



JavaScript 2

Lektion => 1

Utbildare: Mahmud Al Hakim

NACKADEMIN

Lektionstillfällets mål

Mål med lektionen

- Översikt över kursen med kursmål och syfte
- Introduktion till jQuery
- Selektorer
- Caching
- Chaining
- Manipulera innehåll
- Infoga element
- Arbeta med attribut
- **Att läsa: sid. 294–321**

Arbetsmetod

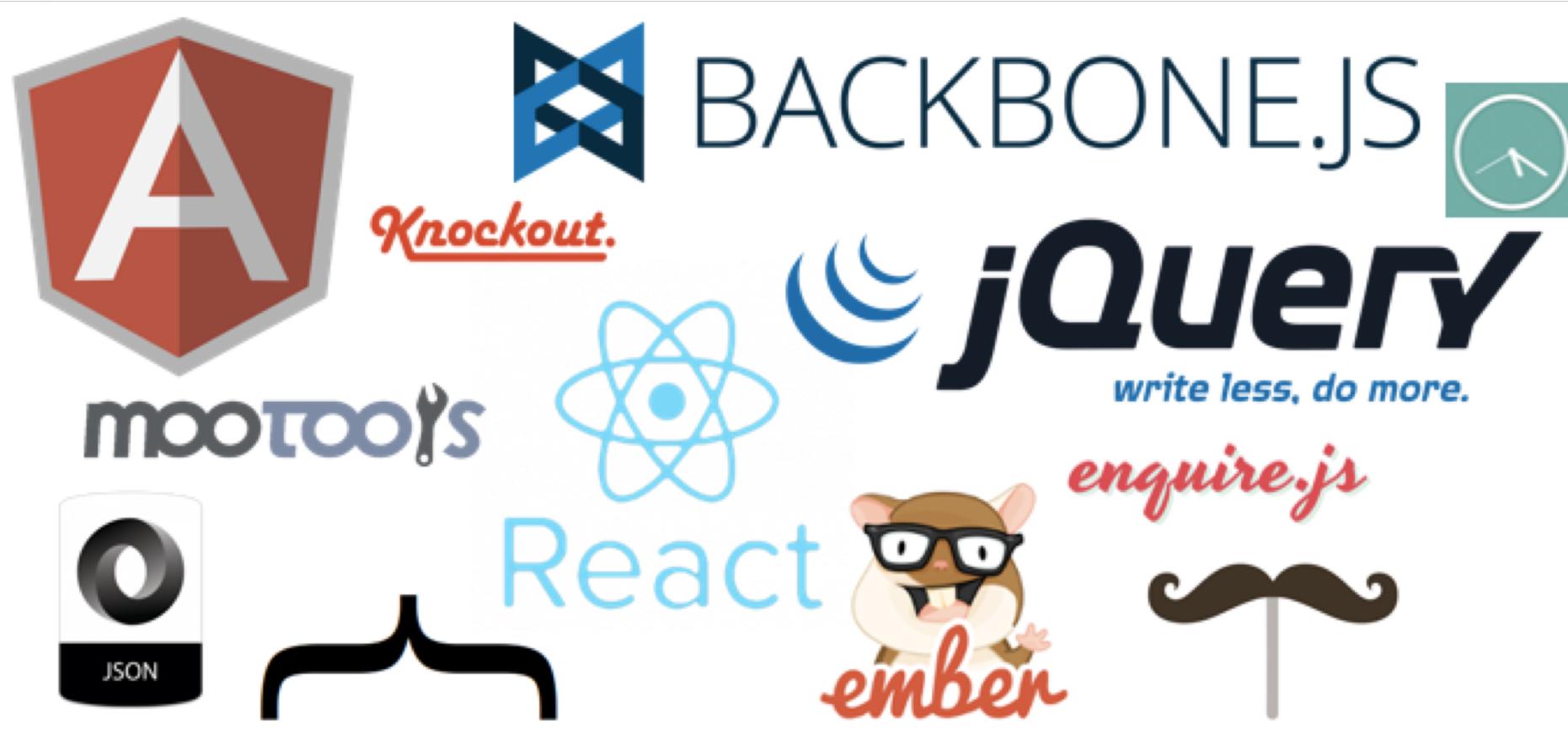
- Teori och praktik varvas under lektionen



