

Av: Nikita Buyondo

Lärare: Kristian Kjeldsen

IP2 FE20 – Redogörelse

Länk till sida: <http://nikitabuyondo.surge.sh/>

12 februari 2021

Då var det dags igen..

Så för denna release av min egen portfolio ska vi gå igenom ett par saker.

Till att börja med så kollar vi på olika CSS-ramverk.

Ett CSS-ramverk är ett bibliotek som möjliggör enklare, mer standardkompatibel webbdesign med så kallad "Cascading Style Sheets". De flesta av dessa ramverk innehåller åtminstone ett rutnät. Mer funktionella ramverk kommer med fler funktioner samt med JavaScript-baserade funktioner. Det som skiljer CSS-ramverk mot JavaScript-ramverk är att CSS-ramverk är designorienterade och har fokus på interaktiva användargränssnitts mönster.

Det finns många olika CSS-ramverk som till exempel Bootstrap som är ett av de största men sen finns även Bulma, Foundation och många, många fler. I dagsläget så ser det ut som att de flesta hemsidor som finns uppe använder sig av någon typ av CSS-ramverk samtidigt är det också något som ständigt håller på att utvecklas.

Bootstrap är en av de mest använda just nu då det finns väldigt mycket handledningsvideos tillgängliga och att de även har valt att anpassa sitt ramverk för "mobile-first" som gör att man stajlar sin hemsida för mobilen först och sedan gör så att den även passar för surfplattor och datorer.

Bulma är ett annat ramverk som vuxit den senaste tiden på grund av att det är lätt att lära sig och lätt att använda. Jag har själv inte använt Bulma men tydligen ska designen vara enkel och stilren samt att koden ska vara lätt att läsa av.

Då jag har gjort min portfolio i React har jag valt att använda mig av Material-UI som innehåller gränssnitt från Googles Material Design. Det är ett väldigt enkelt och användarvänligt. Man får sedan en ganska klassisk och stilren design som jag var ute efter i just detta projekt. Andra som använder sig av Material-UI är NASA, Netflix, Amazon och Spotify.

SASS eller LESS? Det är frågan.

De båda är speciella formatmallstillägg som gör designen enklare och effektivare. Både SASS och LESS kompileras i CSS-formatmallar så att webbläsare kan läsa dem, detta p.g.a. moderna webbläsare inte kan läsa .sass eller .less filer.

Om vanlig CSS skrivs:

```
body {  
  font: 100% Helvetica, sans-serif;  
  color: #333;  
}
```

Så lägger man upp SASS på detta sätt:

```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;
```

```
body {  
  font: 100% $font-stack;  
  color: $primary-color;  
}
```

Om CSS skriv på detta sätt:

```
#header {  
  color: #6c94be;  
}
```

Så skriver du så här om du använde LESS:

```
@nice-blue: #5B83AD;  
@light-blue: @nice-blue + #111;  
  
#header {  
  color: @light-blue;  
}
```

JavaScript syntax är en uppsättning av regler som JavaScript program är uppbyggda på. JavaScript syntaxen definieras av två olika värden. Fasta värden som kallas för Literals och variabla värden som kallas för just Variabler.

De två viktigaste syntaxen för de fasta värdena är:

1. Nummer(siffror) som skrivs med eller utan decimaler. T.ex. 36 eller 58,73.
2. "Strängar" eller "Strings" som är en bit text och skriv inom enkla eller dubbla citattecken.
"Detta är en sträng." 'Och detta är också en sträng.'

Inom programmeringsspråk skriver vi variabler för att förvara värden. För att skapa variabler så måste vi först skriva let, var eller const till att börja med.

Så här kan det se ut:

```
let name = "Nikita"  
var lastname = "Buyondo"
```

Med hjälp av likamedstecken (=) så deklarerar vi ett värde till varje variabel.

Efter att vi har gett variabeln ett värde så kan vi sedan manipulera dess värde på många olika sätt och ge variabeln ett nytt värde.

De tre senaste Javascriptsversionerna är ECMAScript 2016, ECMAScript 2017 och ECMAScript 2018.

