

Referanseark JavaScript

Variabler og datatyper

«Enkle» variabler har én av datatypene tekst (string), tall eller boolean:

```
let tall = 72;  
let annetTall = 3.14;  
let tekst = "Hei";  
let ferdig = false;
```

Gjør om en variabel til datatypen tall (så lenge innholdet kan bli til et tall):

```
Number(variabel)
```

Operatorer

+	Addisjon
-	Subtraksjon
*	Multiplikasjon
/	Divisjon
%	Modulus (rest)
++	Inkrement (øke med 1)
--	Dekrement (minke med 1)

Objekter

Vi bruker sløyfeparentes for å lage et objekt, og komma for å skille egenskapene:

```
let frukt = {  
  type: "banan",  
  farge: "gul"  
}
```

Henter ut enkeltverdier fra et objekt:

```
frukt.type eller frukt["type"]
```

Valg

```
if (betingelse) {  
  // kode  
} else {  
  // kode  
}
```

```
if (betingelse 1) {  
  // kode  
} else if (betingelse 2) {  
  // kode  
} else {  
  // kode  
}
```

Sammenliknings- og logiske operatorer

==	Lik verdi
===	Lik verdi og lik datatype
!=	Ikke lik verdi
!==	Ikke lik verdi eller datatype
>	Større enn
<	Mindre enn
<=	Mindre enn eller lik
>=	Større enn eller lik
&&	Og (begge betingelser er sanne)
	Eller (én eller flere av betingelsene er sanne)
!	Betingelsen er ikke sann

Løkker

En **while**-løkke lar oss gjenta noe så lenge en gitt betingelse er sann (**true**):

```
while (betingelse) {  
  // kode  
}
```

En **for**-løkke styres av en startvariabel, en betingelse og hvordan variabelen skal endres for hver gjennomgang:

```
for (let i = 0; i < 10; i++) {  
  // kode  
}
```

Løkker og objekter

Går gjennom egenskapsnavnene til et objekt:

```
for (let egenskap in objekt) {  
  console.log(egenskap);  
}
```

Går gjennom verdiene til egenskapene i et objekt:

```
for (let egenskap in objekt) {  
  console.log(objekt[egenskap]);  
}
```

Funksjoner

Lager en enkel funksjon:

```
function funksjonsNavn() {  
  // kode  
}
```

Funksjoner kan ha en eller flere parametre:

```
function arealRektangel(lengde, bredde) {  
  let areal = lengde * bredde;  
  console.log("Arealet blir " + areal + ".");  
}
```

Funksjoner kan ha en returverdi:

```
function arealRektangel(lengde, bredde) {  
  let areal = lengde * bredde;  
  return areal;  
}
```

Nøkkelordet `return` kan også brukes til å avslutte en funksjon.

Arrayer

Vi bruker hakeparenteser for å lage en array:

```
let partall = [2, 4, 6, 8];
```

Finner en enkeltverdi i en array (husk at vi teller fra 0):

```
partall[2] // Gir oss 6
```

Finner lengden til en array:

```
partall.length;
```

Går gjennom alle verdiene i en array:

```
for (let i = 0; i < partall.length; i++) {  
  console.log(partall[i]);  
}
```

Eller:

```
for (let tall of partall) {  
  console.log(tall);  
}
```

Arraymetoder:

push(verdi)	Legger til en verdi bakerst i en array
pop()	Fjerner bakerste verdi i en array
unshift(verdi)	Legger til en verdi først i en array
shift()	Fjerner første verdi i en array

Vi har også metoden **splice()** som kan motta flere parametre:

- Første parameter angir indeksen vi begynner med.
- Andre parameter angir antall verdier som skal fjernes (kan være 0).
- Alle parametre etter disse angir verdier som skal legges til ved gjeldende indeks.

Sortere arrayer

Sortere en array med tall:

```
let tall = [7, 98, 13, 142];  
  
function sammenliknTall(a, b) {  
  return a - b;  
}  
  
tall.sort(sammenliknTall);
```

Sortere en array med objekter etter en gitt egenskap (her `navn` i en array med land-objekter):

```
function sammenliknNavn(a, b) {  
  if (a.navn > b.navn) {  
    return 1;  
  } else if (a.navn < b.navn) {  
    return -1;  
  } else {  
    return 0;  
  }  
}
```

```
land.sort(sammenliknNavn);
```

Math-objektet

Tilfeldig tall (fra og med 0 til og, men ikke med, 1):

```
Math.random();
```

Avrunding (`floor` runder ned og `ceil` runder opp):

```
Math.floor(tall);  
Math.ceil(tall);
```

PI:

```
Math.PI;
```

Document Object Model (DOM)

Henter første element med angitt CSS-selektor:

```
let mittEl = document.querySelector("CSS-selektor");
```

Henter alle elementer med angitt CSS-selektor (og får en array i retur):

```
let mineEls = document.querySelectorAll("CSS-selektor");
```

Endrer innholdet i et HTML-element:

```
mittEl.innerHTML = "Nytt innhold";
```

Legger til et nytt HTML-attributt:

```
mittEl.setAttribute("attributtnavn", "verdi");
```

Fjerner et HTML-attributt:

```
mittEl.removeAttribute("attributtnavn");
```

Angir attributter for et HTML-element:

```
mittEl.className = "klassenavn";  
mittEl.id = "id-navn";
```

Legger til et **data**-*-attributt (her **data-nummer**):

```
mittEl.dataset.nummer = "4";  
eller:
```

```
mittEl.setAttribute("data-nummer", "4");
```

Angir CSS-egenskaper for et HTML-element direkte (husk at CSS-egenskaper med bindestrek skrives som ett ord med store forbokstaver der det ellers er bindestrek):

```
mittEl.style.backgroundColor = "#FF00FF";
```

Lager et nytt HTML-element (her et avsnitt):

```
let nyttEl = document.createElement("p");
```

Legger til et HTML-element i et eksisterende HTML-element:

```
mittEl.appendChild(nyttEl);
```

Hendelser

Legger til en lytter på et element (funksjonen kalles når hendelsen inntreffer):

```
mittEl.addEventListener("click", funksjonsNavn);
```

Tar bort en lytter:

```
mittEl.removeEventListener("click", funksjonsNavn);
```

Hendelsesobjektet inneholder informasjon om en hendelse. Vi kan for eksempel finne elementet hendelsen skjedde på (med **e.target**).

Utvalgte hendelser:

click	Et element blir klikket på
keydown	En tast trykkes
keyup	En tast slippes
mousemove	Musepekeren beveges
mouseout	Musepekeren forlater et element
mouseover	Musepekeren føres over et element
input	Et input- eller textarea-element endres
change	Et select-element endres

Informasjonskapsler

Vi lager en informasjonskapsel i *localStorage* ved å skrive:

```
localStorage.variabelnavn = "verdi";
```

Vi kan sjekke om en informasjonskapsel ved å skrive:

```
if (localStorage.variabelnavn) {  
    // Informasjonskapselen eksisterer  
} else {  
    // Informasjonskapselen eksisterer ikke  
}
```

For å slette en informasjonskapsel, kan vi bruke **removeItem()**, slik:

```
localStorage.removeItem("variabelnavn");
```

Vi kan fjerne alle informasjonskapsler vi har satt for nettsiden vår, ved å skrive:

```
localStorage.clear();
```