



JavaScript 1  
alert("Lektion 3");

Utbildare: Mahmud Al Hakim

NACKADEMIN

# Lektionstillfällets mål

## Mål med lektionen

- Att förstå JavaScript
- Att länka JavaScript till HTML
- Introduktion till objekt och metoder
- Att exekvera JavaScript via NodeJS
- Versionshantering med Git och Github

## Laboration 2

- Skapa ditt första repo på Github och öva på några Git-kommandon.

## Arbetsmetod

- Teori och praktik varvas under lektionen

# Kort summering av föregående lektion

## Föregående lektion:

Vi har gått igenom grunderna i CSS och jobbat lite i Chrome DevTools.

# Frontend-teknologier

**HTML**



**CSS**



**JS**



**HTML**

**Struktur**

**CSS**

**Utseende**

**JavaScript**

**Beteende**

# Vad är ett skript (*script*)?

Program som utför uppgifter i andra program.

Skript automatiserar arbetsuppgifter som annars skulle utföras steg för steg av användaren.

De körs inte direkt mot operativsystemet, som vanliga program, utan de förutsätter specifika program som de arbetar mot.

Skript tillför funktioner i program som webbläsare, på webbsidor...

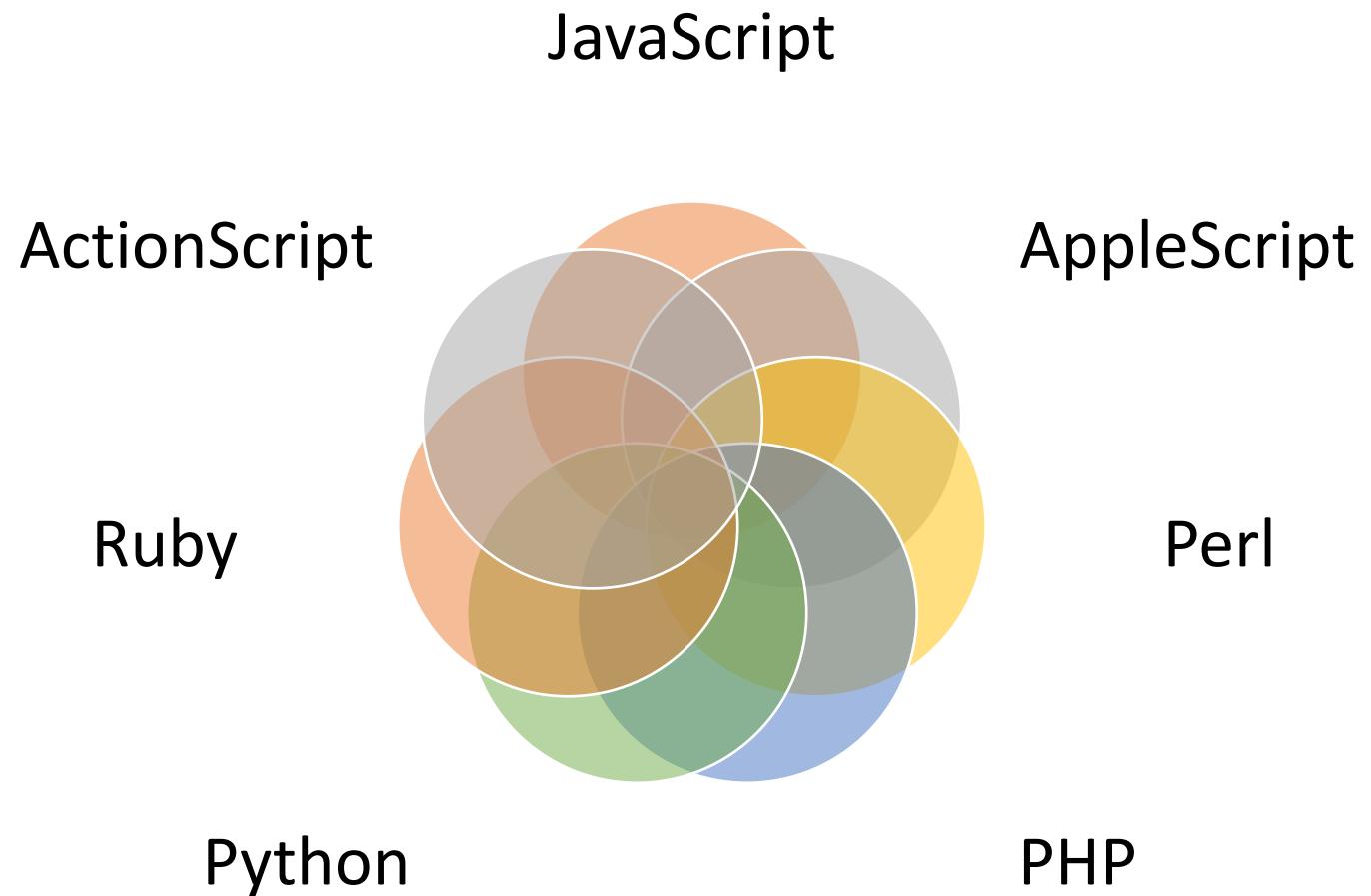
Källa: IDG IT-ord

<https://it-ord.idg.se/ord/skript/>

# Vad är skriptspråk?

- Skriptspråk är inom datavetenskap en benämning på "små" högnivåspråk inriktade på specialiserade uppgifter inom redan befintliga miljöer, i motsats till systemspråk, som används för programmering av tillämpningsprogram.
- Det finns ingen exakt avgränsning mellan systemspråk och skriptspråk, men skriptspråk är ofta interpreterande programspråk som använder dynamisk typning.
- Källa  
<https://sv.wikipedia.org/wiki/Skriptspr%C3%A5k>

# Exempel på skriptspråk



# Vad är JavaScript?

- JavaScript är ett programspråk som används oftast tillsammans med HTML och CSS för att lägga till beteende och interaktivitet på webben.
- Koden körs/översätts direkt i webbläsaren (interpreteras) och behöver inte kompileras som till exempel är fallet för andra programspråk såsom C# och Java.
