

# JQUERY – BASICS ...

---

Redigeret 24. november 2018

## Indhold

|   |    |
|---|----|
| jQuery - et javascriptbibliotek - til download og/eller link.....       | 1  |
| Download - og link til en downloadet version af jQuery-biblioteket..... | 2  |
| Linke til en ekstern CDN-version af jQuery-biblioteket .....            | 2  |
| Opbygning af jQuery .....   | 3  |
| Selectors .....   | 4  |
| Events.....   | 5  |
| Actions .....   | 5  |
| Effekter .....  | 5  |
| Tasteeksempel 1: Hide - show .....                                      | 6  |
| Tasteeksempel 2: Hide - show.....                                       | 6  |
| Tasteeksempel 3: Toggle.....  | 7  |
| Tasteeksempel 4: SlideToggle .....                                      | 8  |
| Tasteeksempel 5: FadeOut.....   | 9  |
| HTML og CSS.....  | 10 |
| HTML.....   | 10 |
| Tasteeksempel 6: Html og text.....                                      | 10 |
| CSS.....  | 12 |
| Tasteeksempel 7: Css.....   | 12 |

## jQuery - et javascriptbibliotek - til download og/eller link

jQuery er et **javascript-bibliotek**, som gør det nemmere for udvikleren at kode javascript.

Koderne i biblioteket gør, at du nemt - med meget lidt kode - kan lave en masse JavaScript-dynamik i dit webprojekt. Desuden rummer jQuery den fordel, at din kode er "cross-browser-kompatibel" – altså at den virker i flest mulige browsere og i flest mulige versioner. Og skulle der være en browser, som ikke forstår dine koder, sørger jQuery for at fange "fejlen" og oversætte den til en fornuftig håndtering af problemet (fx en besked til brugeren om at opdatere sin browser). Du skal altså ikke bekymre dig så meget om koderne nu virker i alle browsere (det håndterer jQuery for dig) – du skal bare få det til at fungere i den browser, som du selv tester det i.

For at kunne bruge jQuery i din løsning, er du nødt til at gøre jQuery-biblioteket tilgængeligt fra den eller de sider, som har brug for det. Biblioteket fylder trods alt en del (ca. 80-90 kB i den

minimerede/compressed version) – fra en smartphone med dårlig forbindelse kan det være noget af en mundfuld, så du bør derfor overveje:

- Hvis du bruger **jQuery på mange sider**: Så bør jQuery-biblioteket måske indsættes på **Layout-pagen** (så biblioteket er tilgængelig på alle sider, der bruger det Layout).
- Hvis du kun bruger **jQuery på nogle få sider**: Så bør jQuery-biblioteket måske kun indsættes i "head" eller en "RenderSection" på disse sider.

**Desuden bør du overveje:**

- Om du skal linke til den komprimerede/minimerede version (den der hedder noget med .min) – eller om det skal være den fulde version.

Jeg foreslår, at du bruger den fulde version, når du udvikler (aht. Intellisense/opslag), og den minimerede version, når du lægger det online.

- Om du skal downloade og linke til en lokal version af jQuery-biblioteket på din egen server – ELKER om du skal linke til en ekstern version på et CDN (Content Delivery Network) hos fx Google eller Microsoft – se næste afsnit:

## Download - og link til en downloadet version af jQuery-biblioteket

1. Download (nyeste) jQuery-fil og lægge den i dit eget websted.
2. Link til den (i <head>) fra de sider (eller bare fra MasterPage), hvor du skal bruge jQuery:

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

Du kan finde nyeste downloads fra jQuerys eget site: <http://jquery.com/download/>

**NB!** At jQuery-filen hedder noget med "**xxx.min.js**" betyder, at det er en **minified** (komprimeret) version af jQuery-biblioteket. I en minified version er alle linjeskift, whitespace, kommentarer mv. barberet væk for at minimere filen mest muligt. Det betyder, at den fylder væsentligt ca. en tredjedel af den fulde version, men til gengæld er den umulig at forstå/læse/arbejde i. **Som nævnt tidligere, bør du bruge den minimerede version, når dit site ligger online (så brugeren skal hente så lidt som muligt), mens det kan være en fordel at bruge den fulde version, når du koder jQuery/JavaScript for at opnå intellisense/kodeforslag.**

