

# Variabler og konstanter

## Lage (deklarer) en variabel

```
Kodeordet var          Variabelnavn  
                      (skal alltid ha liten forbokstav)  
                      ↓  
var antallBiler = 234;           "Settes lik"  
                      ↓  
                      Verdi
```

## Angi verdi på en variabel som allerede er lagd

```
antallBiler = 10;
```

## Angi ny verdi ut i fra gammel verdi

```
Ny verdi             Gammel verdi  
↓                  ↓  
antallBiler = antallBiler + 1; // Øker med én  
  
// Kortformer:  
antallBiler++;      // Øker med én  
antallBiler--;      // Minsker med én  
antallBiler += 3;    // Øker med 3  
antallBiler -= 5;    // Minsker med 5  
antallBiler *= 2;    // Ganges med 2  
antallBiler /= 6;    // Deles på 6
```

## Modulo-operatoren

```
var epler = 47;  
var personer = 3;  
var eplerPrPers = Math.floor(epler / personer);  
var tilOvers = epler % personer ;
```

## Datatyper med primitive verdier

```
var tall1 = -3;           // Heltall  
var tall2 = 5.7;          // Desimaltall  
var tekst = "Hei";        // Tekst  
var test = false;         // true eller false
```

## Konvertere tekst til tall

```
var tekst = "2344.5";  
var tall = parseFloat(tekst);
```

## Konvertere tall til tekst

```
var tall = 2344.5;  
var tekst = tall.toString();
```

## Formatere et tall (bestemme antall desimaler)

```
var tall = 29.3638;           Antall desimaler  
var formatertTall = tall.toFixed(2);  
alert(formatertTall); // 29.36
```

## Lage en konstant

```
Kodeordet const  
const DAGER_I_UKA = 7;
```

## Tekst

### Sette sammen tekst

```
Konkateneringstege (sette sammen tekst)  
var tid = "Jul";  
var melding = "Ikke lenge igjen til " + tid;  
  
alert(melding); // Ikke lenge igjen til Jul
```

### Bruke escape-tegn for linjeskift og tabulator

```
// Angi linjeskift i filer:  
var nyLinje = "\n";  
var tekst1 = "Navn:" + nyLinje + "Roald Grønningen";  
  
// Angi linjeskift i nettsiden:  
var nyLinje = "<br />";  
var tekst2 = "Navn:" + nyLinje + "Roald Grønningen";  
  
// Angi anførselstegn:  
var anf = "\"";  
var tekst3 = "Han er helt " + anf + "syk" + anf;
```

## Trekke ut en tekst fra en annen tekst

```
Indeks 0           Start-indeks   Antall tegn  
↓                 ↓             ↓  
var teksten = "Dramatisk!";  
var uttrekk = teksten.substr(3, 5);  
alert(uttrekk); //matis
```

## Søke etter en tekst i en annen tekst

```
Søkeord      Start-Indeks  
↓             ↓  
var tekst = "Her var det mye trafikk. Tør jeg går.";  
var indeks = tekst.indexOf("trafikk", 0);  
alert(indeks); // 16 Hvis ikke funnet blir indeks -1
```

## Skjemaelementer

### Lage tekstboks

```
Verdi som vises som standard i tekstboksen  
↓  
<input type="text" id="txtNavn" value="startverdi">/input>
```

### Hente verdi fra tekstboks

```
var navn = document.getElementById("txtNavn").value
```

### Lage knapp

```
<button id="btnKnapp" type="button">Klikk meg...</button>
```

### Lage avkrysningsbokser

```
<input type="checkbox" name="chkPizza" />Jeg like pizza<br />  
<input type="checkbox" name="chkPasta" checked="checked" />Jeg  
liker pasta
```

### Sjekke om avkrysningsboks er markert

```
if (document.getElementById("chkPizza").checked === true) {  
    // Kode  
}
```

## Lage nedtrekksliste/listeboks

```
<select id="lstKunder" size="1">
    <option value="34212">Ole Olsen</option>
    <option value="5514">Per Persen</option>
    <option value="21345">Nils Nilsen</option>
</select>
```