- Tolkning av JavaScript-kod sker i besökarens webbläsare.

# What is JavaScript?

- Simon Allardice

<https://youtu.be/955L9-NoBoE>

## INTRODUCTION TO JAVASCRIPT

A dark gray square icon with the word "HTML" in white capital letters.

markup language  
**content**

A dark blue square icon with the word "CSS" in white capital letters.

style sheet language  
**presentation**

A dark maroon square icon with the words "JavaScript" in white capital letters.

programming language  
**behavior**



0:45 / 8:13



# Vad behövs för att skapa JavaScript?

- För att skapa JavaScript behöver du en textredigerare (texteditor)
- Om du använder Windows så finns det en standard texteditor som heter Anteckningar (NotePad). På Mac heter editorn Textredigerare.
- Bättre alternativ och dessutom finns till flera olika operativsystem:
  1. VS Code <https://code.visualstudio.com>
  2. Atom <https://atom.io>
  3. Sublime Text <https://sublimetext.com>
- Det finns även webbaserade alternativ t.ex.
  1. CodePen <https://codepen.io>
  2. JS Fiddle <https://jsfiddle.net>
  3. JS Bin <https://jsbin.com>

# Några fördelar med VS Code

1. Syntax Highlighting (Färgade koder)
2. Code Snippets
3. IntelliSense (Auto Complete)
4. Inbyggd Git-stöd
5. Många användbara tillägg

<https://code.visualstudio.com>

# Vad behövs för att köra JavaScript?

- För att köra JavaScript behövs en JavaScript tolk (interpreter)  
Detta kallas även JavaScriptmotor (engine).
- Alla moderna webbläsare har en inbyggd JavaScriptmotor.
  - Chrome JavaScriptmotor heter **V8**  
<https://developers.google.com/v8/>
  - FireFox JavaScriptmotor heter **SpiderMonkey**  
<https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey>
  - Edge JavaScriptmotor heter **Chakra**  
<https://github.com/Microsoft/ChakraCore>
- För att köra JavaScript utanför en webbläsare (t.ex. på en server)  
så behövs extra verktyg t.ex. **NodeJS** (som använder V8).

