1. Javascript basics

Dagens program:

OBS: i dag arbejder vi med “ren” javascript - ikke med DOM

1. Om **programmeringssprog** og **javascript**
2. Erklæringer af **variabler** og **konstanter**
3. Statements til tekster og tal
4. **if**-statement
5. Statements til interaktion uden DOM:
   console.log, **alert**, **prompt**, **confirm**
6. Funktioner
7. Øvelser (mange!)

Programmeringssprog?

Sprog man bruger til at få computeren til at udføre opgaver

Indeholder anvisninger/kommandoer til computeren

Mest anvendte programmeringssprog

Wikipedia, definition af programmeringsprog: https://simple.wikipedia.org/wiki/Programming_language

Programmeringssprog og andre computersprog

html er **ikke** et programmeringssprog

det er et **opmærkningssprog**

hvilke elementer har vi på en webside (**DOM'**en)?

css er **ikke** et programmeringssprog

det er et **layoutsprog**

hvordan skal elementerne på en webside præsenteres?

javascript er et programmeringssprog!

besked til computeren i et program: hvad skal der ske?

Wikipedia: forskellige typer computersprog:https://en.wikipedia.org/wiki/Computer_language

Programmeringssprog

Et programmeringssprog er et kunstigt sprog

Simple i forhold til naturlige/menneskesprog - og formelle

Et computersprog er beskrevet gennem regler

Syntax-regler: hvordan skal sætningerne sættes sammen og ordene staves?

Semantiske regler: Hvad betyder sætningerne?

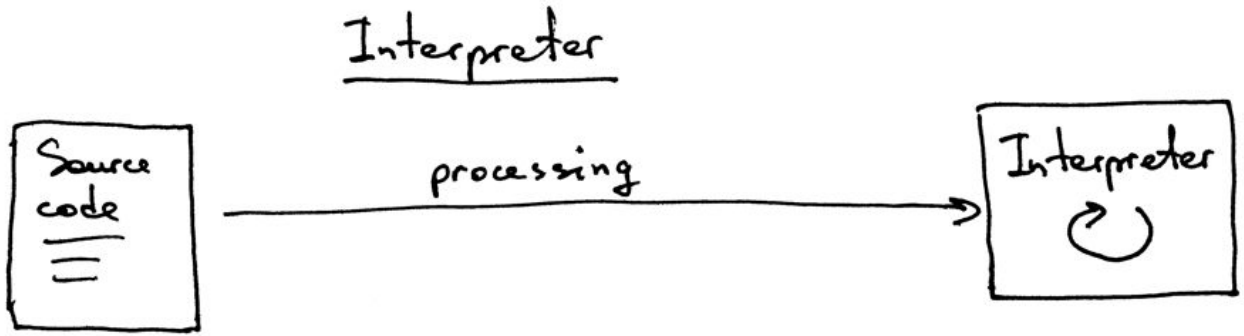
Regler for syntax og semantik beskrives i referencemanualer

Spørgsmål på Stackoverflow: What is the difference between syntax and semantics of programming languages:
https://stackoverflow.com/questions/17930267/what-is-the-difference-between-syntax-and-semantics-of-programming-languages

Fortolkede sprog

- Programmet gives til en **fortolker/interpreter**.
- Fortolkeren oversætter sætningerne til maskinsprog
- Computeren udfører sætningerne efterhånden.

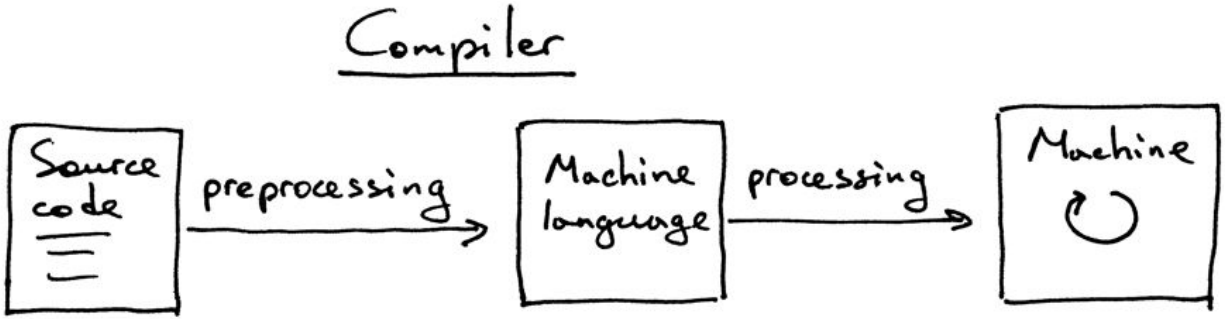
  Eksempler: javascript, php, python, ruby, perl



Oversatte sprog

- Programmes skrives og gives til en **oversætter/compiler**.
- Compileren oversætter det hele til maskinsprog
- Computeren udfører det oversatte program.

