

ECMAScript

også kendt som Javascript

Kenneth Geisshirt

`http://kenneth.geisshirt.dk/`

SSLUG — 18. august 2009

Hvad er Javascript?

Sproget

Værktøjer

Moderne biblioteker

Ikke kun web

Få mere at vide

Hvor mange har skrevet

- ▶ assembler?
- ▶ proceduralt (Comal-80, Pascal, C)?
- ▶ objekt-orienteret (Delphi, C++, Java, C#)?
- ▶ funktion-orienteret (Haskell, Lisp, ML)?

Og hvor mange har prøvet Javascript?

Javascript har

- ▶ syntaks som C
- ▶ alm. kontrolstrukturer som kendt fra procedurale sprog
- ▶ klasse-løse objekter
- ▶ funktioner som førsteangsobjekter
- ▶ stor udbredelse
 - ▶ webbrowsere (lynx undtaget)
 - ▶ udvidelser (Google Widgets, Apple Dashboard Widgets)
 - ▶ scripting (Adobe-produkter, OpenOffice, Unity 3D)
 - ▶ applikationer (Palm Pre)

Se også: <http://en.wikipedia.org/wiki/JavaScript>

Variable

- ▶ Bogstav først, forskel på store og små bogstaver, tal, underscore
- ▶ Erklæres med `var` (`var a;`)
- ▶ Typer:
 - ▶ boolsk (`true`, `false`)
 - ▶ tal (`faq = 42`, `pi = 3.1415`)
 - ▶ strenge (`msg = "hello, world"`, `myLUG = 'SSLUG'`)
 - ▶ tabeller/arrays (`a = [1,2,3]`, `a[2]`)
 - ▶ objekter (`person = {navn: "Kenneth", skonr: 42}`)
 - ▶ funktioner

Eksempler på variable (tal, streng, tabel, objekt)

De sædvænlige kontrolstrukturer

- ▶ Blokke er omsluttet af { og }
- ▶ `if () { ... } else { ... }`
- ▶ `while () { ... }`
- ▶ `for (;;) { ... }`
- ▶ `switch () { case: ... break }`
- ▶ `try { ... } catch () { ... }`
- ▶ `throw`
- ▶ `return`
- ▶ `break`

Eksempler på løkker

```
s=0;
i=1;
while (i<=10) {
    s += i;
    i++;
}
```

```
s = 0;
for(i=1; i<=10; i++) {
    s += i;
}
```

Objekter

- ▶ Javascript er klasseløst objekt-orienteret
- ▶ Ny objekter skabes fra eksisterende (prototyping)
- ▶ Grund-objektet er `Object.prototype`
- ▶ `this` refererer til det aktuelle objekt

```
var person = {  
  navn: {  
    fornavn: "Kenneth",  
    efternavn: "Geisshirt"  
  },  
  sknr: 42  
};
```

```
person.navn.efternavn;  
person["sknr"];
```

Instanser af objekter

```
person = {navn: ""};
```

```
Object.create = function(obj) {  
  var F = function() {};  
  F.prototype = obj;  
  return new F();  
};
```

```
var mig = Object.create(person);  
mig.navn = "Kenneth";  
mig.kaldenavn = "kneth";  
var dig = Object.create(person);  
dig.navn = "Pia";
```

Funktioner

- ▶ Funktioner er funktionelle (?!) blokke af kode
- ▶ Funktioner er førsteklasse objekter
- ▶ Variabel antal argumenter \Rightarrow alle argumenter findes som et array ved navn `arguments`
- ▶ Navngivne argumenter er understøttet
- ▶ Rekursion er muligt — og ofte effektivt til DOM-manipulation

Eksempler på funktioner

```
function f1(besked) {  
    alert(besked);  
}  
f1("Hello, World");
```

```
function f2() {  
    s = "";  
    for(i=0; i<arguments.length; i++) {  
        s += arguments[i];  
    }  
    alert(s);  
}  
f2("Hello, ", "World");
```