# Hur vet webbläsaren att finns JavaScript-kod?

- För att webbläsaren ska veta om att det är JavaScript-kod som gäller anger man detta med hjälp av HTML-taggen <script>.
- Mellan <script> och </script> placerar man sin JavaScript-kod.

<script>

Här skrivs JavaScript-kod!

</script>

# JavaScript i <head>

```
<html>
  <head>
    <title>JavaScript i head</title>

    <b><script></script></b>
      Här skriver du JavaScript-kod
    </b>

  </head>
  <body>
  </body>

</html>
```

# JavaScript i <body>

```
<html>
  <head>
    <title> JavaScript i body</title>
  </head>
  <body>

    <b><script></script></b>
    Här skriver du JavaScript-kod
    </b></script>

  </body>
</html>
```

# Ett första JavaScript

```
alert("Hello World!");
```

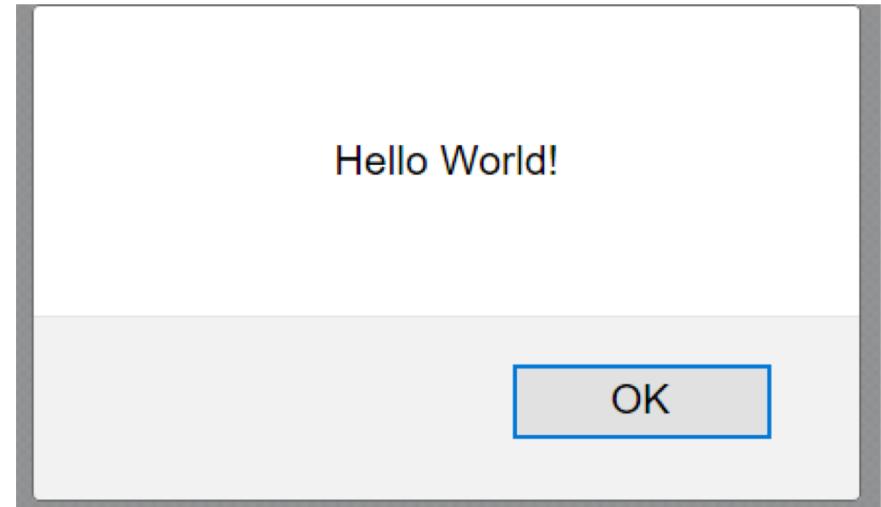
alert() är en funktion som visar ett meddelande på skärmen.

Besökaren måste klicka på "OK"-knappen på för att fortsätta.

Typiskt användningsområde är om du vill försäkra dig att information kommit fram till besökaren.

