



Mittuniversitetet

MID SWEDEN UNIVERSITY

Webbprogrammering med HTML5, CSS3 och JavaScript

Laboration 3 – Grunder i JavaScript

Innehållsförteckning

1 Laborationsbeskrivning.....	3
2 Redovisning.....	3
3 Laborationskrav.....	4
3.1 Övriga krav.....	5
3.2 Överkurs.....	5

1 Laborationsbeskrivning

I denna laboration skall du tillföra funktionalitet med JavaScript till ditt flygbolag. Utgå från din lösning för laboration 2 och isolera arbetet till en ny förgrening med namnet **laboration_3**!

*Notera att du skall förhålla dig till exakt angiven namnkonvention för dessa förgreningar, d.v.s. **laboration** med enbart gemener tillsammans med understreck för att sammanbinda löpnummer. Detta underlättar lärarens administration av inlämningsuppgifter!*

2 Redovisning

Använd **.gitignore** för att utesluta onödiga filer och se till att lokala ändringar är synkroniserat med **remote origin** innan redovisning. När du är färdig med arbetet så utför du den formella inlämningen i Moodle som då notifierar oss att arbetet skall utvärderas. Observera att du inte skall bifoga några filer vid inlämningen, där repot skall föreligga gemensam datakälla!

Alla dina **HTML**-sidor skall innehålla inledande kommentarer som redogör för ditt *namn*, *studentid* samt *epostadress* (se exempel nedan). Du får gärna även kommentera den löpande koden om du vill!

```
<!DOCTYPE html>

<!--
Anders Student
anst1601
anst1601@student.miun.se
-->

<html>
  <head>
    <meta charset="UTF-8">
    ...
```

Unicode skall användas för **character encoding**, så se till att den textredigerare som du använder har stöd för detta!

Lösningen skall sedan kunna passera validering i enlighet med specifikationen för **HTML5**. För detta ändamål används **W3C's Markup Validation Service** ¹, och du bör själv bekräfta att detta uppfylls innan du lämnar in. Ett tips är att då välja **Validate by File Upload** så att rätt *encoding* används! På samma sätt skall din stilmall kunna valideras, för detta används **W3C's CSS Validation Service** ²!

1 <https://validator.w3.org/>

2 https://jigsaw.w3.org/css-validator/#validate_by_upload

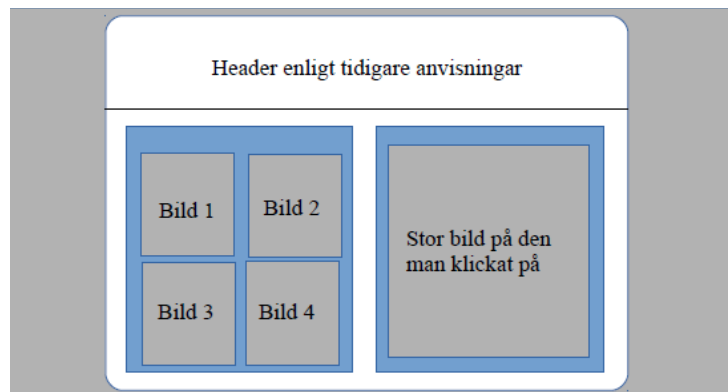
3 Laborationskrav

Du skall skapa två HTML-sidor som hanterar bilder;

ourfleet.html *Flygplanen i bolagets flotta!*

employees.html *Bolagets anställda personal!*

Dessa skall du implementera likt fotoalbum. När besökaren kommer till respektive fotoalbum skall han se klickbara småbilder (thumbnails) i webbläsarens vänstra sida av fönstret, minst 4st skall finnas. När man klickar på dessa mindre bilder skall en större variant av samma bild presenteras i den högra delen av fönstret!



Tänk efter när ni planerar vad som kommer behövas, vilket utrymme som krävs osv. Ni skall använda HTML-element för att skapa layouten. Ni får alltså **INTE** använda er av *layers*, nya fönster eller *frames*!

Bildhanteringen skall fungera så att ni har två varianter av varje bild (två bildfiler). En miniatyr av bilden som skall användas som thumbnail samt en större bild som laddas då man klickar på denna mindre version. Storlekarna skall anpassas till sina respektive områden, så du måste bestämma en höjd och bredd för båda versionerna som sedan används konsekvent!

Om ni vill köra JS-kod då sidan har laddats så skall ni följa exempel **9.5.2 (s.319-323)** i kursboken. Ni skall registrera en funktion som körs på window eventet load. Denna registrering bör ske sist i er JS-fil. Inga `<body onLoad= "...">`, likaså får inga `onClick` förekomma i andra HTML-element!

Eftersom ni enbart får använda en JS-fil som skall fungera på alla sidor så är ett tips att detektera på vilken sida som JS-koden körs och t.ex. anpassa en start-funktion utifrån vad man vill göra på den sidan. Att detektera vilken sida man är på kan man göra med hjälp av `windows.location.pathname`, se följande artiklar för tips; [artikel1](#)³, [artikel2](#)⁴.

³ <https://css-tricks.com/snippets/javascript/get-url-and-url-parts-in-javascript/>

⁴ <https://developer.mozilla.org/en-US/docs/Web/API/Window/location>

3.1 Övriga krav

Det är viktigt att du har en tydlig uppdelning av koden;

HTML Ingen style-attribut i elementen och ingen JavaScript skall förekomma!

CSS All formatering av innehållet skall isoleras till din stilmall, ingen CSS-kod får finnas i HTML-dokumentet!

JS All JavaScriptkod skall skrivas i en separat JS-fil. Ingen förekomst av detta skall förekomma i HTML-dokumentet!

- Alla filer + filändelser skall vara skrivna med gemener!
- All kod skall vara layoutmässigt genomarbetad. Layoutmässigt genomarbetad kod innebär att indragningar, talande variabelnamn, extra radbrytningar och kommentarer används för att göra koden lättläst!
- Innehållet av **README.md** skall återge laborationsarbetets genomförande och struktureras i enlighet med de värden som specificerats i studiehandledningens kapitel 4!

Arbetet för laborationen skall utföras enskilt och par-/grupparbete är inte tillåtet. Däremot uppmuntras utbyte av erfarenheter på de plattformar för kommunikation som tillhör kursen!

3.2 Överkurs

Om man vill så kan man förladda alla stora bilder i en array som laddas när man öppnar sidan. Då sker hämtningen av bilden vid sidladdning och finns i webbläsaren när man sedan vill använda den. Detta är speciellt smidigt då användaren inte har en pålitlig uppkoppling, då detta kan förbättra användarupplevelsen på bekostnad av lite datatrafik. Studera gärna följande kodexempel;

```
var images = new Array("1s.jpg", "2s.jpg", "3s.jpg", "4s.jpg"); // Bildnamnen
var img = new Array(); // definition av en tom array

function init() { // Anropas av window eventet load
  for(var i = 0; i < images.length; i++) { // loopa tills alla bilder laddats
    img[i] = new Image; // allokerar utrymme för ett objekt av typen image
    img[i].src = images[i]; // Placera bilden i arrayen
  }
}

window.addEventListener("load", init, false );
```

Du måste komplettera denna kod med en funktion som skiftar bilder!