NACKADEMIN

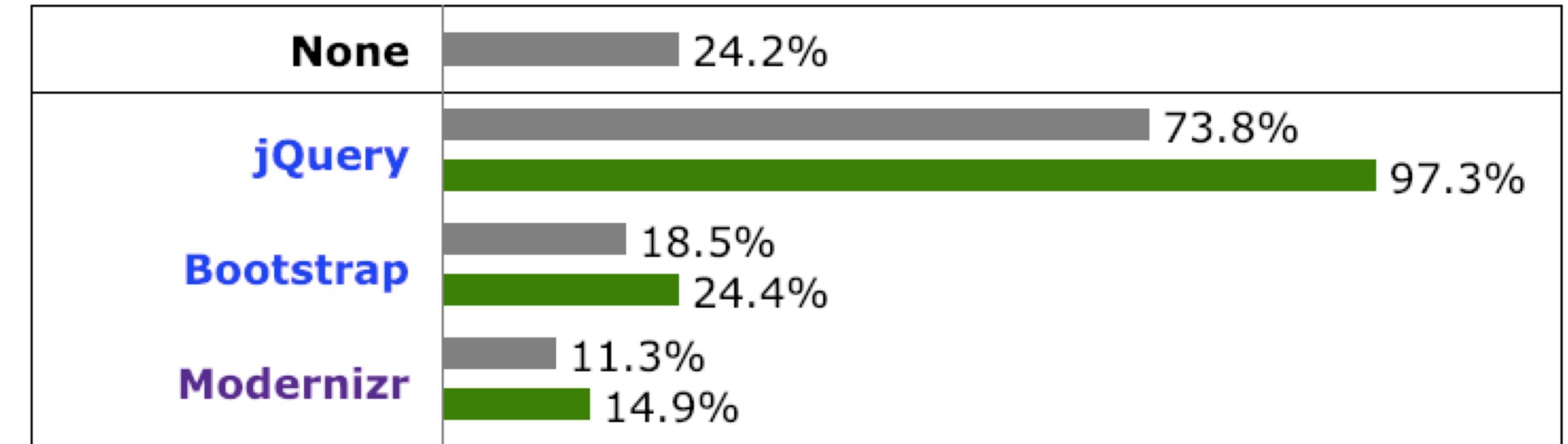
Introduktion

För att förenkla användning av JavaScript och för att undvika återuppfinnandet av hjulet, har erfarna programmerare skrivit ett antal **"bibliotek"** som innehåller färdiga funktioner att använda.



Usage of JavaScript libraries for websites

jQuery is used by 73.8% of all the websites, that is a JavaScript library market share of 97.3%.



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

Vad är jQuery?

- jQuery är ett JavaScript-bibliotek.
- Syftet med jQuery är att förenkla och snabba upp utveckling av webbapplikationer.
- Med jQuery kan du manipulera HTML och CSS.
- Du kan även hantera händelser (events) och skapa animeringar på ett snabbt och effektivt sätt.
- Den första versionen lanserades 2006 av John Resig.
- Officiell webbplats: <http://jquery.com/>



Implementation

- jQuery-biblioteket består av en enda JavaScript-fil.
- Filen kan inkluderas på en webbsida på två olika sätt:

1. Ladda ner filen från <http://jquery.com/download>

Spara filen lokalt och lägg till en script-tagg före </body>

```
<script src="jquery-3.3.1.js"></script>
```

2. Använd CDN (Content Delivery Network)

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"  
>  
</script>
```

One big advantage of using the hosted jQuery from Google or Microsoft:

Many users already have downloaded jQuery from Google or Microsoft when visiting another site. As a result, it will be loaded from cache when they visit your site, which leads to faster loading time. Also, most CDN's will make sure that once a user requests a file from it, it will be served from the server closest to them, which also leads to faster loading time.

jQuery Syntax

`$(selector).action();`

\$ anropar
jQuery

En selektor
(väljare) för att
hitta element

En metod
(event/händelse)
t.ex. click eller hide

The Document Ready Event

- För att vara säker på att jQuery hittar alla element på sidan, så bör du först vänta tills hela dokumentet (inkl. CSS) är nerladdad och färdig (ready).
- Därför placeras jQuery-koden normalt i en speciell funktion som heter ready() så här:

