first `<li>` with the class name `step`

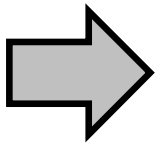
Writing CSS Properties

- You can certainly write the CSS property of **all** elements returned by jQuery, e.g.:

```
$("li.step").css("color", "red");
```

- In above code changes all matching `<li>` to red

1. Cook the macaroni
2. Mix the butter, flour and salt in a saucepan
3. Add and stir the milk until thicken
4. Add slowly the cheese until fully melt
5. Mix with the macaroni



1. Cook the macaroni
2. Mix the butter, flour and salt in a saucepan
3. Add and stir the milk until thicken
4. Add slowly the cheese until fully melt
5. Mix with the macaroni

Reading the Element Content

- Using the DOM, you read the 'inner content' of an element using `innerHTML`
- In jQuery, you can do that using `.html()` and `.text()`

  `.html()` can read, also can create HTML tags

  `.text()` can read, also can create simple text

Using .html()

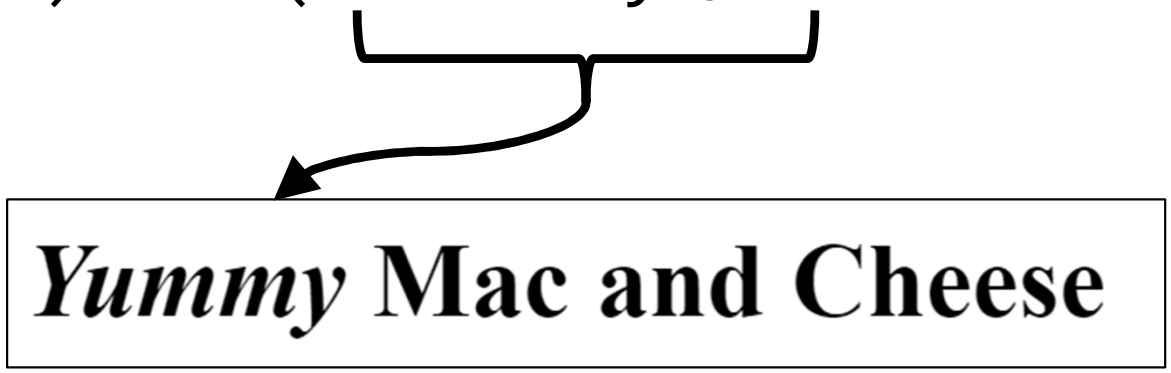
- Using `.html()` is just like `innerHTML`, e.g.:

```
$("#name").html("Yummy Mac and Cheese");
```

Yummy Mac and Cheese

- `.html()` works for HTML content too, i.e.:

```
$("#name").html("<i>Yummy</i> Mac and Cheese");
```



Using .text()

- Using `.text()` gives you a different result when the content has HTML, e.g.:

```
$("#name").text("<i>Yummy</i> Mac and Cheese");
```

- You would want to use `.text()` sometimes when you want to show HTML entities, i.e. `<`, `>`, `&` and so on, as simple text

jQuery and DOM

- In some cases, you may want to get the DOM elements from the jQuery result
- You can convert between a jQuery object and a DOM element easily, e.g.:
  - From a jQuery object to a DOM element

```
let domHeader = $("h2")[0];
domHeader.innerHTML = ...;
```

You can get the element you want, e.g. [1], [2]

  - From a DOM element to a jQuery object

```
let jqHeader = $(domHeader);
jqHeader.html(...);
```