Eksempler: Java, C#, Swift



Script-sprog

Scriptsprog bruges tit blot som synonym for fortolkede sprog

Men kan også betyde et programmeringssporg til småprogrammer (scripts), som fungerer sammen med et andet computersprog

F.eks: javascript, som fungerer sammen med html og css i en browser

Om javascript

Programmeringssprog

Fortolket sprog

Scriptsprog

objektorienteret sprog

Alle **browsere har indbygget en javascript-fortolker/engine**

javascript kan **manipulere** dokumentets html-elementer (**DOM**)

javascript kan også **manipulere** html-elementernes layout (CSS'en)

En standardiserings-organisation, **ECMA** tager sig af beskrive javascript

Sidste version: **ECMA-script6** (https://www.w3schools.com/js/js_es6.asp)

Wikipedia om Javascript: https://en.wikipedia.org/wiki/JavaScript

javascript - et objektorienteret sprog

Javascript har mange **indbyggede objekter**

Man kan også **skabe objekter** i javascript

Et eksempel på et indbygget objekt i javascript er

Math

Math er et indbygget objekt, som har en lang række **egenskaber** og **metoder**.

F.eks:

Math.PI (egenskab)

Math.random() (metode)

. (punktum) mellem objektnavn og egenskab/metode kaldes **dot-notation** (ejefald)

Opsamling: Begreber - hvad var det nu, det betød???

DOM

Computersprog

Opmærkningssprog

Layoutsprog

Programmeringssprog

Compilet sprog

Fortolket sprog

Script-sprog

Javascriptfortolker (hvor?)

objektorienteret sprog

indbyggede objekter

dot-notation

Erklæringer af variabler i javascript

Variabel og erklæringer

En **variabel** er et navn, som man kan tildele en værdi

En variabels værdi kan ændres i programmet

Variabler **erklæres** én gang - helst øverst i scriptet

Eksempler:

```
let forNavn = "Helle"; // tekst
let alder = 62; // tal
let enlig = false; // boolean
let sulten; // ingen værditildeling endnu
```

Good practice:
camel-case til variabelnavne

forNavn, alder og enlig har fået **tildelt værdier** af tre forskellige **datatyper**: tekst, tal og boolean.

Konstanter

Tit har man brug for navne på værdier, som **ikke** skal ændres.

Her bruger man **konstanter** i stedet for variabler

Erklæring af en konstant:

```
const MOMS = 0.25; // momsen er konstant - skal ikke ændres i programmet
```

Good practice:
vasaler til konstanter

Opsamling: Ord og begreber - hvad betød de?

- **let**
- **erklæring af variabel**
- **camelcase**
- **const**
- **erklæring af konstant**
- **tildeling af værdi**
- **datatyper**
  - **tekst/string**
  - **tal**
  - **boolean**

Operatorer og window-metoder

Window-objektet

window-objektet: et indbygget objekt i javascript
browserens vinduet - med eller uden html-elementer (DOM)
window har mange **egenskaber/properties** og **metoder**
I dag skal vi bruge:

- **window.console.log()** - skriver noget i browserens consol
- **window.alert()** - skriver noget i en popup
- **window.prompt()** - spørger om noget i en popup
- **window.confirm()** - beder om ja eller nej til et spørgsmål (i en popup)

https://www.w3schools.com/jsref/obj_window.asp

window.console.log()

Indbygget metode, som kan udskrive en værdi i browserens console-vindue

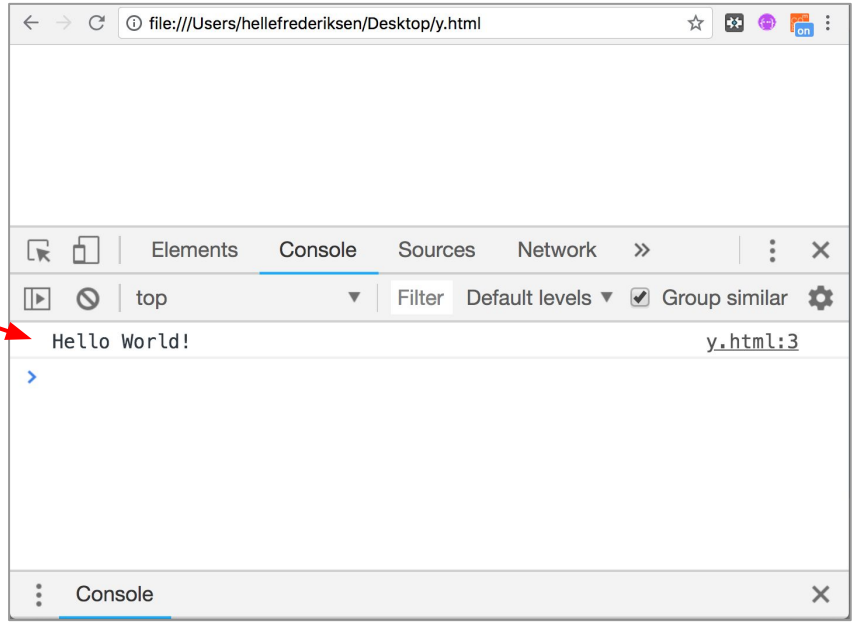
Eksempel:

```
let greeting = "Hello World";
console.log(greeting);
```