```
$(document).ready(function(){
    // placera din jQuery-kod här...
});
```

Eller så här (enklare syntax, funktionen ready anpas automatiskt)

```
$(function(){
    // placera din jQuery-kod här...
});
```

Hello jQuery!

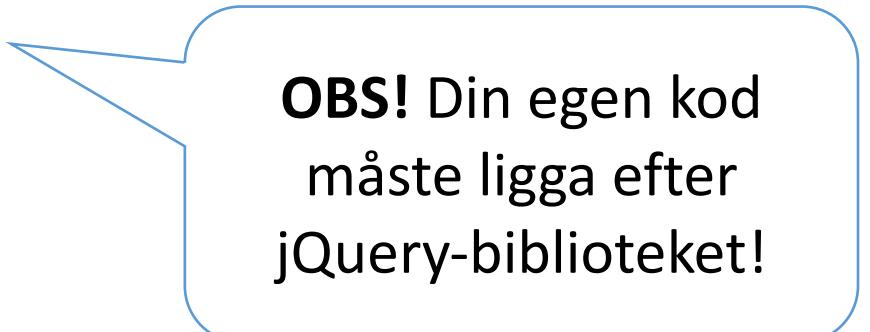
```
<body>
```

```
<script src="jquery-3.3.1.min.js"></script>
<script>
```

```
$(document).ready(function() {
    console.log("Hello jQuery!");
});
```

```
</script>
```

```
</body>
```



OBS! Din egen kod
måste ligga efter
jQuery-biblioteket!

jQuery Selectors

- Med hjälp av en selektor väljs ett eller flera HTML-element.
- Det finns många olika sätt att välja element:
 1. Type Selector
 2. ID Selector
 3. Class Selector
 4. Universal Selector
 5. Groups of selectors
 6. Descendant combinator
 7. Child combinator
 8. Sibling combinator
 9. Subsequent-sibling combinator
 10. This Selector

Type selector (Element Selector)

```
<p>Dölj alla p-taggar</p>
<p>Lorem ipsum dolor sit amet.</p>
```

```
<script>
  $(document).ready(function() {
    $("p").hide();
  });
</script>
```

ID Selector

```
<p id="demo">Dölj detta element</p>
```

```
<p>Lorem ipsum dolor sit amet.</p>
```

```
<script>
  $(document).ready(function() {
    $("#demo").hide();
  });
</script>
```

Class selector

```
<p class="demo">Dölj element som har klassen demo</p>
<p class="demo">Dölj element som har klassen demo</p>
<p>Lorem ipsum dolor sit amet.</p>
```

```
<script>
  $(document).ready(function() {
    $(".demo").hide();
  });
</script>
```

Universal Selector

```
<h1>Dölj alla element</h1>
<p>Lorem ipsum dolor sit amet.</p>

<script>
  $(document).ready(function() {
    $("*").hide();
  });
</script>
```

Groups of selectors – Exempel 1

```
<h1>Ta bort alla h1-element</h1>
<h2>Ta bort alla h2-element</h2>
```

```
<script>
$(document).ready(function() {
    $("h1,h2").hide();
});
</script>
```

Groups of selectors – Exempel 2

```
<h1 id="header">Ta bort elementet som har id="header"</h1>
<p class="demo">Ta bort alla klasser som har class="demo"</p>
<p class="demo">Ta bort alla klasser som har class="demo"</p>

<script>
$(document).ready(function() {
    $("#header, .demo").hide();
});
</script>
```

Descendant combinator

```
<ul>
  <li><a href="#">Ta bort alla länkar som finns i en lista</a></li>
  <li><a href="#">Ta bort alla länkar som finns i en lista</a></li>
</ul>
<a href="#">Behåll alla länkar som finns utanför listan</a> <br>
<a href="#">Behåll alla länkar som finns utanför listan</a>
<script>
$(document).ready(function() {
  $("li a").hide();
});
</script>
```

Child combinator

```
<div>
  <p>Ta bort alla p som finns direkt under en div</p>
  <section>
    <p>Men behåll detta p-element.</p>
  </section>
</div>
<p>och behåll detta p-element.</p>
<script>
$(document).ready(function() {
  $("div > p").hide();
});
</script>
```

Sibling combinator