Antall elementer som er synlige av gangen

## Hente verdi fra nedtekksliste/listeboks

```
var kundenummer = document.getElementById("lstKunder").value
```

## Lage radioknapper

```
<input type="radio" name="kjonn" id="rbtnKjonnMann"
value="mann" />Mann<br />
<input type="radio" name="kjonn" id="rbtnKjonnKvinne"
value="kvinne" />Kvinne
```

Alle elementer i en gruppe må ha samme navn

## Sjekke om radioknapp er markert

```
if (document.getElementById("rbtnKjonnMann").checked === true) {
    // Kode
}
```

# Kontrollstrukturer

## Lage en if-test

```
Kodeordet if      Test   Test-operator
↓             ↓       ↓
if (antallBiler < 20) {
    // Her skrives koden for hva som skal
    // gjøres hvis testen slår til
}
```

## Lage en if-else-test

```
if (antallBiler < 20) {
    // Her skrives koden for hva som skal
    // gjøres hvis testen slår til
}
else {
    // Her skrives koden for hva som skal
    // gjøres hvis testen ikke slår til
}
```

## Lage en if-else if-else-test

```
if (antallBiler < 10) {
    // Koden som skal utføres hvis testen slår til
}
else if (antallBiler < 20) {
    // Koden som skal utføres hvis testen slår til og
    // testen/ene ovenfor ikke slår til
}
else {
    // Koden som skal utføres hvis ingen test slår til
}
```

## Relasjonsoperatorer

== er lik

!= ikke lik (forskjellig fra)

< mindre enn

> større enn

<= mindre enn, eller lik

>= større enn, eller lik

### if-test med *og*

```
Og  
if (antallBiler > 5 && antallBiler < 20) {  
    // Kode som skal utføres hvis testen slår til  
}
```

### if-test med *eller*

```
Eller  
if (antallBiler < 10 || antallBiler > 20) {  
    // Kode som skal utføres hvis testen slår til  
}
```

### Lage en while-løkke

```
Variabel vi tester på  
↓  
var teller = 0;  
Kodeordet while Test  
↓  
while (teller < 5) {  
    // Kode vi skriver her kjøres så mange ganger som vi  
    // går gjennom løkka (i dette eks. 5 ganger)  
    // Forandre verdi på variabelen  
    teller++;  
}
```

### Lage en for-løkke

```
Kodeordet for  
↓  
for (var teller = 0; teller < 5; teller++) {  
    // Kode vi skriver her kjøres så mange ganger som vi  
    // går gjennom løkka (i dette eksemplet 5 ganger)  
}
```

# Funksjoner

## Lage en funksjon som returnerer en verdi

```
Kodeordet      Funksjonsnavn (skal alltid  
function        ha liten forbokstav)          Parametere (variabler som overfører  
                                              verdier til funksjonen). Flere parametere  
                                              skiller med komma.  
  
function leggSammen(tall1, tall2) {  
    // Kode inne i funksjonen  
    // Lokal variabel (gjelder ikke utenfor funksjonen)  
    var sum = tall1 + tall2;  
    return sum;  
}  
Kodeordet return   Verdi som returneres
```

## Bruke (kalle opp) funksjonen

```
Variabel som settes lik det  
funksjonen returnerer          Funksjonskall          Argumenter  
↓                                ↓                         ↓  
var minSum = leggSammen(5, 3);  
  
alert(minSum); // 8           Verdier vi overfører  
                           til funksjonen
```

## Lage en funksjon som ikke returnerer en verdi

```
function melding(navn) {  
    alert("Hei på deg din " + navn + "!");  
}
```

## Bruke (kalle opp) funksjonen

```
melding("Padde"); // Hei på deg din Padde!
```

# Arrayer

## Lage en array

```
var minArray = [ ]; // Array definert uten verdier
```

## Legge inn verdier (elementer) i en array

```
Indeks  
↓  
minArray[0] = "Brød";  
minArray[1] = "Melk";  
minArray[2] = "Juice";
```

## Få tak i en verdi i en array

```
var verdi = minArray[1];  
alert(verdi); // Melk
```

## Iterere gjennom en array

```
Antall elementer i arrayen  
↓  
for (var teller = 0; teller < minArray.length; teller++) {  
    // Gjør noe med hvert element, for eksempel skrive ut:  
    var verdi = minArray[teller];  
    alert(verdi);  
}
```