Funktioner som objekter

- ▶ `Function`-objektet er basis for alle funktioner
- ▶ Funktioner kan manipuleres som objekter
- ▶ Objekter kan have “metoder”
- ▶ Objekter giver mulighed for indre funktioner
- ▶ Bruges ofte til callbacks ved f.eks. AJAX-kald

Eksempler på funktioner som objekter

```
inc = function (i) {  
    return i+1;  
}  
inc(41);
```

Eksempler på funktioner som objekter

```
obj = { x: 0,  
        set: function(x) {  
            this.x = x;  
        },  
        inc: function () {  
            this.x++;  
        },  
        get: function () {  
            return this.x;  
        }  
    };
```

```
obj.set(41);  
obj.inc();  
obj.get();
```

Closures

- ▶ Indre funktioner har adgang til den ydre funktions variable
- ▶ også når den indre funktion lever længere end den ydre
- ▶ På den måde kan vi gemme konteksten (closure) f.eks. ved callback

Eksempel på closure med prototype-biblioteket:

```
var i = 1;
images.each(function(image) { // images = array of images
  var img = new Element('img',
    {id: 'image-' + i, src: image.filename});
  var li = new Element('li');
  li.appendChild(img);
  Event.observe(li, 'click',
    editImage.bindAsEventListener(this, 'image-' + i));
  $('scroller-inner').appendChild(li);
```

Mærkeligheder

- ▶ Virkefelt/scope:
 - ▶ Synlig i **hele** funktionen selv om den erklæres i en blok
 - ▶ Erklær derfor **alle** variable i toppen af funktioner
- ▶ Flydende-kommatal følger ikke IEEE
- ▶ Firefox har en mindre striks parser (tillader komma efter sidste attribut)

Integration med Emacs

- ▶ GNU Emacs har ikke et Javascript mode
- ▶ Heldigvis findes der eksterne modes
 - ▶ **js2**: <http://code.google.com/p/js2-mode/>
 - ▶ Javascript-fortolker i elisp!
 - ▶ god til syntax highlighting
 - ▶ god til at spotte fejl
 - ▶ folde på blok-niveau ({})
 - ▶ **espresso**: <http://www.nongnu.org/espresso/>
 - ▶ understøtter en række biblioteker
 - ▶ integration med Firefox (MozRepl)
 - ▶ virker sammen med Speedbar og Imenu

Firefox og Firebug

- ▶ Firefox har god Javascript-understøttelse
- ▶ Firebug er en udvidelse til Firefox —
<http://getfirebug.com>
 - ▶ Inspektion af DOM
 - ▶ Ret i HTML og CSS og se ændringerne med det samme
 - ▶ Javascript debugger
 - ▶ Let adgang til AJAX-kald
- ▶ Udvidelser til Firebug —
<http://getfirebug.com/extensions/>
 - ▶ YSlow — Yahoos retningslinjer for websites
 - ▶ FireRainBow — syntax highlighting
 - ▶ Net Panel History Overlay — få styr på alle HTTP requests

Biblioteker til web-udvikling

- ▶ Web-applikationer bliver stadig mere komplekse
- ▶ Browsers opfører sig forskelligt ⇒ kedeligt at kode
- ▶ Open Source biblioteker er til stor hjælp
- ▶ Licensen er typisk Berkeley eller MIT
- ▶ Der er en del at vælge imellem
 - ▶ jQuery — stort community, modent, meget udbredt
 - ▶ prototype (og scriptaculous m.fl.) — modent med et løst community, udspringer fra Ruby on Rails miljøeta
 - ▶ Yahoo UI Library (YUI) — leverandør-styret
 - ▶ Google Web Toolkit (GWT) — Java ⇒ Javascript
- ▶ Alle udnytter Javascripts klasseløse objekter