## Linke til en ekstern CDN-version af jQuery-biblioteket

Du kan vælge at benytte dig af et eksternt placeret jQuery-bibliotek. Se fx muligheder her:

<http://jquery.com/download/#other-cdns>

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

**Fordelen** ved at trække på en **ekstern** version af jQuery-biblioteket er:

- At hentning af biblioteket ikke belaster din egen server
- At din webside og biblioteket kan hentes samtidig (fordi det hentes fra to forskellige servere)
- At der er en vis chance for, at jQuery-filen allerede er cachet på brugerens computer

**Ulemperne kan være:**

- At den eksterne jQuery-fil ikke er tilgængelig
- At hvis en forsinket ekstern server betyder, at brugeren skal vente på at jQuery-filen bliver loadet

**OBS!** Pas på med at linke til "latest" version af jQuery-biblioteket – du kan risikere, at din websides jQuery fejler/ikke virker, hvis du f.eks. bruger jQuery, som netop er ændret i den "latest" (nyeste) version.

## Opbygning af jQuery

jQuery kan groft inddeles i:

- **Selectors** – **HVILKET** element skal der pilles ved - [http://w3schools.com/jquery/jquery\\_ref\\_selectors.asp](http://w3schools.com/jquery/jquery_ref_selectors.asp)
- **Events** – **HVORNÅR** skal det ske - [http://w3schools.com/jquery/jquery\\_ref\\_events.asp](http://w3schools.com/jquery/jquery_ref_events.asp)
- **Action** – **HVAD** skal der ske /ændres/styles etc.:
  - **Effekt** ... [http://w3schools.com/jquery/jquery\\_ref\\_effects.asp](http://w3schools.com/jquery/jquery_ref_effects.asp)
  - **Html og css** ... [http://w3schools.com/jquery/jquery\\_ref\\_html.asp](http://w3schools.com/jquery/jquery_ref_html.asp)

Grund-syntaksen er følgende:

```
$(selector).action() ... altså fx:
```

```
$(selector).event( function() {  
  
    $(selector).action()  
  
});
```

Se nogle eksempler her: [https://www.w3schools.com/jquery/jquery\\_syntax.asp](https://www.w3schools.com/jquery/jquery_syntax.asp)

- Hvilken **event** skal sætte det hele i gang (f.eks. klik, mouseover osv.)
- Hvilken **selector** skal udvælges (f.eks. en ID, en CLASS eller et html-tag f.eks. alle sidens P-tags)
- Hvilken **action** skal ske (f.eks. udskiftning af et billede, fadeln, ændring af baggrundsfarve, udskiftning af en tekst, indsættelse af en popup-div osv.)

jQuery-metoderne skal være "pakket ind" i følgende ydre funktion **sådan her**:

```
$(document).ready(function() {  
  
    // jQuery-metoder her ... hvilket, hvornår, hvad ...  
  
});
```

**ELLER sådan her (en forkortet - men lige så rigtig) version:**

```
$(function () {  
  
    // jQuery-metoder her ... hvilket, hvornår, hvad ...  
  
});
```

... det er for at være sikker på, at alt på siden/dokumentet (html mv.) er indlæst/loadet, inden du begynder at henvise til det og ændre på det.

## Selectors

Handler om at **udvælge** – selecte – **HVILKE** elementer, du gerne vil have gjort et eller andet ved. Skal du f.eks. have fat på en bestemt **id**? Eller alle elementer med en bestemt **class**? Eller et bestemt html-tag – f.eks. alle dine p-tags?

Du kender det allerede fra **css**, hvor det jo ser sådan her ud:

```
p { color: "#F00"; font-size: "10px"; }
```

Her er **p** din **selector** ... altså det, der fortæller (udvælger/udpeger), hvilket element, der skal pilles ved ... (**color** er den **property/egenskab**, du vil pille ved ... og **#F00** er den **value/værdi**, egenskaben skal have).