```
<div>Lorem ipsum dolor sit amet.</div>
<p>Ta bort p som finns direkt efter en div</p>
<p>Behåll detta p-element.</p>
```

```
<script>
  $(document).ready(function() {
    $("div + p").hide();
  });
</script>
```

Subsequent-sibling combinator

```
<p>Detta p ligger före en div (väljs ej).</p>
<div>Lorem ipsum dolor sit amet.</div>
<p>Ta bort alla p som är syskon till en div</p>
<h2>Demo</h2>
<div>
    <p>Detta p är barn till div och inte syskon!</p>
</div>
<p>Ta bort alla p som är syskon till en div</p>
<script>
    $(document).ready(function() {
        $("div ~ p").hide();
    });

```

This Selector

```
<p>Klicka här för att ta bort mig</p>
<p>Klicka här för att ta bort mig</p>
```

```
<script>
$(document).ready(function() {
    $("p").click(function(){
        $(this).hide();
    });
});
</script>
```

jQuery Selector Tester

<https://www.w3schools.com/jquery/trysel.asp>

Click a selector:

```
$("#Lastname")
$(".intro")
$(".intro, #Lastname")
$("h1")
 $("h1, p")
$("p:first")
$("p:last")
$("tr:even")
$("tr:odd")
```

Selector:

```
$(“h1, p”)
```

All `<h1>` elements
and all `<p>`
elements.

Result:

`<h1> Welcome to My
Homepage </h1>`

`<div class="intro">
<p>My name is Donald <span
id="Lastname">Duck </p>`

Caching

- jQuery väljer HTML-element genom att ställa frågor till DOM-trädet.
 - jQuery kommer då att lagrar referenser till dessa element i ett objekt.
 - Det är onödigt att be jQuery ställa samma frågor till DOM.
 - Om du behöver jobba med ett eller flera element flera gånger i en applikation så kan du lagra din fråga i en variabel.
 - Detta kallas "**Caching**".
-
- Tips: <https://youtu.be/sN4QyBIPTv4>

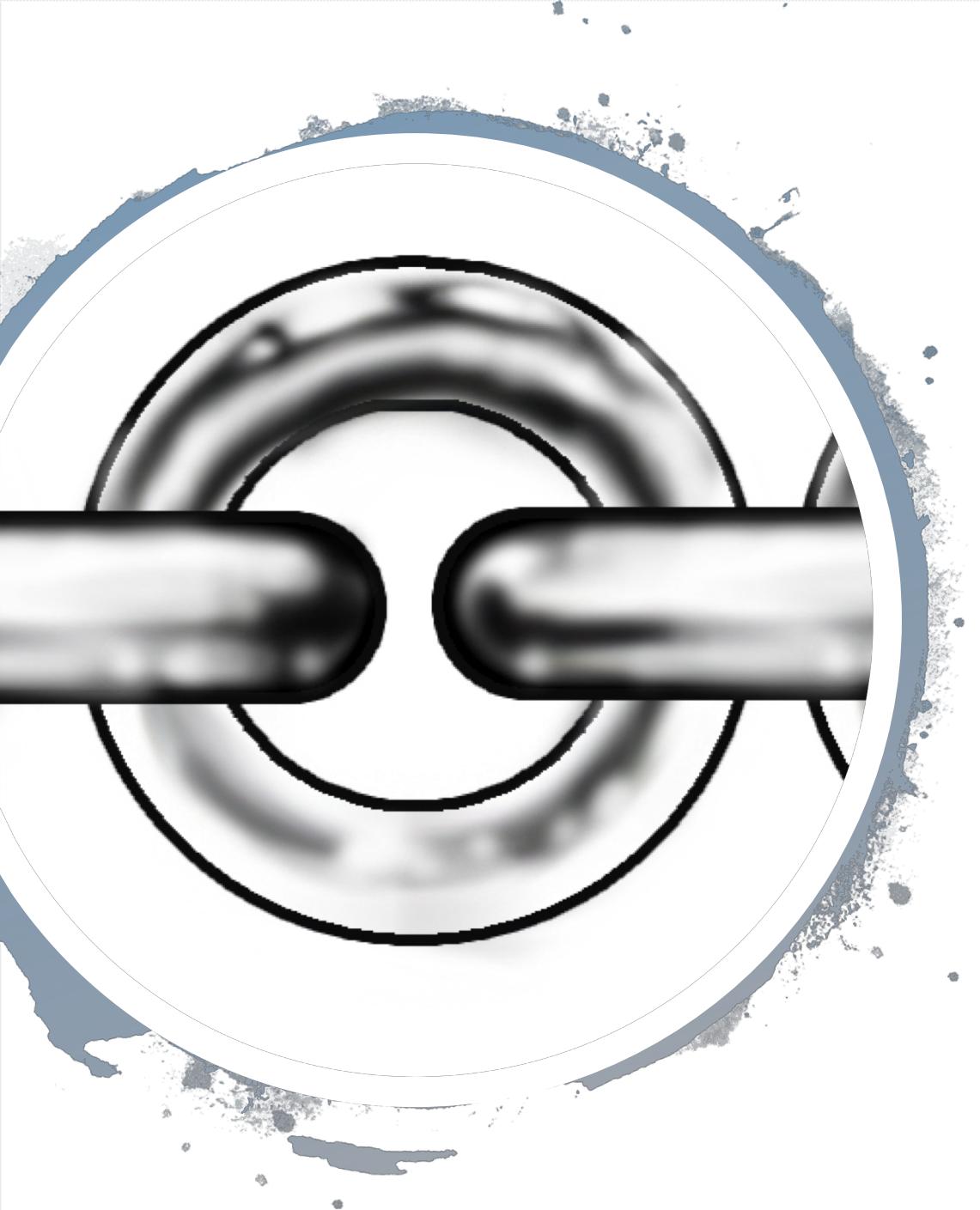
Caching – Exempel