Mere om prototype

- ▶ prototype giver et fundament til avancerede komponenter
 - ▶ website: <http://prototypejs.org>
 - ▶ DOM-manipulation
 - ▶ Håndtering af hændelser
 - ▶ Lettere AJAX-kald
 - ▶ Bedre arrays og strenge
- ▶ script.aculo.us er “uafhængigt” og giver en række effekter
- ▶ <http://scripteka.com> findes et (ufuldstændig) oversigt

Det magiske \$

- ▶ \$ bruges som præfiks til mange funktioner
- ▶ Den finder et DOM-element med et bestemt ID:
`$("#result")`
- ▶ Det er også muligt at ændre et element:
`$("#result").innerHTML = "42"`
- ▶ Der findes andre \$-funktioner, bl.a.:
 - ▶ `$$ (str)` — alle elementer med CSS selector `str`
 - ▶ `$F(id)` — værdien af formular-feltet `id`
 - ▶ `$w` — splitter en streng
 - ▶ `$H` — hash array

Leg med DOM

- ▶ Metoden `show` gør et element (eller undertræ) synligt
- ▶ mens `update` ændrer på indholdet
- ▶ Det er let at oprette nye HTML-elementer med `new Element`
- ▶ som sammen med metoden `appendChild` er en stærk kombination

Tid til eksempler: <http://localhost/zphoto/eksempler.html>

AJAX-kald

- ▶ AJAX-kald er *remote procedure calls* til web-serveren
- ▶ Fordelen er at siden kan opdateres uden et ny sidegenerering ⇒ giver en mere interaktiv oplevelse
- ▶ Web 2.0 applikationer benytter det hele tiden!
- ▶ Request-objektet er **meget** browser-afhængigt
 - ▶ XMLHttpRequest under Firefox og IE 7+
 - ▶ ActiveXObject for IE 5 og 6
- ▶ prototype gør det meget lettere med Ajax.Request-objektet
- ▶ Backend-koden skrives ofte i PHP, Perl eller Ruby
- ▶ Data sendes frem og tilbage i JSON (JavaScript Object Notation)

Tid til eksempler: <http://localhost/zphoto/>

Sjove effekter

- ▶ In place editor
- ▶ Beskæring af billeder
- ▶ Træk og slip
- ▶ Vælg datointerval
- ▶ Vis billede

Tid til eksempler: <http://localhost/zphoto/admin/> og
<http://localhost/zphoto/>

Javascript på desktoppen

- ▶ Javascript er ikke kun til web-applikationer
- ▶ Apple Dashboard, Google Widgets og Yahoo Widgets er til applets
- ▶ Adobe Air er til egentlig applikationer
- ▶ I open source har vi
 - ▶ Firefox bruger en del Javascript, f.eks. Jetpack og GreaseMonkey
 - ▶ WxJavascript/Gluescript — bygger på WxWidgets, mest til Windows
 - ▶ QtScript — en del af Qt
 - ▶ GNOME Seed og Gjs — begge til GNOME 3

GNOME Seed

- ▶ <http://live.gnome.org/Seed>
- ▶ Bygger på GObject Introspection ⇒ GNOME-plattformen
- ▶ WebKit's JavaScriptCore er Javascript-motoren
- ▶ Understøtter også SQLite, Readline og POSIX I/O

Let at installere under Ubuntu Linux 9.04:

```
deb http://ppa.launchpad.net/orange-owners/ppa/ubuntu \
    jaunty main
deb-src http://ppa.launchpad.net/orange-owners/ppa/ubuntu \
    jaunty main
```

Tid til eksempler

Gode bøger

- ▶ *Javascript: The Good Parts* af D. Crockford. O'Reilly & Associates, 2008.
- ▶ *Webmaster in a Nutshell* af S. Spainhour og R. Eckstein. O'Reilly & Associates, 2003
- ▶ *AJAX in Practice* af D. Crane, B. Bibeault og J. Sonneveld. Manning, 2007.
- ▶ *Learning jQuery* af J. Chaffer og K. Swedberg. Packt Publishing, 2007.
- ▶ *Prototype and Scriptaculous in Action* af D. Crane og B. Bibeault. Manning, 2007.