Selectors i jQuery ligner meget det, du kender fra netop css. Her er nogle eksempler:

| Selector:      | Forklaring – hvad vælger den:          |
|----------------|--|
| \$( "*")       | Alle tags/alt                          |
| \$( "p")       | Alle <p>                               |
| \$( "p.intro") | Alle <p> som har en class="intro"      |
| \$( "p#intro") | Alle <p> med id="intro"                |
| \$( ":hover")  | Alt som er "hover"                     |
| \$( ":button") | Alle <button> og <input type="button"> |

|                                   |  |
|-----------------------------------|--|
| <code>\$("ul li:first")</code>    | Første<li> i en <ul> (men ikke alle efterfølgende li'er) |
| <code>\$("[src\$='.jpg']")</code> | Alt som har en attribut "src", som ender med ".jpg"      |

Se flere muligheder – og eksempler – her: [http://w3schools.com/jquery/jquery\\_ref\\_selectors.asp](http://w3schools.com/jquery/jquery_ref_selectors.asp)

## Events

En event er en angivelse af, **HVORNÅR** herlighederne skal "fyres af". Altså – hvilken hændelse eller begivenhed skal indtræffe for at et eller andet sker. Det kan f.eks. være:

| Event:  | Forklaring – hvornår sker det. Når ...                   |
|---|--|
| <code>\$(document).ready(function)</code>     | dokumentet er indlæst/loadet                             |
| <code>\$(selector).click(function)</code>     | der bliver <b>klikket</b> på det udvalgte element        |
| <code>\$(selector).dblclick(function)</code>  | der bliver <b>dobbeltklikket</b> på det udvalgte element |
| <code>\$(selector).focus(function)</code>     | der er <b>fokus</b> på det udvalgte element              |
| <code>\$(selector).mouseover(function)</code> | der er <b>mouseover</b> på det udvalgte element          |

Se flere muligheder – og eksempler – her: [http://w3schools.com/jquery/jquery\\_events.asp](http://w3schools.com/jquery/jquery_events.asp)

## Actions

Handler om, **HVAD** der skal ske ... det kan være **effekter**, **html** og/eller **css**:

### Effekter

Er de indbyggede effekter, som jQuery kan udløse. F.eks.:

| Effekter:                               | Forklaring – hvad sker der. Effekten ...     |
|---|--|
| <code>\$(selector).hide()</code>        | skjuler det valgte                           |
| <code>\$(selector).show()</code>        | viser det valgte                             |
| <code>\$(selector).toggle()</code>      | skifter mellem vis og skjul på det valgte    |
| <code>\$(selector).slideDown()</code>   | "glider ned"                                 |
| <code>\$(selector).slideUp()</code>     | "glider op"                                  |
| <code>\$(selector).slideToggle()</code> | skifter selv mellem at glide op og ned       |
| <code>\$(selector).fadeIn()</code>      | fader ind                                    |
| <code>\$(selector).fadeOut()</code>     | fader ud                                     |
| <code>\$(selector).fadeTo()</code>      | fader til en given opacity                   |
| <code>\$(selector).fadeToggle()</code>  | Skifter mellem at være fadet ind og fadet ud |

Se flere muligheder – og eksempler – her: [http://w3schools.com/jquery/jquery\\_ref\\_effects.asp](http://w3schools.com/jquery/jquery_ref_effects.asp)

## Tasteeksempel 1: Hide - show

Opret en html-fil (kald den f.eks. tasteeks1.html) med bl.a. følgende indhold – test at det virker:

```
<script src="http://code.jquery.com/jquery-latest.min.js" type="text/javascript"></script>
<script type="text/javascript">
    $(document).ready(function () {

        $("p").click(function () {
            $(this).hide();
        });

    });
</script>
/head>
body>

<p>Klik her ... og jeg forsvinder</p>
```

### Eksempel 1

Funktionen gør, at når der klikkes på et <p>, så forsvinder det (this) ... altså <p>-tag'et selv. Opret gerne flere <p>-tags på siden – og klik på dem ... de skulle gerne forsvinde.

*NB! Du vil sikkert opleve, at du - hvis du lægger dit javascript i en selvstændig js-fil - ikke får autocomplete/intellisense (hjælp til at skrive dit jQuery). Det kan du rette op på, hvis du refererer til jQuery-biblioteket på denne måde øverst i din js-fil:*

```
/// <reference path="jquery-1.8.2.min.js" />
```

Den nemmeste måde at gøre det på, er at trække din jQuery-fil fra Solution Explorer ind på js-siden ... så bliver reference-path'en automatisk lavet for dig.