## Lage en array og samtidig legge inn verdier

```
Indeks 0      Indeks 1      Indeks 2  
↓            ↓            ↓  
var minArray = ["Brød", "Melk", "Juice"];
```

## Legge til en verdi (element) i slutten av en array

```
minArray.push("Peanøtter");                            // Alternativ 1  
minArray[minArray.length] = "Peanøtter"            // Alternativ 2
```

## Slette elementer i en array

```
Start-indeks    Antall elementer vi vil slette (fra og med start-indeksten)  
↓                ↓  
minArray.splice(2, 1);
```

## Søke etter et element i en array

```
Får tak i indeksen til elementet vi finne. Hvis det ikke blir funnet, blir verdien -1  
↓  
var indeks = minArray.indexOf("Melk"); ↓ Elementet vi vil finne
```

## Lage en to-dimensjonal array

```
var minArray = [ ];  
      0   1   2 |  
minArray[0] = [5, 31, 7]; | 0  
minArray[1] = [2, 9, 22]; | 1  
minArray[2] = [50, 72, 4]; | 2
```

## Få tak i en verdi i en to-dimensjonal array

```
Indeks i "ytterste" array ↓  
Index i "innerste" array ↓  
var verdi = minArray[1][2];  
alert(verdi); // 22
```

## Assosiative arrayer

### Lage og sette verdier i en assosiativ array

```
var person = {navn:"Olsen", alder:22, yrke:"snekker"};
```

### Få tak i en verdi i en assosiativ array

```
var alder = person.alder; // Objekt-måten
```

```
var alder = person["alder"]; // Array-måten
```

### Lage en array som inneholder assosiative arrayer (to-dim.)

```
var personer = [];  
personer[0] = {navn:"Olsen", alder:22, yrke:"snekker"};  
personer[1] = {navn:"Lia", alder:43, yrke:"politi"};  
personer[2] = {navn:"Hansen", alder:34, yrke:"lærer"};  
  
// Får tak i en verdi i arrayen  
var yrke = personer[1]["yrke"];  
alert(yrke); // Politi
```

## Iterere gjennom en assosiativ array

```
for (var noekkel in person) {  
    alert("Nøkkel: " + noekkel + ",verdi: " + person[noekkel]);  
}
```

## Sjekke om en nøkkel finnes i en assosiativ array

```
if ("navn" in person) {  
    // Kode  
}
```

# Hendelser

## Registrere en lytterfunksjon

```
Element som kan utløse hendelse  
Navn på hendelse  
Navn på funksjon som skal kelles opp  
hvis hendelsen utløses (lytterfunksjon)  
  
document.getElementById("btnSjekk").onclick = sjekk;
```

## Lage en lytterfunksjon

```
Hendelsesobjekt (inneholder info om hendelsen). Trenger ikke være med om det ikke benyttes  
function sjekk(evt) {  
    // Her skrives koden som skal kjøres når hendelsen  
    // utløses  
}
```

## Fjerne en lytterfunksjon

```
document.getElementById("btnSjekk").onclick = null;
```

## Musehendelser

- **onclick** – brukeren klikker med musa.
- **ondblclick** – brukeren dobbeltklikker med musa.
- **onmousedown** – brukeren trykker museknapp ned.
- **onmouseup** – brukeren slipper museknapp opp.
- **onmousemove** – brukeren flytter musa over elementet. Denne hendelsen utføres hver gang musepekeren flytter seg til et nytt punkt, så potensielt et stort antall ganger på musepekerens vei over et objekt.
- **onmouseenter** – brukeren flytter musa fra et sted utenfor elementet til et sted over elementet.
- **onmouseleave** – brukeren flytter musa fra et sted over elementet til et sted utenfor elementet.
- **onmouseover** og **onmouseout** – fungerer som onmouseenter og onmouseleave, bortsett fra at de utføres på nytt for barn til elementer. Et onmouseover vil derfor bli utført både når musepekeren kommer inn over selve elementet og for hvert barn vi sveiper over og når vi går tilbake til hovedelementet igjen. Et onmouseout vil utføres når vi går fra hovedelementet og over på et barn til elementet, og flytter musa tilbake fra et barn.
- **onwheel** – fanger opp bevegelser til scrollhjulet på musa.

