



Mittuniversitetet

MID SWEDEN UNIVERSITY

Webbprogrammering med HTML5, CSS3 och JavaScript

Projektuppgift

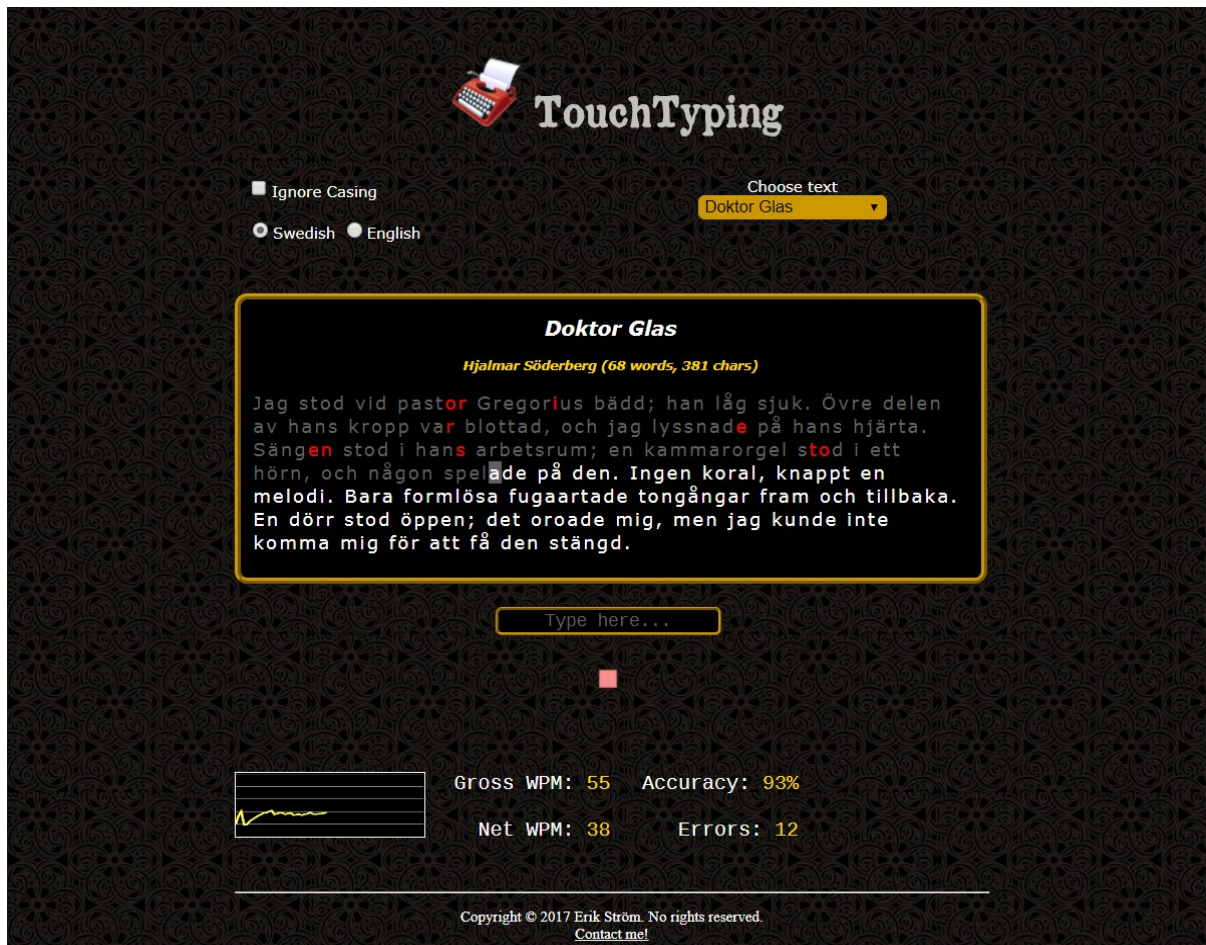
Uppgiften ämnar ge studenten möjlighet att tillämpa kursens innehåll för ett större projektarbete där kunskaper gällande HTML5, CSS3 samt JavaScript skall användas för att konstruera en webbapplikation!

Innehållsförteckning

1 Projektbeskrivning.....	3
2 Redovisning.....	4
3 Projektkrav.....	5
3.1 Basimplementation.....	6
3.1.1 <header> & <footer>.....	7
3.1.2 <main> : Inställningar & Textväljare.....	7
3.1.3 <main> : Textdetaljer.....	8
3.1.4 <main> : Inmatning & Start-/stoppkontroll.....	8
3.1.5 <main> : Statistik.....	9
3.2 Betygskrav.....	10
3.2.1 Betyg D.....	10
3.2.2 Betyg C.....	10
3.2.3 Betyg B.....	10
3.2.4 Betyg A.....	10
4 Utvecklingsmiljö.....	11
4.1 SFTP med WebStorm / PHPStorm.....	11
Referenser.....	13

1 Projektbeskrivning

Som projektuppgift skall du implementera en webbapplikation bestående av en **HTML**-sida, extern stilmall/-ar samt ett stort fokus på **JavaScript**. Målet med uppgiften är att skapa en applikation som mäter användarens skrivhastighet och träffsäkerhet vid skrivning med tangentbordet. Det är alltså inte ett program för att lära ut korrekt fingerplacering (s.k. **Touch Typing**) utan ni skall utgå från att användaren redan kan detta, och istället enbart fokusera på att återge statistik gällande hastighet och felsäkerhet.