## Tasteeksempel 2: Hide - show

Opret nu en html-fil (tasteeks2.html) med følgende indhold – og test også at det virker, inden du går videre:

```

<script src="http://code.jquery.com/jquery-latest.min.js" type="text
<script type="text/javascript">
    $(document).ready(function () {

        $("#forsvind").click(function () {
            $("p").hide();
        });

        $("#kom").click(function () {
            $("p").show();
        });

    });
</script>
</head>
</body>

<p>Klik på en knap herunder ...</p>
<input type="button" id="forsvind" value="Forsvind tekst" />
<input type="button" id="kom" value="Kom tilbage tekst" />

```

#### Eksempel 2

Her har du to funktioner – som holder øje med hver deres "id". Når der bliver klikket på en af de to "id"'er, vil alle <p> på siden enten forsvinde eller blive vist – alt efter hvilken knap, der blev klikket på.

Prøv nu at **ændre koden** til:

```

$("#forsvind").click(function () {
    $("p").hide(1000);
});

$("#kom").click(function () {
    $("p").show(2000);
});

```

#### Eksempel 2 - udvidet

Hvad sker der så? Kan du regne ud, hvad tallene i paranteserne betyder?

### Tasteeksempel 3: Toggle

Det her er en anden måde at gøre det på – **opret en ny html-side** (tasteeks3.html) og prøv det ...

```

<script type="text/javascript">
    $(document).ready(function () {

        $("#skift").click(function () {
            $("p").toggle(1000);
        });

    });
</script>
<body>

    <p>Klik på en knap herunder ...</p>
    <input type="button" id="skift" value="Vis og skjul" />

```

### Eksempel 3

Og prøv så at tilføje det her:

```

$(document).ready(function () {

    $("#skift").click(function () {
        $("p").toggle(1000);
        $(this).val("Der er klikket her");
    });

});

```

### Eksempel 3 - udvidet

**Kig på eksemplerne – prøv at regne, hvad der sker og hvorfor ... spørg din underviser, hvis du vil have det forklaret!**

## Tasteeksempel 4: SlideToggle

Prøv nu denne her – **gerne på en ny html-side** (tasteeks4.html):



```

<script src="http://code.jquery.com/jquery-latest.min.js" type="text/
<script type="text/javascript">
    $(document).ready(function () {

        $("#btn").click(function () {
            $("#mitpanel").slideToggle("slow");
        });

    });
</script>
<body>

<div id="mitpanel" style="background-color: Aqua;">
    <h1>Dette er et lille "panel"</h1>
    <p>Ja, det er jo i grunden baren div ... </p>
</div>

<input type="button" id="btn" value="Klik" />

```

Eksempel 4

## Tasteeksempel 5: FadeOut

Opret endnu en html-side (tasteeks5.html) – og prøv kræfter med denne:

```

<script src="http://code.jquery.com/jquery-latest.min.js" type="t
<script type="text/javascript">
    $(document).ready(function () {

        $("#mitpanel").click(function () {
            $(this).fadeOut(4000);
        });

    });
</script>
<body>

<div id="mitpanel" style="background-color: Aqua;">
    <h1>Klik i "panelet"</h1>
    <p>Så lover jeg at forsvinde stille og roligt ....</p>
</div>

```

Eksempel 5

Prøv også at erstattet fadeOut-eventen herover med denne:

```
$(this).fadeTo("slow", 0.25);
```

## HTML og CSS

Ligesom du med klassisk javascript kan ændre på html (innerHTML) og css (f.eks. `xx.style.backgroundColor`) kan du også med jQuery ændre på html, css mv. Endda meget nemt og med mange muligheder:

### HTML

HTML-actions handler om at ændre i html'en:

| HTML-actions                     | Forklaring – hvad sker der. Action ...   |
|----------------------------------|--|
| <code>\$(selector).html()</code> | indsætter/overskriver html i det valgte  |
| <code>\$("p").append()</code>    | tilføjer html i slutningen af det valgte |
| <code>\$("p").before()</code>    | tilføjer html i før det valgte           |
| <code>\$("p").after()</code>     | tilføjer html i efter det valgte         |