När ECMAScript 2016 lanserades så lade de till:

- JavaScript Exponentiation (**)
- JavaScript Exponentiation assignment (**=)
- JavaScript Array.prototype.includes

Med ECMAScript 2017:

- [JavaScript String padding](#)
- [JavaScript Object.entries](#)
- [JavaScript Object.values](#)
- [JavaScript async functions](#)
- JavaScript shared memory

Och med ECMAScript 2018:

- [Asynchronous Iteration](#)

- Promise Finally
- Object Rest Properties
- New RegExp Features

Standarden för JavaScript kallas ECMAScript. Från och med 2012 stödjer alla moderna webbläsare ECMAScript 5.1. Äldre webbläsare stödjer åtminstone ECMAScript 3. Den 17:e Juni 2015 publicerades den större versionen av ECMAScript, som officiellt kallas för ECMAScript 2015, men refererad ofta som ECMAScript 6 eller ES6. Sedan 2015-versionen släpps nya standarder årsvis.

JavaScript har inget kompileringssteg. Istället läser en tolk i webbläsaren över JavaScript-koden, tolkar varje rad och kör den. Mer moderna webbläsare använder en teknik som kallas Just-In-Time (JIT) kompilering, som sammanställer JavaScript till körbar bytecode precis som den är på väg att köras.

Det finns många olika bibliotek för JavaScript. Vi har Gatsby, Babel, Angular och många fler men för detta projekt har vi använt oss av React.js som är skapat av Facebook.

Med ett JavaScript-ramverk kan jag eller den som programmerar, manipulera funktionerna och använda dem lite hur vi vill. Ramar är mer anpassningsbara för utformningen av webbplatser, och därför föredrar webbutvecklare att använda sig utav dem. JavaScript-ramverk är verktyget som underlättar arbetet med JavaScript. Dessa ramverk gör det också möjligt för programmeraren att göra helheten mer responsiv.

DOMen är en viktig del av att göra webbplatser interaktiva. Det är ett gränssnitt som gör det möjligt för ett programmeringsspråk att manipulera innehållet, strukturen och stilen på en webbplats.

JavaScript är skriptspråket på klientsidan som ansluts till DOM i en webbläsare.

Sedan har vi den virtuella DOMen.

Ett virtuellt DOM-objekt har samma egenskaper som ett riktigt DOM-objekt, men det saknar den verkliga sakens kraft att direkt ändra vad som finns på skärmen. Manipuleringen av DOM är långsam. Manipuleringen av den virtuella DOM är mycket snabbare eftersom ingenting ritas på skärmen. Vi använder oss av Components.

När en ReactComponent ändrar tillstånd vill vi göra så lite ändringar i den "riktiga" DOM som möjligt. ReactComponent konverteras till ReactElement. Nu kan ReactElement infogas i den virtuella DOMen, jämföras och uppdateras snabbt och enkelt. Det görs snabbare än vad det skulle vara i den "vanliga" DOM.

Ajax är en förkortning som står för Asynchronous JavaScript and XML.

Det är inget programmeringsspråk utan AJAX är en teknik för åtkomst till webbservrar från en webbsida. I grunden bygger Ajax på en teknik som kallas XMLHttpRequest. Bakom det kryptiska ordet ligger möjligheten att anropa en server via http-teknik med hjälp av JavaScript.

Huvudsyftet med Ajax är att förbättra hastigheten, prestandan och användbarheten hos en webbapplikation.

Hypertext Transfer Protocol, eller HTTP, är ett applikationsprotokoll som har varit de facto-standard för kommunikation på internet sedan dess uppfinning 1989. Från och med lanseringen av HTTP 1.1 1997 till nyligen har det gjorts få revisioner av protokollet. Men 2015 kom en nyimagerad version som heter HTTP 2.0 i bruk, som erbjöd flera metoder för att minska latensen, särskilt när det gäller mobila plattformar och serverintensiv grafik och videor. HTTP 2.0 har sedan dess blivit allt populärare, och vissa uppskattningar tyder på att ungefär en tredjedel av alla webbplatser i världen stöder det.