# Hello World!

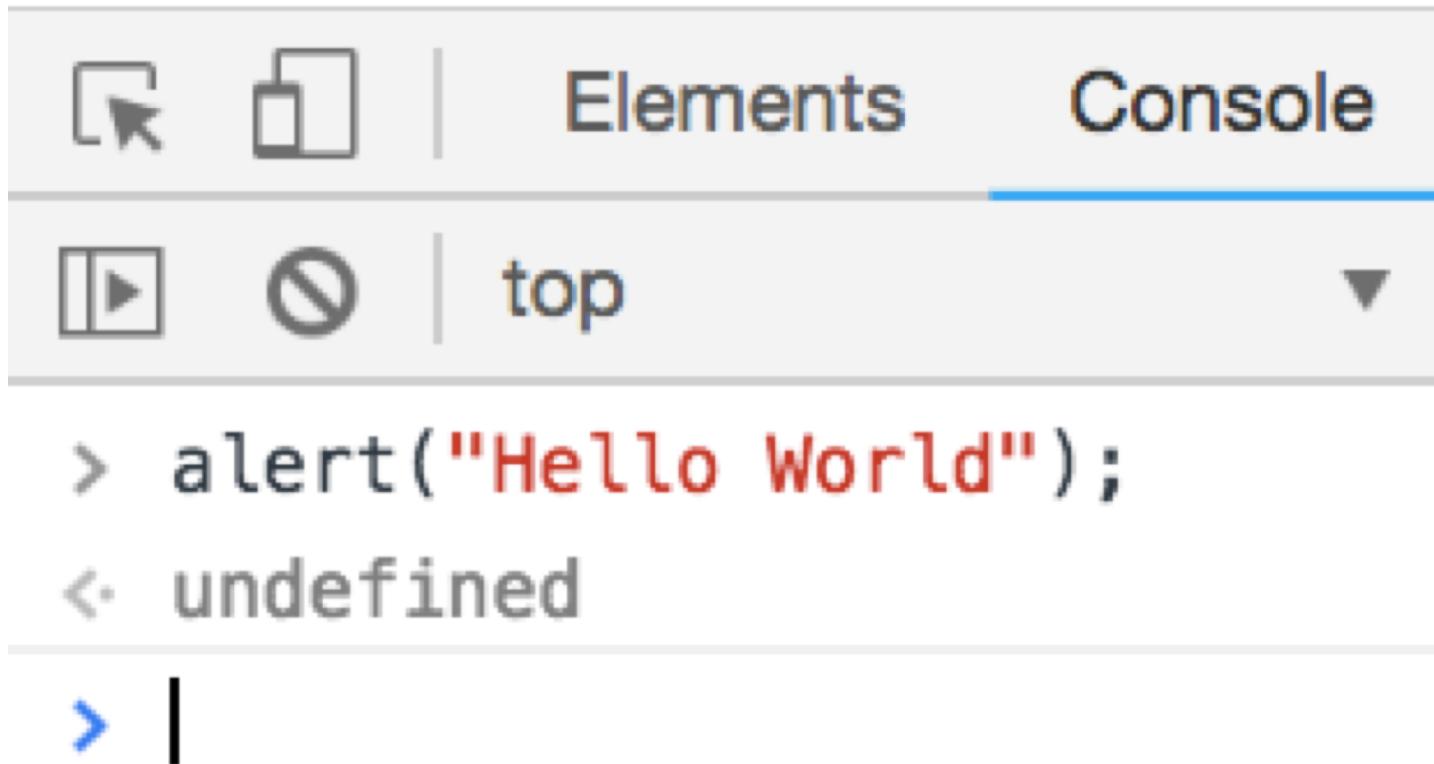
```
<html>
<head>
<title>Exempel 1</title>
</head>
<body>
<script>
    alert("Hello World!");
</script>
</body>
</html>
```



# Kommandotolk (terminal)

- Kommandotolken har som uppgift att ta emot och översätta instruktioner från program till en form som operativsystemet kan bearbeta.
- Bash är den mest spridda kommandotolken.
- Bash ingår som förvald kommandotolk i Linux/macOS och Windows 10.
- I webbläsaren kallas ett sådant verktyg konsol (Console).
- Konsolen meddelar oss vad som är rätt eller fel i JavaScript-kod och kan användas för att skriva ut saker.

Starta konsolen i Chrome DevTools  
Ctrl+Shift+J (Windows)  
Cmd+Opt+J (Mac)



