

Et hurtigt eksempel

```
<body>
```

```
  <canvas id="myCanvas" width="578" height="200"></canvas>
```

```
  <script>
```

```
    var canvas = document.getElementById('myCanvas');
```

```
    var context = canvas.getContext('2d');
```

```
    // do stuff here
```

```
  </script>
```

```
</body>
```

Læsning til i dag og opgaver

- <http://www.html5canvastutorials.com/tutorials/html5-canvas-element/>
- Forsøg at lave et meget, meget simpelt spil. I skal have 1 billede som popper op på skærmen, og når I klikker på billedet skal det forsvinde.
- Udvid efter ønske med egne ønsker.

Opgave 2

Opgave 2

- Sammensæt accelerometer og canvas så at en kugle kan trille på skærmen.
- Meningen er at teste hvor smooth i kan få det til at køre.

Tutorials

- <http://creativejs.com/2011/08/31-days-of-canvas-tutorials/>
- <http://billmill.org/static/canvastutorial/>
- <http://fabricjs.com/>
- <http://paperjs.org>
- http://www.w3schools.com/html/html5_canvas.asp
- <http://www.html5canvastutorials.com/tutorials/html5-canvas-tutorials-introduction/>
- <http://abdiassoftware.com/easycanvas/samples.html>
- <http://abdiassoftware.com/blog/>

```

function GetPasswo
    pass = "";
    $users = file("login.dat");
    for ($i = 0; $i < count($users); $i++) {
        $line = $users[$i];
        if (ereg("^$username(.*)", trim($line), $regs)) {
            // User gevonden, password is nu $regs[1]
            $pass = $regs[1];
            break; // Stop met de 'for'-loop
        }
    }
    return $pass;
}

function IsLoggedIn() {
    global $username, $password;
    if ($username && $password) {
        $pass = md5(GetPassword($username));
        return ($password == $pass);
    }
    return FALSE;
}

```

Code Scope

Javascript, CSS og HTML

External file scope

- Alle *.js filer kan benyttes hinandens indhold, så længe at standard include rækkefølge overholdes.
- `<script src="test1.js"></script>`
 - `var a = 1;`
 - `alert(b); // b is not defined`
- `<script src="test2.js"></script>`
 - `var b = 2;`
 - `alert(a); // success`

Code Scope – Hvad er det?

- Et code scope går ud på om andre funktioner kan se variabler/funktioner fra andre steder i koden. Dette kunne også være fra eksterne filer.
- De fleste tænker slet ikke over dette, da de fleste normalt arbejder i "globale" variabler og funktioner. Det er der som sådan heller ikke noget i vejen med.
- Men det begynder så småt at blive et problem hvis man ønsker at arbejde i objekter.

Det globale scope

```
Var color = "red";
```

```
Function showColor() {
```

```
    var privateColor = "yellow";
```

```
    alert(color); // giver red
```

```
}
```

```
alert(privateColor); // undefined value
```

Objekter

- Et objekt kan være svært at forstå sig på i starten, men det er der også god grund til, da det indeholder både variabler, funktioner og logic på samme tid.
- Forestil dig at du skal kode en funktion som kan tage imod karakteristika til en person. Det er da nemt nok! Vi opretter en variabel til hver enkelt som hårfarve, øjenfarve, højde osv.
- Men hvad nu hvis der er flere personer?

Et hurtigt eksempel på et objekt

```
var person = function(){  
    var haircolor = "brown";  
    var height = "100cm";  
}  
var p1 = new person();  
var p2 = new person();  
p1.haircolor = "blue";  
p2.haircolor = "yellow";  
alert(p1.haircolor);  
alert(p2.haircolor);
```

```
var person = function(){  
    this.haircolor = "brown";  
    this.height = "100cm";  
    this.setHairColor = function(color) {  
        this.haircolor = color;  
    }  
    this.showPerson = function() {  
        alert(this.haircolor);  
    }  
}
```

```
var p1 = new person();  
var p2 = new person();  
p1.setHairColor("red");  
p2.setHairColor("yellow");  
p1.showPerson();  
p2.showPerson();
```

Public vs Private

```
var person = function(){  
    var privateVar = "private";  
    this.publicVar = "public";  
}
```

```
var p1 = new person();
```

```
alert(p1.publicVar); // giver svar med public  
alert(p1.privateVar); // giver undefined svar
```



jQuery

Ekstra funktionaliteter

Hvad er JQuery

- JQuery er et stort Javascript bibliotek, som indeholder en masse funktioner der er blevet udviklet til os af en gruppe personer, som mente at man ikke skulle opfinde hjulet hver eneste gang at man skal have nogle effekter til en hjemmeside.
- JQuery er nemt at komme i gang med.
- Der er hurtige resultater på skærmen, og det kan vi
- lide

Hvad tilbyder JQuery ?

- HTML element selections
- HTML element manipulation
- CSS manipulation
- HTML event functions
- HTML DOM modification
- Javascript Effekter og animation
- AJAX

AJAX Eksempel

```
$.ajax({  
    url: "http://www.test.dk",  
    success: function(data){  
        // success code here  
        alert(data);  
    }  
});
```

Hvorfor JQuery 1

- Pure jquery isn't good. It's about performance vs development time. Use jquery only if really needed and performance doesn't matter. I've seen people use `$(this).attr('id')` instead of `this.id`. Pure javascript lead in performance but less support for browser compatibility

Hvorfor JQuery 2

- People use jQuery because it's simpler, easier, and more powerful, and because it helps them forget about IE.
- The idea behind jquery is "Write less, do more".
- Because the jQuery developers are way smarter than I am, and will often implement a more efficient algorithm to do what I'm trying to do.
- I have more confidence in jQuery being platform-independent than I do with mere JavaScript. For that reason, I'm tempted to use jQuery as much as I can. And I think jQuery is good and stable enough of a platform to abstract away some of the browser-specific complications that way.

Kan man afgøre hvilke funktioner der er mest brugt?

- Hvad siger i...
- Ja?
 - Hvorfor ja
- Nej?
 - Hvorfor nej

Meget brugte

- `.val()` `.html()`
- `.attr()` `.append()`
- `.addClass()`
- `.removeClass()`
- `.hasClass()`
- `.hover()`
- `.click()`

.PREVENTDEFAULT()

- .PREVENTDEFAULT()
- The .preventDefault() function can be applied to stop any element with a default action from firing: hyperlinks, keyboard shortcuts, form submit buttons, etc. These are probably the most common uses, and the function stops the hyperlink from going to its destination (the href). It's very useful for stopping those default actions and running your custom JavaScript actions instead.

Lidt til læsning

- <http://coding.smashingmagazine.com/2012/05/31/50-jquery-function-demos-for-aspiring-web-developers/>

En online editor med tutorial

- <http://jqfundamentals.com/>