

IP2 Redogörelsen

Vanessa Martinez FE20

1.5. Olika css-ramverk

CSS-ramar är väldigt populära idag och det är svårt att föreställa sig en webbplats utan att använda de. Det här är 2 CSS-ramverk som är väldigt populära och proffsiga idag, Bootstrap och Foundation. De är baserade på CSS-preprocessorer. CSS-preprocessorer är ett program som låter dig generera CSS från preprocessors egen unika syntax. Less används på Bootstrap och SCSS på Foundation . Med dessa får du responsiva lösningar direkt, från start. Nackdelen är att när man använder ett färdigt ramverk så ser sajten ut som tusentals andra sajter. Jag använde mig inte utav de, känner mig inte så bekväm med det.

1.6. SASS eller LESS

Sass och Less är en CSS preprocessor, de har mycket gemensamt. Less syntaxen är väldigt lik vanlig CSS. Det är populärt i Bootstrap. Sass är populärt val i webbutvecklingskretsar, framförallt i kombination med Compass. Sass har mer möjligheter till programmering. Skillnaderna är som mest i detaljerna. Man förenklar arbetet med att använda någon av de, den man känner sig bekvämast med. I en undersökning visade sig att fler föredrog att använda Sass. Här finns 3 fördelar med de:

1. Sparar tid
2. Mer organiserad och enkel att installera
3. Renare kod med återanvändbara delar och variabler.

Jag använde mig bara av vanlig CSS.

1.1 JavaScript, grundläggande syntax:

Som alla andra språk, vanliga språk och programmeringsspråk så har JavaScript en bestämd syntax, regler och konventioner för hur man formulerar sig och skriver för att det ska bli begripligt. Här visar jag 2 korta exempel på hur en JavaScript grundläggande syntax fungerar och innebörden med den.

Det här är Booleans, det använder man för att se ifall det är true eller false.

I det här fallet blir svaret true, för att email inkluderade "@". Om man använder "let" så kan man ändra värdet på variabeln, men om man använder const så kan man inte ändra värdet på variabeln.

```
let email = 'vanessa@live.se';
let result = email.includes('@');
console.log(result);
```

Här använde jag mig utav if och const. Age är 25, om Age (25) är större än 20 så visa mig texten "you are over 20", eftersom age är större än 20 så visar den texten.

```
const age = 25;
if(age > 20){
  console.log('you are over 20');
}
```

I min hemsida använde jag mig mycket utav const.

1.2 Javascriptsversioner och dess kompatibilitet(likhet):

Det finns en del JavaScript versioner och det har utvecklats en hel del, det blir bara bättre. Brendan Eich uppfann den år 1995 och blev en ECMA-standard 1997. ECMAScript är den officiella namnet på språket men förkortas till tex ES1, ES2 osv. ECMAScript stöds fullt ut i alla moderna webbläsare. Den jag använder till mitt projekt idag är ES6 och det är den senaste versionen som vissa utvecklare kallar "Modern JavaScript" pga alla tillägg som tex nya nyckelord som let och const, ny funktionssyntax med hjälp av pilfunktioner, skapande av klasser m.m. Men man ska inte bortse från äldre kod dvs äldre versioner av JavaScript, det finns fortfarande många projekt som byggs och underhålls med äldre kod. Så det är bra att bekanta sig med JavaScript-syntaxen före ES6.

1.3. Hur JavaScript körs i webbläsaren:

Kanske många har hört talas om V8 motorn som ett koncept, och de flesta vet att JavaScript har en enda Call Stack, det innebär att den bara kan utföra en instruktion i taget. V8 motorn används tex i Chrome och Node.js. V8 består av två huvudkomponenter som är Memory Heap och Call Stack. Call Stack är en datastruktur som i princip registrerar var i programmet vi är. Om vi går in i en funktion lägger vi den på toppen av stacken. Om vi återvänder från en funktion dyker vi upp på toppen av stacken. Det är allt stacken kan göra. **"Blowing the stack"** det kallas när man når den maximala storleken för Call Stack. Det kan hända ganska enkelt, om man inte testat sin kod ofta. Problemet är att medan Call Stack har funktioner att utföra, kan webbläsaren faktiskt inte göra något annat, det blockeras. Det betyder att webbläsaren inte kan svara, den kan inte köra någon annan kod, den är bara fast. När din webbläsare börjar bearbeta så många uppgifter i Call Stack, kan den sluta fungera, och de flesta webbläsare

vidtar åtgärder genom att ta upp ett fel och fråga dig om du vill avsluta webbsidan. Det finns en lösning ifall man vill köra tung kod utan att få massa problem på vägen, det är asynchronous callbacks. Jag använde API mha AJAX (fetch)

