

# Events

# JavaScript kode

- ❖ JavaScript kode kan indsættes i en .html fil:
  - ❖ i en HTML event attribut (on...) i en HTML fil
  - ❖ i et `<script>` element
- ❖ Men JavaScript kode bør placeres i en separat .js fil:
  - ❖ lokalt med `<script src="...">`
  - ❖ på en anden web server med `<script src="https://...">`
- ❖ JavaScript i en separate .js fil har disse fordele:
  - ❖ lagdeling med adskillelse af JavaScript og HTML
  - ❖ genbrug af koden på flere websider
  - ❖ ændring i koden kun et sted

# JavaScript udføres i 2 faser

I indlæsningsfasen opbygges DOM'en og dernæst udføres JavaScript koden i `<script ... defer>`

❖ hvis der er flere `<script ... defer>` elementer, udføres de efter hinanden i den skrevne rækkefølge

I event-håndteringsfasen udføres en eller flere event handler(s) ved events

Event handler properties og metoder:

```
element.onevent = handler;  
element.addEventListener("event", handler);  
element.removeEventListener ("event", handler);
```

Events:

❖ onclick, onchange, onkeyup, onfocus, onblur, onmouseover, onmouseout, ...

```
<!-- input.html -->
<!DOCTYPE html>
<html>
  <head>
    <title>Input</title>
    <script src="input.js" defer></script>
  </head>
  <body>
<input>
    <button>+1</button>
  </body>
</html>
```

```
// input.js
let knap = document.querySelector("button");
let input = document.querySelector("input");
knap.onclick = () => input.value++;
```

# Listen af events

- ♣ onclick
- ♣ onchange
- ♣ onkeyup
- ♣ onfocus
- ♣ onblur
- ♣ onmouseover
- ♣ onmouseout
- ♣ ...

```
element.onclick = handler
```

```
element.addEventListener("event", handler)
```

```
element.removeEventListener("event", handler)
```

# Event håndtering

Javascript udfører event handlers i en enkelt tråd - i den rækkefølge events sker.

En langvarig event handler blokerer for, at andre events håndteres.

Derfor bør events håndteringen være *kortvarig*

Langvarige opgaver kan udføres i en separate tråde.

# Eksempel

```
<!-- langvarig.html -->
<!DOCTYPE html>
<html>
  <head>
    <title>Langvarig</title>
    <script src="langvarig.js" defer></script>
  </head>
  <body>
    <button>Start langvarig</button>
  </body>
</html>
```

```
// langvarig.js
function langvarig() {
  let start = Date.now();
  console.log(start % 100000);
  while (true){
    let nu = Date.now();
    if ((nu - start) >= 2000) return;
  }
}
let knap = document.querySelector("button");
knap.onclick = langvarig;
```

# Event handler

- ❖ Et event vandrer opad i DOM træet, fra det element hvorpå event'en er sket og videre op til body, document og window
- ❖ En event handler får som første parameter et event objekt, som har flere properties og metoder:
  - ❖ type – elementets type
  - ❖ target – elementet der udløser event'et
  - ❖ currentTarget – elementet med den kaldte event handler
  - ❖ stopPropagation() – forhindrer vandring
  - ❖ ...
- ❖ I en event handler er this det samme som currentTarget



# Eksempel

```
<!-- propagation.html -->
<!DOCTYPE html>
<html>
  <head>
    <title>Propagation</title>
    <script src="propagation.js" defer></script>
  </head>
  <body>
    <h3 id="xyz">Overskrift</h3>
    Tekst
  </body>
</html>
```

```
// propagation.js
let h3 = document.querySelector("h3");
function propagation(event) {
  console.log(event.type);
  console.log(event.target.id);
  console.log(event.target);
  console.log(this);
  event.target.style.color = 'red';
  this.style.color = 'blue';
  // event.stopPropagation();
}
h3.onclick = propagation;
document.body.onclick = propagation;
```