I Chrome(og de fleste andre browsere):

Højreklik i browservinduet og vælg inspect / undersøg

Vælg fanebladet Console



Sammenlægning af tekster

Efter erklæringen, kan variablerne ændres i **statements**

```
let minTekst = "Her er en tekst";
minTekst = minTekst + " , som fortsætter her!";
```

+ er en operator til sammenlægning af tekster, **text concatenation**

linje 2 kan også skrives som:

```
minTekst += " , som fortsætter her!";
```

+= er også en operator til **concatenation**.
Tager den værdi, variablen har i forvejen, og tilføjer højresiden

Template litterals

I stedet for at tekst-concatenation, kan man bruge **Template litterals**

```
let minTekst = "her er en tekst";
minTekst = `${minTekst}, som fortsætter her`;
```

Udenom hele teksten bruge **`...`** (accent grave)

I teksten, kan man indsætte variabler eller udtryk - **$** foran og **{**...**}** omkring variabler eller udtryk.

Template literals (Template strings), MDN web docs:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals

Indbyggede metoder til tekster

Tekst-objekter har en række metoder, fx:

```
let tekst = "Eksempel på en tekst";


let len = tekst.length; // tekstens længde (20)
tekst = tekst.toUpperCase(); //tekst er nu “EKSEMPEL PÅ TEKST”
tekst = tekst.toLowerCase(); //tekst er nu “eksempel på tekst”
```

JavaScript Stings, W3schools: https://www.w3schools.com/jsref/jsref_obj_string.asp

Statements til beregning af tal

I statements kan man udregne tal.

Eksempel 1

```
let pris = 100;
const moms = 0.25;
pris = pris + pris*moms;
```

Eksempel 2

$$\text{BMI} = \frac{\text{Vægt (kg)}}{\text{Højde(m)} \times \text{Højde(m)}}$$

```
const height = 1.66;
const weight = 85;
const bmi = weight/(height * height);
```

Tal-operatorer

+ Addition
eks: **pris = indkøbsPris + MOMS;**

- Subtraction
eks: **indkøbsPris = pris - MOMS;**

*** Multiplication**
eks: **totalPris = antal * pris;**

/ Division
eks: **pris = totalPris/antal;**

++ Increment
eks: **antal++;**

-- Decrement
eks: **antal--;**

Indbyggede metoder til beregninger

js har et indbygget objekt, **Math**, som har en række nyttige metoder til tal:

```
let tal = 3.5;

tal = Math.round(tal); // tal er nu afrundet til (4)

tal = Math.pow(tal,2); // tal opløftes til 2. potens (16)

tal = Math.random(); // tal er et tilfældigt tal mellem 0 og 1 (fx. 0.843219827740112)

tal = Math.round(Math.random()*10); // tal er nu et heltal mellem 0 og 10

erTal = isNaN(tal); // er sand, hvis tal ikke er et tal - her er den false
```

Math, W3schools: https://www.w3schools.com/js/js_math.asp

Vi går igang igen kl. 10.30

video: https://youtu.be/fjVgoJsptd8

Øvelse 0 - opret en githubmappe til dagens øvelser

tema5
  undervisningsopgaver
    01-js-basics
      00-skelet.html

1. Opret på github et repository, **tema5.**
2. Lav en mappe, **tema5**, på din computer (måske under en 2-semester-mappe, som ligger i en KEA-mappe?).
3. Åbn **tema5**-mappen i Brackets, og klon til det nye repository.
4. Lav i tema5 en undermappe, **undervisningsopgaver**.
5. Lav i **undervisningsopgaver** en mappe, **01-js-basic** til dagens øvelser.
6. Åbn **01-js-basic** i Brackets
7. Læg et genbrugeligt , tomt html-skelet, **00-skelet.html** ind i mappen
8. Udfyld dette regneark med dit navn og din github-konto

Vi går igang igen kl. 10.30

video: https://youtu.be/bhG3QfliIXs

Øvelse 01 - udregn areal

tema5
  undervisningsopgaver
    01-js-basics:
      00-skelet.html
      01-areal.html

Åbn skelet.html, og gem den som en ny til, **01-areal.html** i **01-js-basics**

Skriv i script-tagget et program, som kan udregner areal ud fra en længde og en bredde.

Længde, bredde og areal skal erklæres som variabler i programmet.

