

GIK339 – LABB 2

SKAPA OCH ÄNDRA HTML VIA JAVASCRIPT

För 4hp i kursen *Dynamiska Webbapplikationer* ska tre labbar och ett onlineprov utföras. Detta är den andra labben. Den utförs i labbgrupper, enligt de som finns i kursrummet (markerade *Labbgrupp XX*).

Läs igenom instruktioner för förberedelser, uppgifter och inlämning noggrant och tveka inte att höra av er till mie@du.se om det är några frågor.

Lycka till!

Innehåll

Förberedelser	3
Uppgifter	4
Notering rörande generativ AI (GAI).....	4
Uppgift 1 Skapa HTML-fil.....	4
Uppgift 2 Skapa HTML-kod (valfri styling).....	4
Uppgift 3 - Skapa och länka script-fil.....	6
Uppgift 4 - Skapa variabler	6
Uppgift 5 - Skapa en fördefinierad funktion	7
Uppgift 6 Koppla eventlyssnare.....	7
Inlämning	9
Kod	9
Loggar	9
Video	9

Förberedelser

1. Skapa ett lokalt repository på någon av er dator, exempelvis enligt instruktioner i [*Git, GitHub, & GitHub Desktop for beginners*](#).
2. Döp det till **gik339-labb2-[gruppnummer]**.
 - a. **Notera:** Ta inte med punkten ovan. Det blir fel på repositoryt om den slutar med en punkt.
 - b. Om ni följer videon ovan kommer detta skapa en mapp med samma namn på en den position som står angivet i fältet **Local path** i GitHub for Desktop.
3. Öppna mappen för repositoryt i VS Code. Sökvägen till mappen finns angivet i fältet **Local path** i GitHub for Desktop.
4. Utför labben enligt instruktioner
 - a. **OBS!** när ni gjort förändringar och vill [publicera ert repository till GitHub](#), se till att inställningen **Keep this code private** **inte** är markerad.
5. Om gruppen vill arbeta från olika datorer, se till att båda jobbar utifrån samma repository på GitHub.
 - a. Alltså, person 2 ska inte skapa ett nytt repository, utan hämta [det befintliga på GitHub](#) och ladda ner det lokalt till sin dator.
 - b. Det ska alltså bara finnas ett (**1**) repository som ni ska arbeta mot, och vars länk ni ska lämna in enligt avsnittet [Inlämning](#).

Uppgifter

Nedan följer de uppgifter som ni ska ha utfört för att få godkänt på labben. Uppgifterna är *minimum*, vilket innebär att ni får labba runt med egna tillägg om ni vill, så länge *instruktionerna följs*. Instruktionerna i uppgifterna kanske inte alltid är helt logiska, men det finns en poäng med dem. Vid vissa tillfällen ska ni hitta och motivera egna lösningar, vid andra tillfällen ska ni använda er av något specifikt, t.ex. ett specifikt event.

Jag hänvisar ibland till de föreläsningar och lektioner som behandlar ett givet område som ni ska arbeta med. Alla föreläsnings- och lektionsfiler finns [här](#). Ni behöver titta i mapparna för respektive vecka för att hitta det hänvisade materialet. Länkar till material finns också i kursrummet.

[Exempel på hur webbsidan ska fungera när alla uppgifter är slutförda finns i denna video.](#)

Notering rörande generativ AI (GAI)

I denna labb får ni använda er av AI enligt [nivå 3](#). Det kan vara exempelvis för att förstå instruktioner, utveckla en befintlig idé, förbättring av syntax eller organisering av kod.

Ni behöver förstå och kritisera all AI-genererad kod och vara beredd att förklara den om lärare ber er visa att ni förstår koden.

Alla loggar/kommentarer som genererat kod behöver inkluderas vid inlämning. Vidare detaljer rörande hur detta ska redovisas finns i avsnittet [Inlämning](#).

Uppgift 1 Skapa HTML-fil

Skapa filen index.html och sätt upp en grundläggande HTML-struktur.

