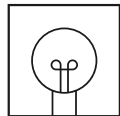
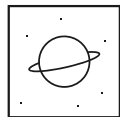


# JavaScript

# Interaktive nettsider!



Ble laget på ti dager av  
Brendan Eich i 1995



Først til Netscape, så senere til  
andre nettlesere



Standardisert i 1997 –  
EcmaScript!



# Hvordan komme i gang

JavaScript kan brukes direkte i HTML med <script>:

```
<script type="text/javascript">  
|...console.log('Hei, verden!');  
</script>
```

Men det er ryddig å referere til eksterne filer:

```
<body>  
|...<script type="text/javascript" src="filnavn.js"></script>  
</body>
```

# Variabler



Dynamisk typesetting

Typer bestemmes automatisk og kan endre seg under kjøring

Tre ulike måter å deklarere variabler på:

```
var a = 1; // function-scoped variabel  
let b = 2; // block-scoped variabel  
const c = 3; // block-scoped konstant
```

# Datatyper

Syv datatyper

boolean

null

number

undefined

string

symbol

Object



Hva med lister/arrays?

Lister er egentlig objekter!

# Strenger

Vi kan deklarere en streng på to måter:

```
const dobbeltfnutter = "JavaScript";  
const enkeltfnutter = 'JavaScript';
```

To måter med variabler: Konkaterering og *template literals*

```
const hilsen1 = 'Hei, ' + name + ', hvordan går det?';  
const hilsen2 = `Hei, ${name}, hvordan går det?`;
```

# Objekter

```
const webIntro = {  
  ...audience: 'Nyansatte',  
  ...year: 2019,  
}
```

Hent egenskaper ved et objekt:

```
webIntro.audience; ..... // 'Nyansatte'  
webIntro['audience']; ..... // 'Nyansatte'  
webIntro['description']; .. // undefined
```

Sett egenskaper:

```
webIntro.year = 2020;  
webIntro.description = 'Introkurs i webprogrammering';
```

# Booleans

Følgende er «falsy»

- false
- 0
- ""
- null
- undefined
- NaN

Alt annet er «truthy»! 😎

- [] (tom liste)
- {} (tomt objekt)
- ...



# Funksjoner

Kan definere en funksjon  
på flere ulike måter:

```
function functionStatement(x) {  
  ...return x;  
}
```

```
const functionExpression = function(x) {  
  ...return x;  
};
```

```
const arrowFunction = (x) => { return x; };
```

```
// Implisitt returverdi  
const pluss = (a, b) => a + b;
```

```
// Uten parantes  
const doble = a => a * 2;
```

# Lister

Ingen distinksjon mellom «array» og «list»

```
const listeMedTall = [1, 2, 3];  
const listeMedStrenger = ["Hei", "på", "deg"];  
const listeMedBeggeDeler = ["Hei", 2, 3];  
  
const listeMedObjekter = [  
  { x: 1, y: 4 },  
  { x: 3, y: 2 },  
];  
  
const nøstetListe = [1, [2, [3]]];
```

# Listeoperasjoner

## Map

```
const doble = n => n * 2;  
  
[1, 2, 3].map(doble); // [2, 4, 6]  
[1, 2, 3].map(n => n * n) // [1, 4, 9]
```

## Filter

```
const oddetall = [1, 2, 3, 4].filter(n => n % 2); // [1, 3]
```

## Reduce

```
const sum = [1, 2, 3].reduce((delsum, n) => delsum + n, 0); // 6
```

# Document Object Model

JavaScript gir oss et API for å interagere med DOMen via **document**-objektet:

```
<div id="boks">
```

```
|...<a href="mer.html">Klikk her</a> for å lære mer
```

```
</div>
```

```
const boks = document.querySelector('#boks');
```

```
boks.innerHTML; // <a href="mer.html">Klikk her</a> for å lære mer
```

```
boks.textContent; // Klikk her for å lære mer
```

# Hendelser («events»)

Vi kan også lytte til «events» når noe skjer med et DOM-element, f.eks.

- Et klikk på en knapp
- Brukeren skriver noe i et tekstfelt
- Musepekeren beveger seg
- ... og mye, mye mer!

```
document.querySelector('#boks').addEventListener('click', () => {  
  ... // Gjør noe gøy når boks har blitt trykket på!  
});
```

# Opplegget



Her finner dere  
oppgavene

<https://bekk.gitbook.io/web-intro/>



Kan du noe fra før eller  
har forstått greia?

Hopp over det!



Ikke vær redd for å  
spørre om hjelp

#frontend på slack