**Bilder:**

JPEG: Naturliga bildtyper, som fotografering, har smidiga färgförändringar vilket innebär att JPEG-formatet också fungerar bra för bilder som innehåller gradienter och varierande toner och färger. JPEG har dock data loss.

GIF: Denna komprimeringsmetod gör detta bildformat idealiskt för linjekonst, logotyper eller andra enkla bilder utan gradienter eller varierande färg.

GIF kan använda transparans och animering, pixlarna är antingen 100% transparant eller 100% ogenomskinling.

PNG: PNG-format tenderar att resultera i större bildstorlekar än JPEG när den används för fotografiska bilder. När det används för bilder av linjekonstyp, resulterar PNG i mycket mindre filstorlekar än JPEG och nästan alltid mindre storlekar än GIF. Pixlar i en PNG-bild kan vara 50% transparanta, vilket möjligör bättre blandning och mjukare grafik är tex en GIF-bild.

Det finns:

PNG8: Typ samma som GIF (färgjämförning.

PNG24: Mycket bredare färg möjligheter.

Jag har valt att göra om Hackett-logotypen till PNG8 med bara 16 färger eftersom den inte använder sig av så många färger/nyanser.

Även valt att göra om main.png till jpg samma kvalité och 2.7mb mindre.

Även valt att göra om newletter\_bg till jpg och alla andra modell bilder till jpg. Minskat storleken med ca 75% och kvalitén kvarstår.

**Css selektorer:**

Jag föredrar att använda mig av klasser i alla sammanhang som går, men det finns fördelar med IDs tex att det är smidigt att använda med javascript. Det är det snabbase vägen en webbläsare kan selecta ett element och skapa tydlighet i dokumentet. Men det negativa är:

1. Elementet kan inte användas flera gånger på samma sida.
2. Det är början på en spiral mot specifisitet
3. Eventuella prestationsvinster som gynnas genom att använda id’t, försvinner genom att lägga till någon annan selektor till vänster om id’t

**HTTP-Requests:**

Network delay är problemet med http, protokollet är väldigt känsligt för network delay eftersom det bara kan ha 6 öppna tcp connections per host, pipelining fungerar inte, (Klienten frågar efter många saker med request). Med http2 så räcker det med en connection som skickar allt genom multiplexing. Http2 är ett binärt protkoll medans Http1 är ett text protokoll.

**Preprocessors:**

Fördelar:

1. Modularisering
2. Återanvända kod, mixins osv
3. Snabbare utveckling, nested slectors

Nackdelar:

1. Ett extra lager av komplexitet
2. Behövs compilas
3. Svårt att live edita en sida för lite snabba ändringar
4. Hela teamet behöver komma överens om det valda språket

Bra ilustration: http://http.tunetheweb.com/performance-test/