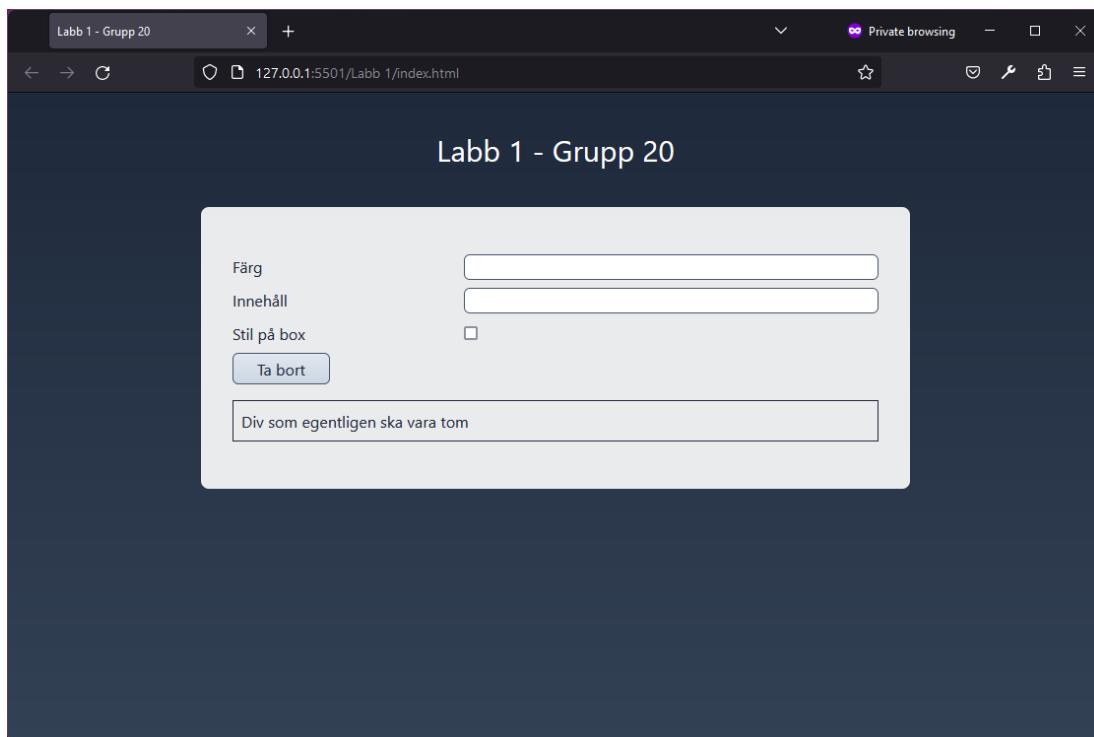


I VS Code kan man skriva `!` och tangenten `tab` i en fil med filändelsen `.html` för att få en grundläggande HTML-struktur.

Uppgift 2 Skapa HTML-kod (valfri styling)

Skapa ett antal HTML-element för vidare användning i labben. Ni får styra dessa element hur du vill. Skriv gärna gruppnummer i exempelvis `title`-elementet eller något annat element på er site.

Det kan se ut ungefär såhär:



Figur 1 - Exempel på utseende på er webbsida.

Detaljer rörande vad som *minst ska* finnas på webbsidan följer:

Tre **input**-element med tillhörande **label**-element

1. **Input-element 1:** Attributet **type** ska vara **"text"**, attributen **name** och **id** ska vara **"color"** och attributet **class** ska vara **"textfield"**.
2. **Input-element 2:** Attributet **type** ska vara **"text"**, attributen **name** och **id** ska var **"content"**, och attributet **class** ska vara **"textfield"**.
3. **Input-element 3:** Attributet **type** ska vara **"checkbox"**, attributen **name** och **id** ska vara **"divStyle"**, attributet **class** ska vara **"checkbox"**.

Ett **button**-element

Ni får bestämma vad det ska stå på knappen, hur den ska se ut och eventuella andra attribut. Se till så att ni senare ska kunna hämta upp knappen via JavaScript, exempelvis genom att ge elementet en klass eller ett id.

Ett **div**-element

Se till så att ni senare ska kunna hämta upp elementet via JavaScript, exempelvis genom att ge elementet en klass eller ett id. Elementet ska inte innehålla någonting (det står dock text i exemplet ovan, men det är bara för att det ska synas att elementet finns).

Uppgift 3 - Skapa och länka script-fil

Se till att ditt HTML-dokument får tillgång till en extern scriptfil. Skapa en scriptfil och länka den i ditt HTML-dokument via ett `script`-element. Se avsnitt **Webbläsaren** i **Föreläsning 1** för olika sätt att läsa in script-filer till HTML-dokumentet.