```
<ul>
  <li>One</li>
  <li>Two</li>
  <li>Three</li>
</ul>
```

```
$(document).ready(function() {
  let myListItems = $('li');
  console.log(myListItems.toArray());
});
```

Caching – Exempel fortsättning

```
$document.ready(function() {  
    let myListItems = $('li');  
    myListItems.addClass('hot');  
});
```



Chaining

- Om du behöver använda flera jQuery metoder på samma selektor så kan du koppla dessa metoder efter varandra som en kedja.
- Detta kallas "**Chaining**".

Chaining – Exempel

```
<h1>Chaining</h1>
<script>
  $(document).ready(function() {
    // Med chaining
    $('h1').hide().show(500);

    /* Utan chaining
    $('h1').hide();
    $('h1').show(500); */
  });
</script>
```

Manipulera innehåll

- Med jQuery du kan arbeta med innehåll med hjälp av olika metoder
 1. `html()`
 2. `text()`
 3. `val()`
 4. `remove()`
 5. `hide()`
 6. `empty()`
 7. `replaceWith()`

Metoden html() – Exempel 1

<p>jQuery-metoden **html()**
hämtar text och html-kod.</p>

```
<script>
  $(document).ready(function() {
    $demo = $('p').html();
    console.log($demo);
  });
</script>
```

Metoden html() – Exempel 2

```
<p id="demo"></p>
<script>
$(document).ready(function() {
  $('#demo')
    .html("jQuery-metoden <b>html()</b> sätter text och html-kod");
});
</script>
```

Metoden text() – Exempel 1

```
<p>jQuery-metoden <b>text(</b>
hämtar enbart text (ej html-kod)</p>
<script>
  $(document).ready(function() {
    $demo = $('p').text();
    console.log($demo);
  });
</script>
```

Metoden text() – Exempel 2

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

<script>
$(document).ready(function() {
  $('#demo')
    .text("jQuery-metoden text() sätter enbart text (ej html-kod)");
});
</script>
```

Metoden val() – Exempel 1

```
<input id="demo"  
value="jQuery-metoden val() hämtar text från ett inputfält.">  
  
<script>  
$(document).ready(function() {  
    $demo = $('#demo').val();  
    console.log($demo);  
});  
</script>
```

Metoden val() – Exempel 2

```
<input id="demo" value="" size="30">

<script>
$(document).ready(function() {
  $('#demo')
    .val("jQuery-metoden val() sätter text i ett inputfält.");
});
</script>
```