Resultatet skal vises i console-vinduet, og have denne form:

Længden er 3 meter og bredden er 5 meter. Arealet er 15 kvadratmeter

Test programmet med forskellige værdier for længde og bredde

Commit og push til gitHub, når du er tilfreds med opgaven

Opsamling - hvad var det nu, det betød?

window-objektet

window.console.log()

sting

concatenation

Template litterals

Math.round(tal)

Math.pow(n, m)

Math.random(tal)

tekst.length

tekst.toUpperCase()

tekst.toLowerCase()

isNaN(tal)

Betingelse / condition
if-statement

if-statement

Hvis et eller flere statements KUN skal udføres, hvis en betingelse er opfyldt:

```
if(betingelse/condition){
        statement1;
        statement2;
}
```

- hvor betingelse er en boolsk værdi (sand eller falsk) eller et boolsk udtryk

eksempel:

```
if(alder < 18) {
        console.log("Barn");
};
```

< er en **logisk operator**

alder < 18 er en **betingelse/condition**

if else else if: w3school: https://www.w3schools.com/js/js_if_else.asp

Logiske operatorer

==	Er lig med
!=	Er forskellig fra
>	Er større end
<	Er mindre end
>=	Er større end eller lig med
&&	Og (mellem to betingelser)
||	Eller (mellem to betingelser) alt+i på mac-keyboard
!	Ikke (foran en betingelse)

if-statement - flere eksempler

```
if(alder >=18) {
        console.log("Voksen");
}

if(køn == "k"){
        console.log("Kvinde");
}

if(køn !="k"){
        console.log("Mand");
}

if(køn !="k" && alder<18){
        console.log("Dreng");
}

if(alder<18 || alder>65){
        console.log("Ikke erhvervsaktiv alder");
}
```

if-else-statement

Et eller flere statements skal KUN udføres, hvis en betingelse er opfyldt - og ELLERS skal nogle andre udføres:

```
if( betingelse ){
    statement1;
}else {
    statement2;
}
```

Eksempel:

```
if( alder < 18 ){
    console.log("Barn");
} else {
    console.log("Voksen");
}
```

forgrenede if-statements

Man kan konstruere en if-sætning med mange else-grene:

```
if(betingelse 1){
        statement1;
} else if(betingelse 2){
        statement2;
} else if(betingelse 3){
        statement3;
} else{
        statement4;
}
```

Eksempel på forgrenede if-statements

Eksempel:

```
if(alder < 6) {
        console.log("før skolealder");
} else if(alder < 15) {
        console.log("skolealder");
} else if(alder < 65) {
        console.log("erhvervsaktiv alder");
} else {
        console.log("pensionsalder");
}
```

kl. 11.00 går vi igang igen

video: https://youtu.be/qn0LueadD_I

Øvelse 02 - er arealet mellem 100 og 200??

tema5
undervisningsopgaver
js-basics:
01-areal.html
02-areaTest.html

Gem 01-areal.html i en ny kopi, **02-arealTest.html**

I programmet skal du erklære to variabler: **længde** og **bredde**.

Programmet skal udregne arealet og fortælle om arealet er for lille, ok eller for stort. Resultatet skal vises i console.

Hvis arealet er under 100, skal der stå: **Arealet er for lille**

Er arealet mellem 100 og 200, skal der stå: **Arealet er ok**

Er det større end (eller lig med) 200, skal der stå: **Arealet er for stort**

Test programmet, så du ser alle tre muligheder for respons i funktion.

Commit og push til gitHub, når du er tilfreds med opgaven

Opsamling - hvad var det nu, det betød????

Begreber og ord

if

betingelse/condition

logisk operator

else

else if

Logiske operatorer

==

!

!=

<

>

<=

>=

&&

||

Interaktivitet

Interaktivitet

I plejer at lave interaktive websider ved at skrive i dokumentet (DOM'en).

Vi gemmer alt om dokumentet til i morgen.

window-objektet har nogle metoder til interaktion:

alert, **prompt** og **confirm**.

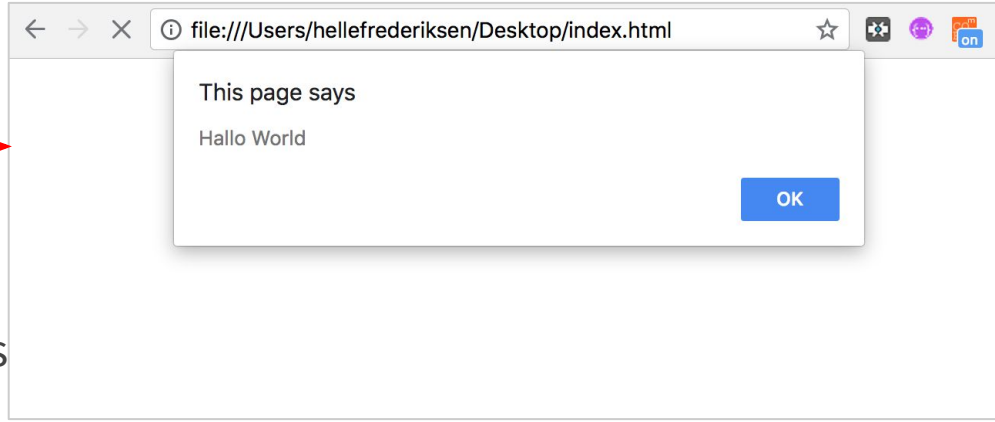
- De anvendes ikke ret meget på websider!