1.4. JavaScript-bibliotek:

Ett JavaScript-bibliotek är som en samling av verktyg (funktioner, klasser, metoder, m.m.) som löser olika typer av problem det används för att underlätta och snabba upp utvecklingen. Biblioteken är helt enkelt utvecklade för att samla den vanligaste funktionaliteten på ett ställe och för att det ska räcka med att anropa en enda funktion för att köra en annars komplex funktionalitet. jQuery är ett bibliotek som många använder, den låter dig göra dynamiska saker på statiska webbsidor lättare än andra metoder och kräver inte programmering på serversidan för att uppnå resultaten. Prototype är en annan JS-bibliotek som kan ses som en konkurrent till jQuery. Jag använde mig utav react router.

1.5. JavaScript-ramverk:

Angular, vue och React är kända JavaScript-ramverk. Det sägs dock att React är mer ett bibliotek än ett JavaScript-ramverk. Fördelen med JavaScript ramverk är att projekt som tog månader att koda kan idag uppnås mycket snabbare, de flesta ramar är öppen källkod och gratis. En nackdel med ramverk är att man inte har riktigt koll på all kod och det gör svårare att felsöka, men de flesta kända ramverk har manualer och forum så man hittar det man söker ganska snabbt. Jag använde mig utav React.

1.6. DOM:

Document Object Model, kallas DOM. Det vanligaste och tydligaste sättet att representera DOM på, är att visa det som ett träd. Trädet har ett rotelement och alla objekt i modellen kallas för noder. De vanligaste nodtyperna är: Document, Element, Attribut och Text. Noderna är relaterade till varandra som en familj. En nod kan ha en förälder, syskon och barn. DOM gör så att alla taggar i ett HTML dokument kan behandlas som fristående objekt. När en klient (webbläsare) laddar ett (X)HTML-dokument så skapar den en så kallad dokumentmodell, elementträd eller nod träd som det heter med DOM-termer. Varje nod i trädet motsvaras av en tagg, attribut i tagg eller text inom tagg. DOM-gränssnittet (API:t) innehåller ett antal metoder för att manipulera noderna (taggarna, attributen och texten) i ett dokumentets nod träd, d.v.s. ändra existerande noder samt lägga till och ta bort noder (element). Jag använde mig utav den virtuella DOM, det är som en "kopia" av den riktiga dokumentobjektmodellen, fast den ritas inte ut på sidan så fort något ändras.

1.7. AJAX:

Ajax är en förkortning som står för Asynchronous JavaScript and XML. Ajax är en samling tekniker för att bygga interaktiva webbsidor, inte ett programmeringsspråk som många tror. Kortfattat så kan man säga att idén är att göra webbsidor som kan hämta in mer information löpande från servern utan att sidan behöver laddas om. Nu finns stora möjligheter att låna skript och använda färdiga bibliotek för att göra det enklare för sig. Jag hämtade data från ett API med hjälp av AJAX och använde fetch för att hämta information om väder.

1.8. HTTP 1.1 / 2.0:

HTTP står för HyperText Transfer Protocol, metoden som används när webbläsaren skickar och tar emot information från nätet. I den här processen skickar en klient en textbaserad begäran till en server genom att anropa en metod som GET eller POST. Som svar skickar servern en resurs som en HTML-sida tillbaka till klienten. GET används när vi navigerar, webbläsaren skickar en förfrågan till URL och webbservern skickar tillbaka ett svar. Servers response innehåller en tresiffrig statuskod som är 200 vilket betyder att det är "OK", förfrågan lyckades och servern skickar ett svar, den andra statuskoden är 404 "Not found" det visas när inte servern hittar [URL:en](#). POST används när vi fyller i ett formulär och klickar på knappen "submit". En av nackdelarna med HTTP 1.1 var att HTTP1.1 var begränsad till att endast behandla en utestående begäran per TCP-anslutning, vilket tvingade webbläsare att använda flera TCP-anslutningar för att behandla flera förfrågningar samtidigt. En av fördelarna med HTTP 2.0 är att en enda TCP-anslutning används för att säkerställa ett effektivt nätverksresursutnyttjande trots sändning av flera dataströmmar. HTTP1.1 och HTTP / 2.0 skiljer sig på många sätt, med vissa funktioner som ger högre kontrollnivåer som kan användas för att bättre optimera webbapplikationsprestanda och andra funktioner som helt enkelt förbättras. Jag använde http genom att länka min github och linkedin.