## Informasjon i hendelsesobjektet til musehendelser

- `clientX` og `clientY` - koordinatene til musa i nettleserens koordinatsystem. Øverst til venstre på nettsiden er 0,0.
- `screenX` og `screenY` - koordinatene til musa i skjermens koordinatsystem. Øvest til venstre på skjermen er 0,0.
- `button` - gir deg hvilken museknapp som ble trykket. 0 er venstre, 1 er midten/musehjul og 2 er høyre. Er musen konfigurert for venstrehendte i brukerens system, endres verdiene tilsvarende.
- `altKey`, `ctrlKey` og `shiftKey` - gir deg en true-/false-verdi på om angitte tast var nedtrykt på tastaturet mens musehendelsen ble utført.
- `detail` - gir deg en tallverdi på hvor mange ganger i en sammenhengende serie som museknappen ble trykket. Altså 2 for dobbeltklikk og 3 for trippelklikk.
- `deltaX`-den horisontale scrollingen på musehjulet.
- `deltaY`-den vertikale scrollingen på musehjulet.

## Drag and drop

```
document.getElementById("draelement").ondragstart = dra;
document.getElementById("slippelement").ondrop = slipp;
document.getElementById("slippelement").ondragover = tillat;

function dra(evt) {
    evt.dataTransfer.setData("text", evt.target.id);
}

function tillat(evt) {
    evt.preventDefault();
}

function slipp(evt) {
    evt.preventDefault();
    var elementId = evt.dataTransfer.getData("text");
    // Kode
}
```

## Tastaturhendelser

- `onkeydown` - når en tast trykkes ned (også systemtaster).
- `onkeyup` - når en tast slippes opp (også systemtaster).
- `onkeypress` - når en tast trykkes ned. Gjentas regelmessig så lenge tasten holdes nedtrykt (ikke systemtaster).

## Informasjon i hendelsesobjektet til musehendelser

- `keyCode` - gir deg et nummer som representerer tasten på tastaturet som ble trykket.  
`charCode` - gir deg et nummer som representerer tegnet som blir trykket på tastaturet.  
Dette er Unicode-verdien til tegnet.
- `altKey`, `ctrlKey` og `shiftKey` - gir deg en true-/false-verdi på om angitte tast var nedtrykt på tastaturet sammen med den andre tasten som hendelsen indikerer.

## Starte en tidsstyrte hendelse

```
Global variabel      Funksjon som skal utføres   Antall millisekunder mellom hver utførelse  
↓                ↓                            ↓  
intervallID = setInterval(tikk, 200);  
  
function tikk() {  
    // Kode  
}
```

## Stoppe en tidsstyrte hendelse

```
Global variabel med referanse til intervallet som er startet  
↓  
clearInterval(intervallID);
```

## Grafikk

### Lage canvas

```
<canvas id="tegneflate" width="200" height="200"></canvas>
```

### Hente ut context

```
var ctx = document.getElementById("tegneflate").getContext("2d");
```

### Tegne en linje

```
ctx.beginPath();  
ctx.moveTo(50, 100); // x-verdi og y-verdi  
ctx.lineTo(100, 200); // x-verdi og y-verdi  
ctx.stroke();
```

HUSK: x-verdi måles fra venstre kant, y-verdi måles fra toppen.

### Tegne en sirkel

```
ctx.beginPath();  
ctx.arc(50, 50, 25, 0, 2 * Math.PI); // x-verdi, y-verdi, diameter,  
                                         sirkelstart og sirkelslutt  
ctx.stroke();
```

## Tegne et rektangel

```
ctx.beginPath();
ctx.rect(50, 100, 100, 100); // x-verdi, y-verdi, bredde og høyde
ctx.stroke();
```