Javascript popup boxes, W3school: https://www.w3schools.com/js/js_popup.asp

window.alert()

Js-funktion, som kan vise en alert-boks med en tekst i browservinduet.

```
let tekst= "Hallo World";
alert(tekst);
```



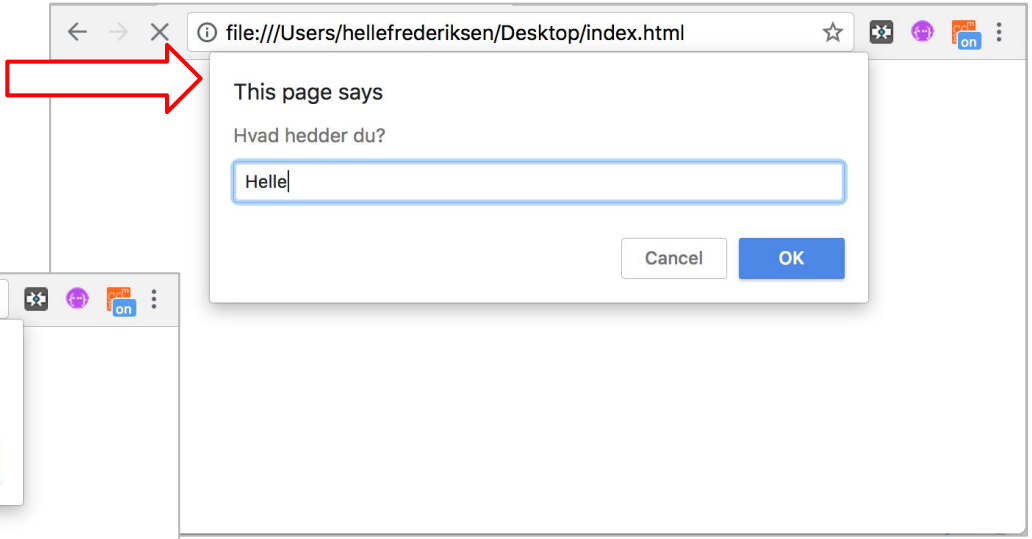
Kan bruges dér, hvor vi ellers har brugt console.log() - men alertboxen vises i en boks på brugerens vindue.

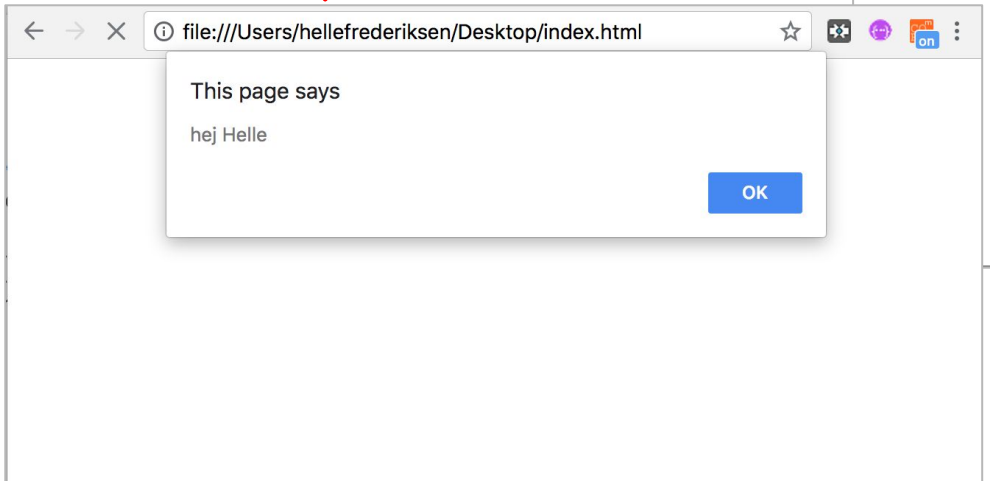
Scriptet stopper indtil brugeren klikker **OK**.

window.prompt()

Metode, som opfordrer brugeren til at angive en værdi, som kan gemme i en variabel

```
let navn=prompt("Hvad hedder du?");
alert(`hej ${navn}`);
```

