



Mittuniversitetet

MID SWEDEN UNIVERSITY

Webbprogrammering med HTML5, CSS3 och JavaScript

Laboration 1 – Grunder i HTML5

Innehållsförteckning

1 Laborationsbeskrivning.....	3
2 Redovisning.....	3
3 Laborationskrav.....	4
4 Arbeta med Förgreningar.....	5
4.1 Skapa Förgrening Lokalt.....	6
4.2 Skapa Förgrening via Bitbucket.....	7

1 Laborationsbeskrivning

Du skall skapa en webbplats för ett flygbolag. Webbplatsen kommer senare att växa fram under kursens fyra laborationer, där mer funktionalitet tillförs. I denna första laboration skall du bara göra en entrésida där du i all enkelhet välkomnar besökaren till ditt flygbolag!

Arbetet, tillsammans med alla nödvändiga projektfiler, skall versionshanteras inom tilldelat studentrepo på kursens **Bitbucket Team**:

`studid_laborations_vt20.git`

Skapa en förgrening med namnet **laboration_1** inom repot och isolera arbetet till denna, då det ger en tydlig separation från övriga förgreningar. Se **kapitel 4** i detta dokument för detaljer kring hur nya förgreningar kan skapas.

*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">
    ...
```

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!

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**, 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!

3 Laborationskrav

Ditt projekt skall följa denna filstruktur;

root/

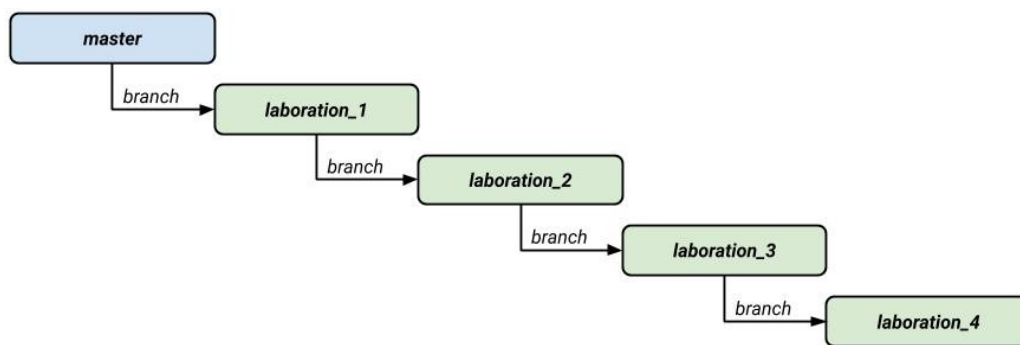
```
img/  
...  
.gitignore  
index.html  
README.md
```

1. Din **HTML5**-fil skall heta **index.html** och inget annat!
2. Bredden på din sida skall vara max **960px** och allt innehåll skall vara centrerat på skärmen horisontellt. Vertikalt skall informationen börja i överkant!
3. I bakgrunden på din sida ska du ha en bild. Den och alla andra bilder skall ligga i katalogen **img/**.
4. Sidan skall innehålla 3 st delar; **huvudsektion** (*header*), **innehållssektion** samt **fotsektion** (*footer*). Du skall använda **HTML5** taggar som finns för att skapa denna struktur och skall INTE använda taggen **<div>** för att strukturera upp innehållet på sidan!
5. I **huvudsektionen** skall det finnas en logotyp för ditt flygbolag. Den bilden kan se ut precis hur som helst. Vidare skall det här finnas navigationslänkar till eventuella sidor som tillförs i framtiden!
6. I **innehållssektionen** skall det finnas en text som välkomnar besökaren till din webbsida!
7. I **fotsektionen** skall det finnas information om företaget samt en **copyright** symbol (©) med tillhörande text. Denna del skall ha mindre text än övriga sektioner och det skall finnas en linje som avskiljer denna sektion från *innehållssektionen*!
8. All **CSS**-kod skall skrivas **embedded** i ditt **HTML5**-dokument, d.v.s. det får till denna uppgift inte användas separata filer för *stilmallar* (**stylesheet**). Vidare skall denna information deklarerars i ett **style block** i dokumentets **HEAD**, det får alltså inte förekomma **style-attribut** i sektionernas taggar!
9. Alla filer + filändelser skall vara skrivna med gemener!
10. All kod skall vara layoutmässigt genomarbetad. Layoutmässigt genomarbetad kod innebär att indragningar, talande variabelnamn, extra radbrytningar och kommentarer används för att göra koden lättläst!
11. Innehållet av **README.md** skall återge laborationsarbetets genomförande och struktureras i enlighet med de värden som specificerats i studiehandledningens kapitel 4!

4 Arbeta med Förgreningar

Ni har i tidigare kurser introducerats till versionshantering och bekantat er med några grundläggande Git-kommandon. Då utfördes det löpande arbetet direkt mot huvudgrenen (**master**), något som egentligen inte är särskilt lämpligt. Istället bör utvecklingsarbetet utföras i dedikerade förgreningar och enbart sammanföras med huvudgrenen först när arbetet är färdigt. Genom att isolera utvecklingen på detta sätt garanteras att huvudgrenen förblir opåverkad av eventuella problem som uppstår under arbetet och du har hela tiden tryggheten att återgå till **master** för en fungerande version av projektet.

Konceptet av förgreningar är mycket centralt för Git och det finns en stor mängd olika förgreningsstrategier som kan tillämpas. Under denna kurs kommer vi dock inte att fördjupa oss inom dessa strategier och inte heller kommer vi använda arbetsförgreningar såsom de egentligen är ämnade. Alla laborationer i kursen skall utgöra en egen dedikerad förgrening, men inte vid något tillfälle skall dessa sammanföras med **master**. Med andra ord tar vi ett litet steg framåt vad gäller versionshantering, men undviker medvetet den komplexitet som föreligger sammanföringar där exempelvis s.k. **merge conflicts** ofta uppstår och behöver hanteras.



Förutom detta så lämpas denna struktur mycket väl för kursens upplägg, där varje laboration bygger vidare på föregående laboration. När du är färdig med en laboration skapar du alltså en ny förgrening med den tidigare labben som utgångspunkt. Denna separation isolerar således arbetet på ett sätt som gör det smidigt att påbörja nästa labb utan att behöva invänta återkoppling på den tidigare. Eventuella kompletteringar utförs direkt på aktuell förgrening och påverkar alltså inte övriga labbarbeten.

För att skapa nya förgreningar så finns det lite olika tillvägagångssätt varav vi nu skall visa två möjligheter, ett lokalt samt ett med fjärran datakälla som utgångspunkt.

4.1 Skapa Förgrening Lokalt

För att skapa en ny förgrening lokalt är det viktigt att du befinner dig i / har checkat ut den förgrening som skall kopieras. När du har klonat ditt studentrepo så finns enbart grenen master och du kan verifiera detta genom följande kommando:

```
$ git branch
* master
```

Asterisk (*) anger vilken gren som är aktiv. Nu är master tämligen tom på innehåll och kommer så vara alltigenom kursen, eftersom vi tillför innehåll till dedikerade arbetsförgreningar. För att kopiera tillståndet av master till en ny förgrening använder vi kommandot...

```
$ git checkout -b laboration_1
```

... vilket skapar en ny förgrening vid namn **laboration_1** och som automatiskt blir aktiv (**checked out**). Flaggan **-b** anger här att vi skall skapa en ny namngiven branch. För att skifta mellan **master** och **laboration_1** behöver vi enbart använda kommandot **git checkout name_of_branch**. Verifiera att vi nu har två förgreningar:

```
$ git branch
* laboration_1
  master
```

Du arbetar sedan som vanligt och utför **commits** direkt till **laboration_1**. När du första gången skall synkronisera denna förgrening med **remote origin** måste du ange en **target upstream** och för att undvika förvirring skall denna namnges detsamma som den lokala förgreningen:

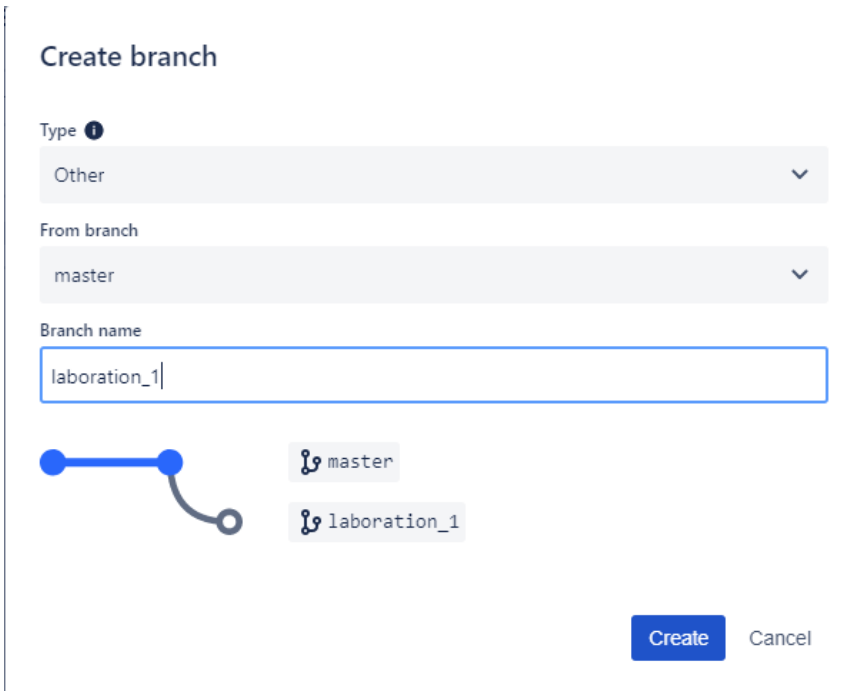
```
$ git push -u origin laboration_1
```

4.2 Skapa Förgrening via Bitbucket

Förgreningar kan även skapas direkt på **Bitbucket** genom dess verktyg, som du finner via

Branches → Create Branch

I den ruta som nu dyker upp anger du vilken förgrening som skall kopieras samt namnet på den nya förgreningen. Typ av förgrening kan du lämna orörd då det inte är något som berör denna kurs.



Efter det att förgreningen är skapad så dyker den inte automatiskt upp när du som tidigare listar förgreningar via kommandot **git branch**. Anledningen för detta är att grenen ännu inte finns i det lokala repot, vilket kan åtgärdas med en vanlig **git pull**. Du kan även lista samtliga förgreningar, både för lokalt samt fjärran, genom att lägga till flaggan **-a**:

```
$ git branch -a
  laboration_1
* master
remotes/origin/HEAD → origin master
remotes/origin/laboration_1
remotes/origin/master
```

Värt att nämna är att **remotes/origin/HEAD** pekar på repots förvalda (default) förgrening!