## Endre linjefarge og tykkelse

```
ctx.strokeStyle = "red"; // En CSS-farge
ctx.lineWidth = 4;
```

NB! Gir kun effekt til det vi tegner etter endringen

## Tegne fyll

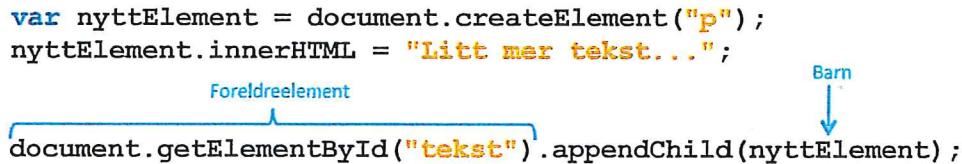
```
ctx.fillStyle = "red";
ctx.fill();
```

NB! Må gjøres før vi evt. tegner kantlinje med stroke()

## Endre HTML-dokumentet

### Lage et nytt element og legge det til i dokumentet

```
var nyttElement = document.createElement("p");
nyttElement.innerHTML = "Litt mer tekst...";
document.getElementById("tekst").appendChild(nyttElement);
```



### Fjerne et element fra dokumentet

```
document.getElementById("tekst").removeChild(nyttElement);
```



## Endre CSS

### Sette nye stiler direkte

```
document.getElementById("tekst").style.color = "red";
document.getElementById("tekst").style.display = "none";
document.getElementById("tekst").style.borderStyle = "solid";
```

### Endre CSS-kasse

```
document.getElementById("ingress").className = "roedtekst";
```

### Fjerne CSS-kasse

```
document.getElementById("ingress").className = "";
```

## Lyd og video

### Spille av lydfil

```
var lyd = new Audio("lydfil.mp3");
lyd.play();
```

### Sette inn lydavspiller

```
<audio controls="controls" id="lydavspiller">
  <source src="lydfil.mp3" type="audio/mpeg">
  <source src="lydfil.ogg" type="audio/ogg">
  Nettleseren støtter ikke lydavspilling
</audio>
```

### Sette inn videoavspiller

```
<video width="640" height="480" controls="controls"
id="videoavspiller">
  <source src="videofil.mp4" type="video/mp4">
  <source src="videofil.ogg" type="video/ogg">
  Nettleseren støtter ikke videospilling
</video>
```

### Manipulere en avspiller

```
document.getElementById("lydavspiller").play();
document.getElementById("lydavspiller").pause();
document.getElementById("lydavspiller").currentTime = 0;
```

## Detektere avspillingsslutt

```
document.getElementById("lydavspiller").onended = slutt;  
  
function slutt() {  
    // Kode  
}
```

## Dynamisk innhold

Husk at du må kjøre prosjektet gjennom en webserver.

### Laste inn data fra fil

```
var xmlhttp;  
  
function oppstart() {  
    xmlhttp = new XMLHttpRequest();  
    xmlhttp.onreadystatechange = statusforandring;  
    xmlhttp.open("GET", "innhold.dat", true);  
    xmlhttp.send();  
}  
  
function statusforandring() {  
    if (xmlhttp.readyState === 4 && xmlhttp.status === 200) {  
        var data = xmlhttp.responseText;  
    }  
}
```

### Dele opp strukturert data

```
var data = xmlhttp.responseText;  
var linjer = data.split("\n");  
  
for (var i = 0; i < linjer.length; i++) {  
    var datafelt = linjer[i].split("%");  
  
    // Gjør noe med verdiene i arrayen datafelt  
}
```

### Hente ut XML-data

```
var xmlData = xmlhttp.responseXML;
```

## Hente ut verdier og attributtverdier fra XML-data

```
var verdi = xmlData.getElementsByTagName("pris") [0].  
    childNodes [0].nodeValue;  
  
var verdi2 =  
xmlData.getElementsByTagName("bil") [0].attributes.  
    getNamedItem("regnr").value;
```

## Hente ut JSON-data

```
var jsonData = JSON.parse(xmlhttp.responseText);
```

## Hente ut verdier fra JSON-data

```
var verdi = jsonData.datafelt;  
  
var verdi2 = jsonData.datafelt2.dataliste [0].datafelt
```

## Skrive til web storage

```
localStorage.setItem("navn", "Ole Olsen");
```

## Lese fra web storage

```
var navn = localStorage.getItem("navn");  
if (navn === null) {  
    navn = "ukjent";  
}
```