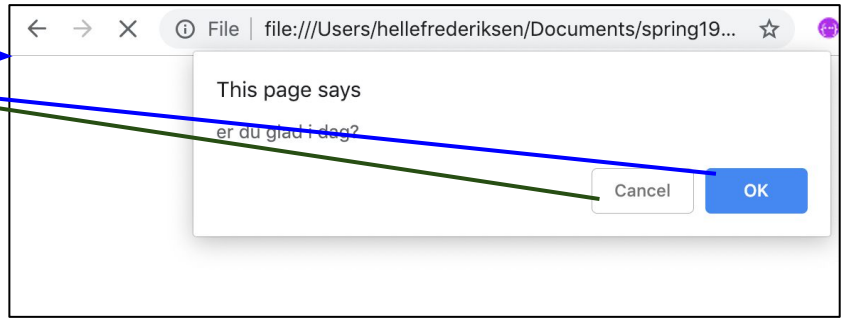




window.confirm()

Metode, som opfordrer brugeren til at svare ja (OK) eller nej (Cancel) til et spørgsmål.

```
if(confirm("er du glad i dag?")){
        alert(":)");
}else{
        alert(":(");
}
```

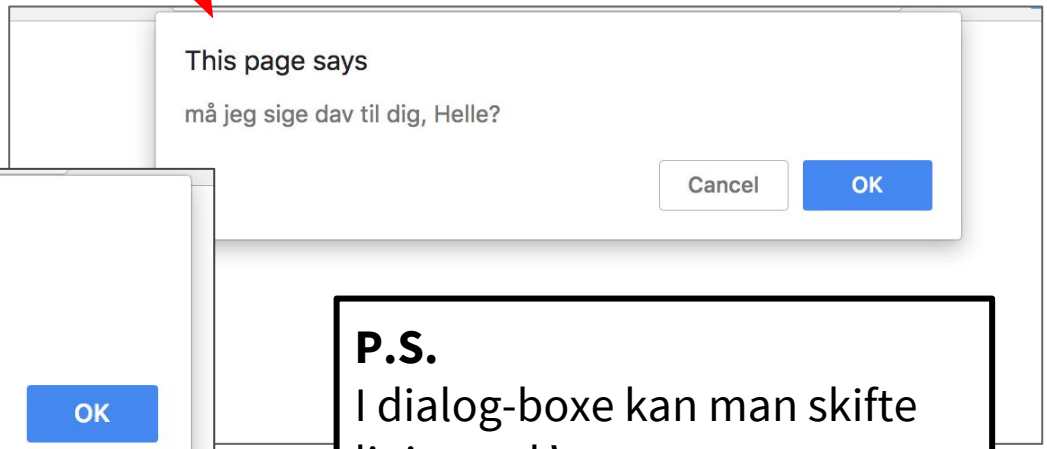


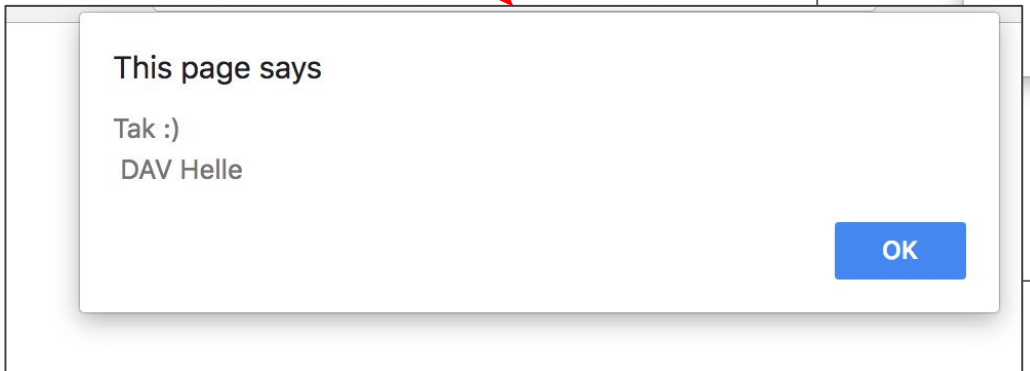
Dialog

Eksempel

```
let navn=prompt("Hvad hedder du?");
let sigGoddag=confirm(`må jeg sige dav til dig, ${navn}?`);

if(sigGoddag){
        alert(`Tak :)\n DAV ${ navn}`);
}
```





P.S.
I dialog-boxe kan man skifte linje med **\n**

Lav øvelse 3 og 4 og hold frokost!
Vi starter igen kl. 12:30

Øvelse 3 Sig goddag

Lav en ny html-fil, **03-goddag.html**, og gem den i 01-js-basics

Skriv et program, som spørger brugeren om hans/hendes navn (prompt), og som herefter svarer med “Goddag Helle” (alert), hvor navnet selvfølgelig er det angivne.