Se flere muligheder – og eksempler – her: [http://w3schools.com/jquery/jquery\\_ref\\_html.asp](http://w3schools.com/jquery/jquery_ref_html.asp)

### Tasteeksempel 6: Html og text

På en ny html-side (tasteeks6.html) - prøv denne her:

```
$(document).ready(function () {  
  
    $(":button").click(function () {  
        $("p").html("En ny tekst");  
    });  
  
});  
</script>  
</>  
<?>  
  
<h1>En overskrift</h1>  
<p>Første linje med tekst.</p>  
<p>Anden linje med tekst.</p>  
  
<input type="button" value="Klik" />
```

#### Eksempel 6

Læg mærke til den specielle selector – `":button"` – kig i oversigten over selectors herover, hvad den betyder.

Prøv nu at udvide den med følgende – hvor du føjer lidt html-koder på:

```

$(document).ready(function () {

    $(".button").click(function () {
        $("p:first").html("En <b>fed</b> ny <i>kursiv</i> tekst");
    });

});
</script>
</td>
</tr>
</table>

<h1>En overskrift</h1>
<p>Første linje med tekst.</p>
<p>Anden linje med tekst.</p>

<input type="button" value="Klik" />

```

#### Eksempel 6 - udvidet

Test den ...

Og prøv nu det her:

```

$(document).ready(function () {

    $(".button").click(function () {
        $("p:first").text("En <b>fed</b> ny <i>kursiv</i> tekst");
    });

});

```

#### Eksempel 6 - test

... kan du se, at der er forskel på, om du tilføjer text eller og html? Ret den tilbage, når du har set resultatet.

Prøv også denne – evt. samme side med en anden knap:

```

$(document).ready(function () {

    $(".button").click(function () {
        $("p:first").before("FORREST");
        $("p:first").after("BAGEST");
    });

});

```

#### Eksempel 6 – ekstra 1

Her er to andre muligheder – test dem og se forskellen:

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

    $(".button").click(function () {
        $(".p:first").append("FORREST");
        $(".p:first").prepend("BAGEST");
    });

});
```

Eksempel 6 – ekstra 2

## CSS

Man kan selvfølgelig også manipulere med stylingen/css'et:

| CSS-actions   | Forklaring – hvad sker der. Action ...                 |
|---|--|
| <code>\$("p").css("color")</code>   | Returnerer tekst-farven (color) på p-elementet ... GET |
| <code>\$("p").css("color", "red")</code>                                    | Ændrer på tekstfarven mv. ... SET                      |
| <code>\$("p").css({"color": "white", "background-color": "#98bf21"})</code> |  |

Prøv at taste dette eksempel:

## Tasteeksempel 7: Css

På en ny html-side (tasteeks7.html):

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

    $("h1").mouseover(function () {
        $(this).css("background-color", "yellow");
    });

});
</script>
<h1>En overskrift</h1>
```

Eksempel 7

Udvid med dette her – og tilpas den selv med noget styling efter eget valg:

```

$(document).ready(function () {
    $("h1").mouseover(function () {
        $(this).css({ "color": "white", "background-color": "#98bf21", "padding": "10px" });
        $(this).text("Ny stylet tekst");
    });
});

script>
>

<h1>En overskrift</h1>

```

#### Eksempel 7 - udvidet

Hvad med funktioner, if-else, loop osv.?

jQuery ER JavaScript, så alt hvad du kan i JavaScript, kan du også i jQuery. Så brug funktioner, betingelser, loops osv., som du plejer.

Eksempel på betingelse:

```

$("h1").click(function () {
    ($(this).text() === "Overskrift1") ? $(this).text("NyOverskrift1") :
    $(this).text("Overskrift1");
});

```

... som bare er en forkortet måde at skrive det her på:

```

$("h1").click(function () {
    if ($(this).text() === "Overskrift1") {
        $(this).text("NyOverskrift1")
    } else {
        $(this).text("Overskrift1");
    }
});

```

Så er du vist klar til at prøve lidt selv på egen hånd 😊