

Svelte Begynderguide

Velkommen! Til denne begynderguide til Svelte. Står du i den situation at du aldrig har hørt om Svelte før men tilfældigvis er faldet over det og nu får du lyst til at prøve kræfter med det, men du har ingen ide om hvordan du skal begynde? Så er du kommet til det rigtige sted! Også selvom du ikke står i overstående situation.

I denne begynderguide kommer vi til at kigge på hvad Svelte er, hvad det kan, hvordan det fungerer og hvordan det sættes op.

Hvad er Svelte?

Svelte er et komponent framework ligesom React. Svelte består af at opbygge en webapplikation ved hjælp af Svelte filer og bruger efterfølgende en compiler til at binde det sammen.

Hvordan udtales Svelte?

Jeg troede til at starte med at det blev udtalt Svelté med tryk på e'et til sidst, men det udtales "Svelt".

Med den meget vigtige info lad os komme i gang.

Hvad indeholder et .svelte fil?

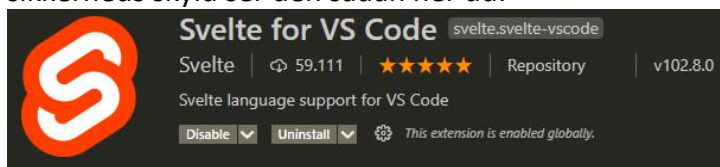
En .svelte fil er bygget op af ren HTML. Den kan indholde et script-tag et style-tag og ellers alle de HTML tags som du har lyst til at skrive. Vi laver en fil senere i guiden som viser hvordan det ser ud.

Hvordan fungerer Svelte?

Det fungerer således at den compiler vi har valgt tager vores Svelte filer og binder dem sammen for så derefter at omdanne dem til vanilla javascript som alle browsere kan forstå. Dette er med til at vi kan skrive mindre kode hurtigere og "bruge mere tid udenfor" som Svelte-skaberen selv forklarer.

Allerførst:

Vi kommer til arbejde i Visual Studio Code så det allerførste vi skal gøre er at hente en extension som senere gør at Visual Studio kan forstå de .svelte filer vi kommer til at lave. Den extension vi skal hente hedder "Svelte for VS Code" og den er af Svelte. For en sikkerheds skyld ser den sådan her ud:



Når vi har den installeret den, er vi klar til at begynde.

Der er 2 måder at opbygge dette projekt på.

Opbygning A - Template:

Vi kan hente en Svelte template, hvor det hele er sat op for os og hvor vi bare kan gå i gang med at kode. Dette opnås ved at skrive følgende i vores terminal:

```
npx degit sveltejs/template
```

Således henter vi den seneste version af GitHub repoet men ved at skrive "degit" foran, så slipper vi for at få alt historien fra repo'et med.

Opbygning B - Fra bunden:

Det første vi skal gøre når vi starter er at hente den compiler der hedder Rollup (Du kan bruge hvilken som helst compiler - Der ligger en masse ude på nettet men i denne guide fokuserer vi på Rollup). Dette gør vi ved at vi skriver følgende i vores terminal:

```
npm i -D rollup
```

Dette installerer Rollup compileren og laver en package-lock.json fil til dig som hvor alt indholdet ligger i.

Det næste vi skal, er at lave en ny fil og kalde den rollup.config.js - Denne skal bruges så vi kan få vores compiler til at virke.

I filen skriver vi følgende:

```
export default {  
  input: 'src/main.js',  
  output: {  
    file: 'public/build/bundle.js',  
    format: 'esm',  
    sourcemap: true  
  }  
};
```

Efterfølgende skal vi lave en src mappe og i den skal vi have tilføjet en main.js fil.

Når vi har gjort det kan vi i vores terminal skrive følgende:

```
echo "import hello from './hello.js';  
  
hello();" > src/main.js  
  
echo "export default () => {  
  console.log('hello!');  
}" > src/hello.js
```

Dette gør 2 ting. Til at starte med bliver der i vores main.js importeret en function fra et hello.js komponent. Dette er vores test komponent som vi skal bruge for at se om vi kan få hul igennem. Terminalen skriver således vores hello.js og fylder det ud for os.

Til allersidst skal vi køre følgende i vores terminal:

```
npx rollup -c
```

Så er vi ved at være klar til at gå rigtig i gang. Overstående starter vores bundler hvis kigger ude din filer skulle vi nu meget gerne have fået en public mappe med bundle.js og bundle.js.map - Hvis vi åbner bundle.js filen kan vi se at den har samlet vores hello.js og main.js til én JavaScript fil.

Nu tænker du sikkert “Vent? Mangler der ikke noget meget essentielt?” og jo, det gør der, og det filføjer vi nu.

I vores public mappe skal vi lave en index.html fil - Når vi har lavet den kan vi kopierer følgende ind i filen:

```
<!DOCTYPE html>
<html lang="en"><head>
<meta charset='utf-8'>
<meta name='viewport' content='width=device-width,initial-scale=1'>

<title>Svelte Begynderguide</title>

<script type="module" src='/build/bundle.js'></script>
</head>
<body>
Åben konsollen (f12)!
</body>
</html>
```

“Så, nu er vi klar til at se om det virker” hører jeg dig sige, men ikke helt endnu, vi mangler at installerer en sidste ting.