Metoden remove()

```
<div>
  <p>Metoden remove() tar bort element
    och dess barn från DOM-trädet!</p>
</div>
<script>
  $(document).ready(function() {
    $("div").remove();
  });
</script>
```

Metoden hide()

```
<div>  
<p>Metoden hide() döljer element och dess barn med  
hjälp av inline CSS (style="display: none;")</p>  
</div>  
<script>  
$(document).ready(function() {  
    $("div").hide();  
});  
</script>
```

Metoden empty()

```
<div>
  <p>Metoden empty() tar bort alla barn
  från DOM-trädet!</p>
</div>
<script>
$(document).ready(function() {
  $("div").empty();
});
</script>
```

Metoden replaceWith()

```
<div id="demo">Denna div byts ut mot p.</div>
<script>
$(document).ready(function() {
  $("#demo").replaceWith("<p>Här kommer ett nytt stycke.</p>");
});
</script>
```

Infoga element

.before()

.after()

 item

.prepend()

.append()

append() och prepend()

```
<ul>
  <li>Item</li>
</ul>
<script>
$(document).ready(function() {
  $("ul").prepend("<li>Start</li>");
  $("ul").append("<li>End</li>");
});
</script>
```

before() och after()

```
<ul>
  <li>Item</li>
</ul>
<script>
$(document).ready(function() {
  $("ul").before("<p>Before the list</p>");
  $("ul").after("<p>After the list</p>");
});
</script>
```

Arbeta med attribut

- Följande metoder används för att manipulera attribut
 - attr()
 - removeAttr()
 - addClass()
 - removeClass()
 - toggleClass()

Metoden attr() – Exempel 1

```
<a href="https://webacademy.se">
```

Metoden attr() hämtar värdet som finns i det sökta attributet.

```
</a>
```

```
<script>
```

```
$(document).ready(function() {
```

```
    $demo = $('a').attr("href");
```

```
    console.log($demo);
```

```
});
```

```
</script>
```

Metoden attr() – Exempel 2

```
<a href="https://webacademy.se">
```

Metoden attr() sätter ett nytt värdet i attributet href.

```
</a>
```

```
<script>
```

```
$(document).ready(function() {  
    $('a').attr('href','http://w3schools.com');  
});  
</script>
```

Metoden attr() – Exempel 3

```
<a href="https://webacademy.se">
```

Metoden attr() kan skapa nya attribut t.ex. target.

```
</a>
```

```
<script>
```

```
$(document).ready(function() {
```

```
    $('a').attr('target', '_blank');
```

```
});
```

```
</script>
```

Metoden removeAttr()

```
<p style="font-size:120%;color:red">  
Ta bort attributet style.</p>  
<script>  
$(document).ready(function() {  
    $("p").removeAttr("style");  
});  
</script>
```

Övning – Skapa en enkel inköpslista i jQuery

```
<h1>Vad vill du handla idag?</h1>
<input type="text" id="newItem" autofocus>
<button id="addBtn">Lägg till</button>
<ul id="list"></ul>
```

Vad vill du handla idag?

Ost

Lägg till

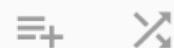
- Bröd
- Mjölk

Tips! Bra kurs om jQuery från Brad Traversy
https://youtu.be/3nrLc_JOF7k



jQuery Crash Course

5 videos • 33,498 views • Last updated on Sep 18, 2016



Traversy
Media

SUBSCRIBED 205K



- 1 jQuery Crash Course [1] - Intro & Selectors
Traversy Media
- 2 jQuery Crash Course [2] - Events
Traversy Media
- 3 jQuery Crash Course [3] - DOM Manipulation
Traversy Media
- 4 jQuery Crash Course [4] - Effects and Animation
Traversy Media
- 5 jQuery Crash Course [5] - Ajax
Traversy Media

Summering av dagens lektion

- Vi har gått igenom grunderna i jQuery.
 - Selektorer
 - Caching
 - Chaining
 - Manipulera innehåll
 - Infoga element
 - Arbeta med attribut
 - **Att läsa: sid. 294–321**
- Reflektioner kring dagens lektion?

NACKADEMIN

Framåtblick inför nästa lektion

- Vi ska jobba vidare med jQuery
 - Manipulera CSS
 - Händelser (events)
 - Effekter
 - Animeringar
 - Traversing DOM
 - Formulär

NACKADEMIN