Resterande uppgifter av labben ska i huvudsak utföras i `script`-filen som skapas i detta steg.

Uppgift 4 - Skapa variabler

I script-filen som skapades i föregående steg, ska ett antal variabler skapas för att hålla olika HTMLElement-objekt.

1. Variabel 1: Checkboxen

- Hämta checkboxen (`<input type="checkbox">`) på.
- Använd **valfri DOM-funktion**¹ för att hämta upp checkboxen.
 - Använd helst en **DOM-funktion** som ni inte använt i vid hämtning av något annat HTML-element.
- Lagra den i en variabel av lämplig typ (`var`, `let` eller `const`). Ge variabeln lämpligt namn.

2. Variabel 2: Alla textfält

- Hämta alla textfält (`<input type="text">`) som **en enda samling** av element genom att söka fram dem via deras klassnamn (`textfield`).
- Använd **valfri DOM-funktion** för att hämta upp samlingen av textfält.
 - Använd helst en **DOM-funktion** som ni inte använt i vid hämtning av något annat HTML-element.
- Lagra samlingen i en variabel av lämplig typ (`var`, `let` eller `const`). Ge variabeln valfritt namn.

3. Variabel 3: Knappen

- Hämta knappen (`<button>`) med hjälp av **valfri DOM-funktion**.
 - Använd helst en **DOM-funktion** som ni inte använt i vid hämtning av något annat HTML-element.
- Lagra knappen i en variabel av lämplig typ (`var`, `let` eller `const`). Ge variabeln valfritt namn.

4. Variabel 4: Div-elementet

- Hämta div-elementet (`<div>`) med hjälp av **valfri DOM-funktion**.
 - Använd helst en **DOM-funktion** som ni inte använt i vid hämtning av något annat HTML-element.
- Lagra elementet i en variabel av lämplig typ (`var`, `let` eller `const`). Ge variabeln valfritt namn.

¹ Med DOM-funktion här, och vid efterföljande omnämnen, avses de olika funktioner som går igenom i **Föreläsning 2**, exempelvis `getElementById()`.



Uppgift 5 - Skapa en fördefinierad funktion

Funktionen ska skapas som en **funktionsdeklaration**. Funktionen kommer på sikt att användas som **eventlyssnare** till inputfälten. Se avsnitt **Funktioner** i **Föreläsning 1** för mer om funktioner.

Döp funktionen till vad ni vill, helst något lämpligt för syftet.

Följande ska hända:

- ✓ När funktionen (**eventlyssnaren**) triggas ska **avsändaren** (**target**) **skrivas ut till konsolen**.
- ✓ Om avsändaren (**target**) är inputfältet med **id "content"**, ska dess värde **skrivas ut till** **div-elementet**.

Detaljer:

- **Ta emot** eventobjektet som skickas i bakgrunden när eventlyssnaren triggas (lämpligen döpt till **e**).
- Använda eventobjektet (**e**) för att
 - **Skriv ut till konsolen** vilket inputfält som genererade eventet (**target**).
 - **Ta reda på** **inputfältets** (**target**) **name-attribut**. Innehållet i **name-attributet** kan lämpligen lagras i en variabel.
- **Om** **name-attributet** är "**content**", ska **innehållet** (**value**) i **inputfältet** **skrivas ut till** **div-elementet**, som ni hämtade och lagrade i en variabel i [Uppgift 4, steg 4](#).
 - Använd exempelvis egenskapen **innerHTML** för att ge **div-elementet** ett innehåll.

Uppgift 6 Koppla eventlyssnare

Koppla eventlyssnare till textfälten, checkboxen och knappen. Se avsnitt **Event - Eventlyssnare** i **Föreläsning 3** för detaljer kring att koppla eventlyssnare. Avsnittet **Funktioner** i **Föreläsning 1** kan också vara till hjälp.

Eventlyssnare till checkboxen

Koppla en **anonym funktion** som eventlyssnare till **checkboxen**, som ni hämtade och lagrade i en variabel i [Uppgift 4 steg 1](#).

Det som ska hända är följande:

- ✓ Vid **förändring** i **checkboxen** ska **div-elementet** **ändra bakgrundsfärg** till den **färg som står skriven** i **input-fältet** med **namn/id "color"**.