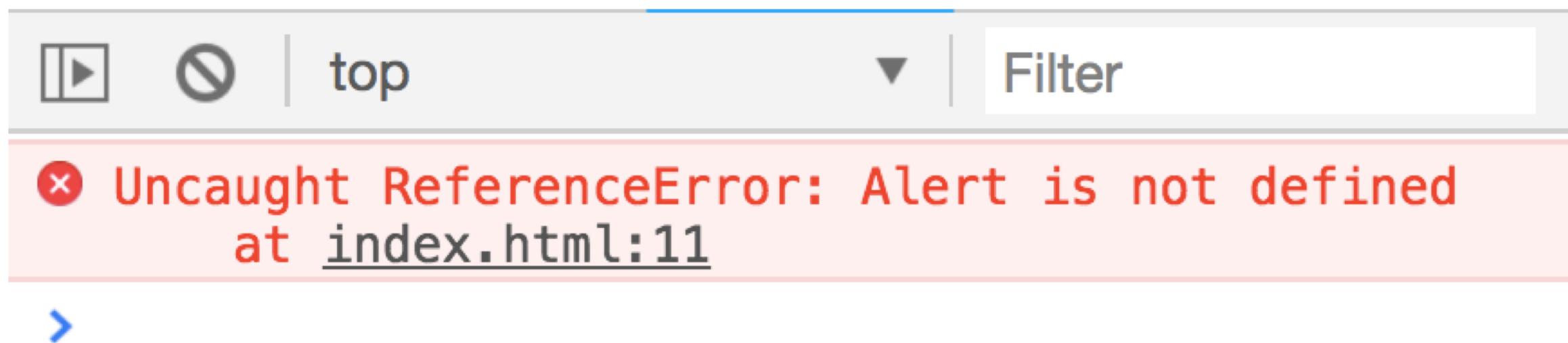
The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The top navigation bar includes icons for selection, inspection, and zoom, followed by 'Elements' and 'Console'. Below the tabs are controls for play/pause and stop, and a dropdown menu set to 'top'. The main console area displays the following interaction:

```
> alert("Hello World");
< undefined
> |
```

The command `alert("Hello World")` was run, resulting in the text "Hello World" being displayed in an alert box. The response `< undefined` is shown below the result. A cursor is visible at the end of the last line.

# OBS! JavaScript är Case Sensitive

- JavaScript är Case Sensitive (skiftlägeskänsligt)  
d.v.s. språket är känslig för stora och små bokstäver!
- Exempel:  
Alert och alert är två olika ord i JavaScript!



# Viktiga grundregler

1. JavaScript kod skall läggas in mellan <script> och </script>
2. JavaScript är *skiftlägeskänslig (Case Sensitive)*
3. Lägg alltid in en text mellan raka citationstecken  
(dubbla " " eller enkla ' ' )

# Externa JavaScript filer

- Externa JavaScript-filer kan användas av flera dokument.
- JavaScript-kod skrivas en separat fil med filnamnstillägget .js  
(t.ex. script.js)
- Script-taggen med attributet src infogar skriptet i HTML-dokumentet:

```
<script src="script.js"></script>
```

OBS!

Man får inte länka till externa JavaScript-filer och skriva kod inne i script-taggen samtidigt.

# Extern JavaScript-fil

exempel-02.html

```
<html>
<head>
<title>Exempel 2</title>
</head>
<body>
<script src="script.js"></script>
</body>
</html>
```

script.js

```
alert("Hello World!");
```

# Objektorienterad programmering (OOP)

- Objektorienterad programmering (Object Oriented Programming, OOP) är en programmeringsmetod i vilken ett program kan innehålla en varierande uppsättning objekt som interagerar med varandra.
- Den grundläggande idén med OOP är att vi använder objekt för att modellera saker som vi vill representera i våra program och ge ett enkelt sätt att komma åt metoder (funktioner).
- Här finns en lista på alla objekt som finns i JavaScript  
<https://www.w3schools.com/jsref/default.asp>

# What is object-oriented language?

- Simon Allardice

<https://youtu.be/SS-9y0H3Si8>



# Console

- Console är ett objekt som finns i JavaScript.
- Med hjälp av detta objekt kan vi bl. a. skriva ut information till webbläsarens konsol och kommandotolken/terminalen.
- En viktig metod i objektet är log()
- [https://www.w3schools.com/jsref/met\\_console\\_log.asp](https://www.w3schools.com/jsref/met_console_log.asp)
- Exempel

```
console.log("Hello world!");
```