I vores terminal skriver vi:

```
npm i -D serve
```

Og efterfølgende:

```
npx serve public
```

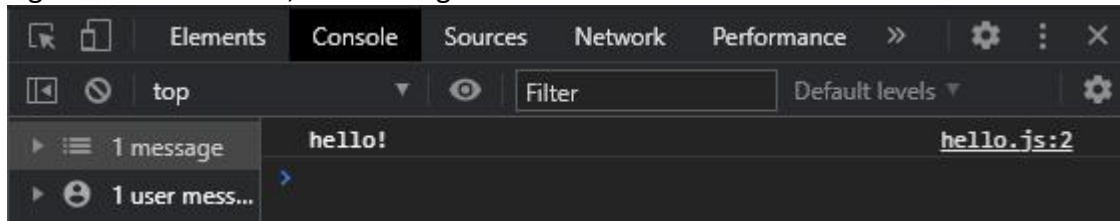
Dette installerer Serve som gør at vi kan få lov til at se vores projekt på localhost og når vi taster “npx serve public” smider den vores public mappe op.

Som default bruger den localhost:5000 så hvis vi åbner en browser på den side burde vi genre kunne se følgende:



Åben konsollen (f12)!!

Og åbner vi konsollen, skulle der gerne være en besked klar til os.



Tip: Hvis du har en liveserver extension i Visual Studio Code kan blot bruge den.

Tillykke! Hvis du får noget frem i din browser og i konsollen betyder det at du har sat det hele rigtig op! Men som du nok tænker så har vi slet ikke kigget på Svelte endnu og det er ganske rigtig! Det er det næste vi kommer til.

Få Rollup til at forstå hvad .svelte er:

Desværre så forstår rollup ikke hvad en .svelte fil er automatisk, dette skal vi lære den og måden vi gør det på er at vi i vores terminal skal installere 4 plugins.

Det første er selve Svelte, efterfølgende er rollup-plugin-svelte og @rollup/plugin-node-resolve (Hjælper rollup med at finde rundt i node_modules) og til sidst rollup-plugin-css-only (Som er et workaround til når der tilføjes css til bundleren)

Vi kan installere det hele på én gang ved at vi i vores terminal skriver:

```
npm i -D svelte rollup-plugin-svelte @rollup/plugin-node-resolve rollup-plugin-css-only
```

Når de er installeret skal vi tilføje dem til vores rollup.config.js - Den skal nu se sådan her ud:

```
import resolve from '@rollup/plugin-node-resolve';
import svelte from 'rollup-plugin-svelte';
import css from 'rollup-plugin-css-only'

const production = !process.env.ROLLUP_WATCH;

export default {
  input: 'src/main.js',
  output: {
    file: 'public/build/bundle.js',
    format: 'esm',
    sourcemap: true
  },
  plugins: [
    resolve(),
    svelte({
      compilerOptions: {
        // enable run-time checks when not in production
        dev: !production
      }
    }),
    css({ output: 'bundle.css' })
  ]
};
```

Når vi har sat dette op er vi klar til at lave vores første .svelte komponent.

I din src mappe lav en ny fil og kald den App.svelte - Dette kommer til at være din hovedfil og alt kommer til at kører herfra. Til at starte med holde den simpel og blot H1 tag med en prop og en simpel styling:

```
<script>
export let name;
</script>

<h1>Hello {name}!</h1>

<style>
h1 {
font-family: 'Franklin Gothic Medium';
}
</style>
```

Vi skal nu ind i vores main.js og rette den således at vi fjerner vores hello() function og erstatter det med JavaScript som fortæller vores compiler at den skal bygge vores App.svelte i stedet for nu. Din main.js skal se sådan her ud nu:

```
import App from './App.svelte';

new App({
  target: document.body,
  props: {
    name: 'world'
  }
});
```

Under props skriver vi hvad vi godt kunne tænke os skal stå på {name} pladsen i vores App.svelte - i dette tilfælde vil vi gerne have vores prop til at være "world".

Den sidste fil vi mangler at redigere nu er vores index.html - Her skal vi ind og tilføje et link til den bundle.css som bliver oprettet på grund af en af de plugins vi har installeret tidligere. Din index.html skal derfor se sådan her ud nu:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset='utf-8'>
  <meta name='viewport'
content='width=device-width,initial-scale=1'>
  <title>
    Svelte Begynderguide
  </title>
  <link rel="stylesheet" href="/build/bundle.css">
  <script type="module" src="/build/bundle.js"></script>
</head>

<body>
</body>

</html>
```

Det sidste vi mangler at gøre nu, er at køre

```
npx rollup -c
```

I vores terminal igen. Vi skal faktisk skrive det hver gang at vi ændre eller tilføjer noget i vores projekt.

Hvis vi tager et kig i vores public mappe nu skulle vi gerne se at bundle.js er blevet opdateret til at indeholde vores App.svelte, samt skulle der også gerne være en bundle.css som indeholder alt vores css.

Det der forhåbentlig gerne skulle ske nu er at hvis vi åbner vores server med

```
npx serve public
```

Skulle det gerne se sådan her ud i browseren.



Hello world!

Tillykke igen!

Du ved nu hvordan man sætter Svelte op fra bunden og bygger dit helt eget svelte komponent!

Jeg vil anbefale dig nu at lege lidt rundt med det og se hvad du kan få frem.

Hvis du har lyst, kan du også prøve kræfter med svelte's egen tutorial på <https://svelte.dev/tutorial/basics> den går meget mere i dybden og gennemgår alt det du kan med svelte og det er meget. Du kan vælge at bruge din eget projekt til at kode med, ellers har de i en editor i browseren som du kan gøre brug af.

Bonus eksempler

Nedenfor kunne jeg godt tænke mig at vise dig nogle få eksempler af ting du kan gøre med svelte.

Jeg nåede desværre ikke at tilføje nogle bonus eksempler.. Dem jeg ville have tilføjet kan ses i min sandbox på sveltesandbox.netlify.app