Detaljer:

- Eventlyssnaren ska köras vid eventet "**change**".

- **div**-elementet som avses är det elementet som ni hämtade och lagrade i en variabel enligt [Uppgift 4, steg 4](#).
- Hämta färgen från **input**-fältets **value**-egenskap.
 - **Notera:** **input**-fältet finns redan lagrat i en samling (hämtad och lagrad i en variabel enligt [Uppgift 4, steg 2](#)), men där finns **samtliga input**-fält. Ni ska bara använda ett av dem nu. Välj själva hur ni åstadkommer detta.
- Använd valfritt sätt för att sätta den upphämtade färgen på **div**-elementet (se avsnitt [Påverka HTML i Föreläsning 2](#)).
 - **Notera:** det spelar ingen roll om checkboxen markeras eller avmarkeras. Bakgrundsfärgen ska ändras oavsett läge hos checkboxen. Ni kan dock för all del testa att ha olika funktionalitet beroende på om checkboxen blir markerad eller avmarkerad.

Eventlyssnare till textfälten

Koppla en den **tidigare skapade funktionen** (från [Uppgift 5](#)), som eventlyssnare till **varje textfält**.

- Eventlyssnaren ska köras när **någon skriver i fälten** eller **lämnar fälten**. Välj själv lämpligt event att lyssna efter.

Eventlyssnare till knappen

Koppla en **anonym funktion** till **knappen** (hämtad och lagrad i en variabel enligt [Uppgift 4, steg 3](#)).

Det som ska hända är följande:

 Vid **klick på knappen** ska **div**-elementet **försvinna**.

Detaljer:

- Eventlyssnaren ska köras vid eventet "**click**".
- **div**-elementet som avses är det **element** som ni hämtade och lagrat i en variabel enligt [Uppgift 4, steg 4](#).
- Hela **div**-elementet ska tas bort från DOM-trädet.
 - Det går att utföra detta på flera sätt, t.ex. med metoden **remove()**.
 - **Notera:** om ni vill testa flera gånger får ni ladda om sidan så att **div**-elementet kommer tillbaka.

Inlämning

Nedanstående ska lämnas till uppgiften märkt Labb 2 i kursrummet. Vidare instruktion och datumangivelser finns i kursrummet.

Kod

- Lämna in länk till GitHub-repository som skapades och lades upp på GitHub under [förberedelserna](#).
 - Länken ska se ut något i stil med: [https://github.com/\[ditt-github-användarnamn\]/gik339-labb2-\[gruppnummer\]](https://github.com/[ditt-github-användarnamn]/gik339-labb2-[gruppnummer]).
- **Observera!** Lämna **inte** in en zip:ad mapp med er kod!

Loggar

Om generativ AI har använts ska dokumentation rörande användning inkluderas i inlämningen. Detta kan röra sig om länkar till loggar eller dokument som visar exempelvis prompter och svar från de verktyg ni använt (t.ex. **ChatGPT** eller **GitHub Copilot**)

Video

Gör en inspelning på **max 5 minuter**. Ni ska båda ha kameror på och namn synliga så att jag kan se vem som pratar (kan exempelvis uppnås genom att använda Zoom).^[2]

I videon ska ni:

- Demonstrera att din webbplats fungerar enligt ovanstående uppgiftsbeskrivningar. Läs noggrant hur det är meningen att de olika eventlyssnarna ska fungera.
 - Ha konsolen (fliken "**Console/Konsol**" i DevTools) öppet i din webbläsare så att det syns att interaktion med webbsidan ger önskad output.
 - Visa att **div**-elementet faktiskt tas bort ur DOM-trädet vid klick på knappen (via fliken "**Elements/Element**" i DevTools).
 - Båda ska vara delaktiga i att visa och förklara webbsidan och output i DevTools.
- Välj minst en varsin av nedanstående uppgifter och förklara rad för rad hur koden fungerar:
 - Koden som skrevs i [Uppgift 4](#)
 - Resonera kring för-/ nackdelar med de olika DOM-funktioner som användes för att hämta elementen.
 - Koden som skrevs i [Uppgift 5](#)
 - Koden som skrevs i [Uppgift 6](#)
- Om ni använt generativ AI ska också peka på var det använts och förklara på vilket sätt den förbättrar er kod.

² Notera att om du gör labben ensam behöver du redovisa samtliga delar själv.