# Exekvera JavaScript utanför webbläsaren med NodeJS

JS script.js X

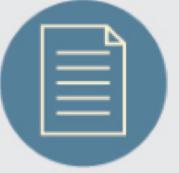
```
1 console.log("Hello World!");
```

En fil som  
skriver ut info  
till konsolen!

```
[Mahmuds-MacBook-Air:JavaScript-Exempel mahmudalhakim]$ node script.js  
Hello World!  
[Mahmuds-MacBook-Air:JavaScript-Exempel mahmudalhakim]$
```

node kan  
exekvera koden  
i terminalen

# Några användbara Bash-kommandon

 <b>pwd</b> PRINT WORKING DIRECTORY	 <b>echo</b> PRINT SOME ARGUMENTS
 <b>mkdir</b> MAKE DIRECTORY	 <b>cd</b> CHANGE DIRECTORY
 <b>ls</b> LIST DIRECTORY	 <b>cat</b> PRINT THE WHOLE FILE

Källa: <https://learntocodewith.me/command-line/unix-command-cheat-sheet/>

# Reviewing common Bash commands

<https://www.linkedin.com/learning/learning-bash-scripting/reviewing-common-bash-commands>

```
scott@orion:~$ pwd  
/home/scott  
scott@orion:~$ ls  
auth.log  fruit  my.sh  pets  trees  
scott@orion:~$ █
```



# Versionshantering

- Versionshantering innebär att tidigare versioner av dokument kan återskapas, och ändringar gjorda i dessa tidigare versioner kan spåras.
- Möjlighet till parallel utveckling, exempelvis rättning av äldre versioner parallellt med vidareutveckling av nya, är också väsentlig.
- Moderna utvecklingsverktyg har ofta inbyggt stöd för versionshantering t.ex. Atom och VS Code.



# Git och Github



- Git är ett versionshanteringssystem.

<https://git-scm.com/>

- Github är ett webbhotell för programvaruutvecklingsprojekt som använder versionshanteringssystemet Git.

- Microsoft köpte Github för 65 miljarder kronor (2018)

<https://github.com/>



# Skapa en repository på GitHub

## Create a new repository

A repository contains all the files for your project, including the revision history.

---

Owner



Repository name

/



# Konfigurera Git

```
git config --global user.name "Mahmud Al Hakim"
```

```
git config --global user.email "Mahmud.alhakim@gmail.com"
```

# Några viktiga kommandon

git init

git add .

git status

git commit -m "Meddelande"

git remote add origin (adress till en remote git)

git push -u origin master

(-u är lika med --set-upstream )

git log

# Git & GitHub Crash Course For Beginners

[https://youtu.be/SWYqp7iY\\_Tc](https://youtu.be/SWYqp7iY_Tc)



A Crash Course By Brad Traversy

TraversyMedia.com



0:15 / 32:41



# Laboration 2

1. Initiera (`init`) en ny lokal repo
2. Skapa en `README.md`
3. Lägg till filen till din lokal repo
4. Gör en `commit`
5. Ändra filen `README.md`
6. Lägg till filen igen och gör en ny `commit`
7. Testa kommandot `git status`
8. Skapa en ny repo på ditt Github-konto
9. Koppla din online repo (`remote`) till din lokal repo
10. Ladda upp projektet (`push`)

# Summering av dagens lektion

- Vi har idag gått igenom hur man länkar JavaScript till HTML.
- Vi har jobbat lite med objektet console och metoden log.
- Vi har exekverat JavaScript-kod via NodeJS.
- Ett första steg mot Git och Github (kämpigt men nyttigt :-)
- Reflektioner kring dagens lektion?
  - Vad tar du med dig från dagens lektion?
  - Finns det något som var extra svårt att förstå?
  - Finns det något som vi behöver repetera?
  - Hur upplevde du dagens arbetsmetoder?

# Framåtblick inför nästa lektion

- Läs kapitel 1 i bok 2 (JavaScript-boken).
- Under nästa lektion kommer vi att arbeta med satser, variabler och typer.