Commit og push til gitHub, når du er tilfreds med opgaven

tema5
  undervisningsopgaver
    01-js-basics:
      01-areal.html
      02-areaTest.html
      03-goddag.html

Lav øvelse 3 og 4 og hold frokost!
Vi starter igen kl. 12:30

video

Øvelse 4 udregn areal i interaktiv dialog med bruger

html-filen til denne øvelse, skal hedde **04-arealInteraktiv.html**

Lav en ny udgave af areal-programmet.

Programmet skal bede brugeren angive en længde og en bredde

Programmet skal udregne arealet, og give brugeren resultatet efter formen:

Længde: 5 meter, bredde: 7 meter, areal: 35 meter

Til sidst skal programmet spørge brugeren om der skal startes forfra.

Hint: få programmet til at starte forfra med **location.reload();**

tema5
undervisningsopgaver
js-basics:
01-areal.html
02-areaTest.html
03-goddag.html
04-arealInteraktiv.html

Opsamling

alert()
prompt()
confirm()

Vi starter igen kl. 12:30

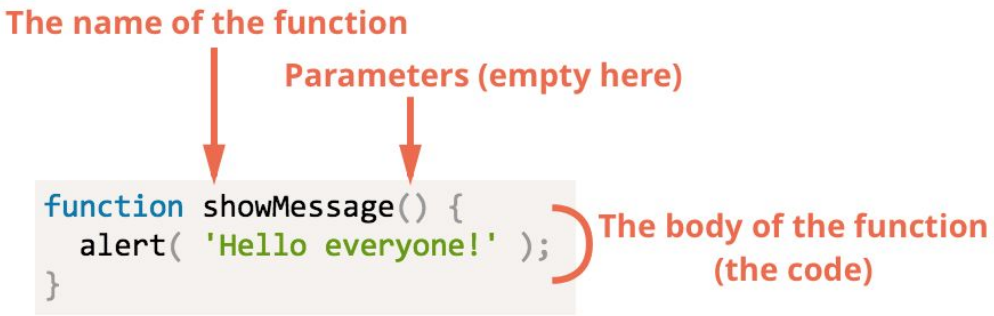
Funktioner

Funktioner

Funktioner

Funktioner

Et stykke isoleret kode.
En funktion skal først **erklæres**



Når en funktion er erklæret, sker der ingenting.
Først når den **kaldes**, udføres funktionen: **showMessage();**

Funktioner - hvorfor?

- Man kan undgå at gentage kode
- Programmer bliver modulopbygget og mere overskueligt
- Funktioner kan også tit genbruges i andre programmer

Derfor forsøger man altid at oprette funktioner, hvor man kan.

Funktion med parameteroverførsel

```
function visBesked(txt){
      alert(txt);
}
```

Erklæring af funktionen
Funktionen er parat til at alerte en tekst, som den får **overført som parameter**

```
visBesked("Goddag");
```

Funktionen kaldes med værdien “Goddag” som parameter. Den skriver “Goddag” i popup’en

```
let besked = "Farvel";
visBesked(besked);
```

Funktionen kaldes med variablen **besked** som parameter. Den skriver “Farvel” i en popuppen, fordi besked har værdien “Farvel”, som **parameteroverføres**

Funktioner, som returnerer en værdi

```
function visBesked(message){
      let first = "Info: ";
      return `${first} ${message}`;
}
 let besked="Kamilla vejleder i eftermiddag";
 alert(visBesked(besked));
 let info="Helle underviser i dag";
 alert(visBesked(info));
 alert(visBesked("Louise underviser først i næste
 uge"));
```

Funktionen sætter teksten “info:” foran teksten, som parameteroverføres

Popup’en siger:

“Info: Kamilla vejleder i eftermiddag”

Opsamling

funktion
parameteroverførsel
funktion, som returnerer en værdi

video

Øvelse 5 Arealer og funktioner

tema5
undervisningsopgaver
js-basics:
01-areal.html
02-areaTest.html
03-goddag.html
04-arealInteraktiv.html
05-arealFunktion

Filens navn: **05-arealFunktion.html**

Lav en funktion, **areal**, som med længde og bredde som parametre, kan udregne et areal, og returnere resultatet**.**

Programmet skal:

- Bede brugeren om længde og bredde (prompt)
- Videregive resultatet til brugeren og spørge, om brugeren starte forfra (confirm)
- Starte forfra, hvis brugeren vil det

Commit og push til gitHub, når du er tilfreds med opgaven

video

Øvelse 6 Arealudregning med fejlmeddelelse

Filens navn: **06-arealFejl.html**

I det forgående program, arealFunktioner.html:

1. Test, hvad der sker, hvis brugeren angiver tekst i stedet for tal som længde eller bredde
2. Test også, hvad der sker, hvis brugeren ikke angiver en værdi for længde eller bredde
3. Prøv at ændre på programmet, så det tager højde for fejldata og giver brugeren en fejlmelding
4. Commit og push til gitHub, når du er tilfreds med opgaven

tema5
undervisningsopgaver
js-basics
01-areal.html
02-areaTest.html
03-goddag.html
04-arealInteraktiv.html
05-arealFunktion.html
06-arealFejl.html

Eftermiddagsopgaver

Øvelse 7 Sig godmorgen

html-filen til denne øvelse, skal hedde **godmorgen.html**

Skriv et program, som i en alert-boks siger:

Godmorgen mellem kl. 5 og kl 10,

Goddag mellem kl 10 og 18,

Godaften mellem 18 og 24 og

Godnat mellem 24 og 5.

obs: denne javascript-funktion fortæller, hvilken time, man befinder sig i:

```
new Date().getHours()
```

Commit og push til gitHub, når du er tilfreds med opgaven

Øvelse 8 - udregn bmi

html-filen til denne øvelse, skal hedde **bmi.html**

Du skal lave et program, som ud fra en persons højde og vægt udregner bmi'en

$$BMI = \frac{\text{Vægt (kg)}}{\text{Højde(m)} \times \text{Højde(m)}}$$

Du skal ikke i dialog med brugeren i denne opgave - læg højde og vægt ind som variabler.

Resultatet skal vises i console-vinduet efter denne form:

Bmi'en er 25

Commit og push til gitHub, når du er tilfreds med opgaven

Øvelse 9 - udregn bmi og bestem fedmeklasse

html-filen til denne øvelse, skal hedde **bmiKlasse-9.html**

Programmet skal udregne bmi på

baggrund af højde og vægt

(tag udgangspunkt i en kopi af øvelse 2).

Ud fra dette skema, skal du i console.log udskrive både bmi og fedmeklasse.
Fx: **Bmi'en er 27, fedmeklasse: overvægtig**

Commit og push til gitHub, når du er tilfreds med opgaven

Fedmeklasse	**bmi**
Undervægtig	mindre end 19
Normal	Mellem 20 og 25
Overvægtig	Mellem 25 og 30
Fed	Mellem 30 og 35
Svær fedme	Mellem 35 og 40
Ekstremt fed	over 40

Øvelse 10 Udregn bmi i interaktiv dialog med bruger

html-filen til denne øvelse, skal hedde **bmiInteraktiv.html**

Lav en ny udgave af bmi-programmet.

Programmet skal bede brugeren angive sin højde og vægt.

For danske brugere vil det være bedst at opgive højden i cm.

Programmet skal så udregne bmi'en, og fortælle brugeren resultatet

Til sidst skal programmet spørge om brugeren vil have besked om sin vægtklasse, og kun hvis brugeren siger ja, skal vægtklassen oplyses

Hvis brugeren ikke opgiver tal for højde og vægt, skal programmet give en fejlmelding.

Commit og push til gitHub, når du er tilfreds med opgaven

Øvelse 11 - quiz-program

html-filen til denne øvelse, skal hedde **quiz.html**

Lav et program, som kan stille 5 spørgsmål til brugeren, et ad gangen, og give et point for hvert rigtige svar.

Når alle spørgsmål er besvaret, skal programmet fortælle brugeren, hvor mange svar, der var rigtige.

Undervejs skal programmet fortælle om hver enkelt svar var rigtigt.

Bestem selv, hvad quizzen skal handle om.

Din løsning er bedst, hvis den indeholder en funktion, som kan tage sig af at sammenligne det rigtige svar med et svar fra brugeren.

Commit og push til gitHub, når du er tilfreds med opgaven

Øvelse 12 - Gæt et tal

html-filen til denne øvelse, skal hedde **guess.html**

Programmet skal finde et tilfældigt tal mellem 0 og 20, og bede brugeren om at gætte tallet det.

Når brugeren har gættet, fortæller programmet om tallet var rigtig, eller om det var for højt eller for lavt.

Så får brugeren lov at gætte igen, og sådan fortsætter programmet til brugeren har fundet det rigtige tal.

Når brugeren har gættet tallet, fortæller programmet, hvor mange gæt, der blev brugt, og spørger om brugeren vil prøve igen med et nyt tal.

Commit og push til gitHub, når du er tilfreds med opgaven

Øvelse 13 - Date-rådgivning

html-filen til denne øvelse, skal hedde **date.html**

Når man skal date, bør man finde en, som hverken er for ung eller for gammel.

Der findes en regel, som hedder “half your age plus seven”.

Lav et program, som kan tage imod din egen alder og din dates alder, og fortælle dig, om reglen er overholdt - både til den ene og den anden side.

Sørg endelig for, at programmet ikke kan benyttes af mindreårige eller pædofile - begge skal være over 15!

Commit og push til gitHub, når du er tilfreds med opgaven