En basimplementation för applikationen, motsvarande betyget **E**, skall först och främst konstrueras. Därefter lägger du successivt till ytterligare funktionalitet för att nå de högre betygsskalorna. Viktigt här är att du läser igenom dessa krav och planerar för hur arbetet skall läggas upp. Varje nytt steg upp för betygstrappan förutsätter att kraven för föregående steg är uppfyllt, d.v.s. en implementation motsvarande betyget **A** förutsätter att alla tidigare betygskriterier även de blivit uppfyllda.

2 Redovisning

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

studid_project_vt20.git

Notera att detta repo är separat från det repo som användes för laborationer och till skillnad från den förgreningsredovisning som förelåg laborationsuppgifterna skall **master** i projektrepot utgöra projektets redovisningstillstånd, d.v.s. där eventuella utvecklingsförgreningar har sammanförts med huvudgrenen inför inlämning. Det är naturligtvis tillåtet att ett flertal olika grenar existerar vid redovisningstillfället, men bedömningen kommer enbart behandla **master**.

HTML-sidans 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 så önskar.

```
<!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.

När du är färdig med arbetet skall du genomföra en formell inlämning i Moodle och då använda den inlämningslåda som är avsedd för projektredovisningen. Deadline för inlämningen infaller **7/6 23:55** och sena inlämningar kommer inte att tillåtas. Det kommer finnas ytterligare två tillfällen att redovisa, men med tillhörande betygsavdrag. Detta innebär att betyget **A** enbart är möjligt att uppnå vid första redovisningstillfället, medan **B** blir högsta för andra försöket och **C** blir det högsta möjliga betyget för tredje och sista tillfället. Tidpunkter för dessa senare tillfällen publiceras i Moodle.

OBS! Modifieringar av projektrepot som förekommer efter angiven deadline resulterar i ytterligare betygsavdrag: A → B → C → D → E. Dock ej lägre än E!

¹ <https://validator.w3.org/>

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

3 Projektkrav

Ditt projekt skall följa en filstruktur och formalia enligt följande specifikation:

Filstruktur

```
root/  
[audio/] Grade >= C  
...  
css/  
...  
[font/] Grade >= D  
...  
img/  
...  
js/  
...  
.gitignore  
index.html  
README.md  
[texts.xml] Grade >= B
```

Uppdelning av kod

HTML	<i>Ingen style-attribut i elementen och ingen JavaScript skall förekomma!</i>
CSS	<i>All formatering av innehållet skall isoleras till din stilmall, ingen CSS-kod får finnas i HTML-dokumentet!</i>
JS	<i>All JavaScriptkod skall skrivas i en separat JS-fil. Ingen förekomst av detta skall förekomma i HTML-dokumentet!</i>

- Alla filer + filändelser skall vara skrivna med gemener, med undantag för **README**.
- All kod skall vara layoutmässigt genomarbetad. Layoutmässigt genomarbetad kod innebär att indragningar, talande variabelnamn, extra radbrytningar och kommentarer används för att göra koden lättläst.
- All JavaScriptkod skall vara väl kommenterad.
- Innehållet av **README.md** skall återge projektarbetets genomförande och struktureras i enlighet med de värden som specificerats i studiehandledningen kapitel 4 (ersätter laborationsrubriken med **Projektuppgift**). **Se till att dokumentet renderas korrekt innan redovisning!**
- Använd **.gitignore** för att utesluta onödiga filer och se till att lokala ändringar är synkroniserat med **remote origin** innan redovisning.
- Notera att ditt tilldelade projektrepo i sitt initiala tillstånd är tomt och att du själv måste tillföra alla nödvändiga dokument. För att undvika slarvmissar är det således lämpligt att lägga till dokument såsom **.gitignore** och **README.md** redan tidigt i arbetet.
- Avvikelser från angivna strukturkrav kommer att inverka negativt på bedömningen och omfattande brister kan leda till betygsavdrag.

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

*Kursens examination bygger på det innehåll som ingår i de olika delmomenten, samt förekommande extra krav som specificeras av projektbeskrivningen. Detta innebär att inga yttre ramverk eller teknologier andra än **HTML5**, **CSS3** samt **JavaScript** får användas. All programkod skall vara producerad av dig och inte genererats med hjälp av verktyg som transkriberar optimal programkod!*

3.1 Basimplementation

HTML-sidan skall förhålla sig till en bredd av **600px** och vara centrerad i fönstret, och det skall finnas en lämplig bakgrundsbild som alltid fyller hela fönstret. Alla nödvändiga inkluderingar av stilmall/-ar och externa skript skall förläggas till **<head>**, där det dessutom skall finnas en passande **titel** för sidan. Vidare är det strikt förbjudet att skriva JavaScript-kod i HTML-dokumentet och ni avråds från att använda s.k. **inline event handlers**, d.v.s. att ange event-attribut direkt till elementen. Istället bör ni följa goda konventioner och separera innehållet (HTML) från logiken (JS). Följande exempel illustrerar detta:

```
<!-- JS-kod direkt i HTML = STRIKT FÖRBJUDET -->
<button onclick="alert('Hello, code insertion is not allowed!');">Press me</button>

<!-- Inline event handler = AVRÅDET -->
<button onclick="someFunc()">Press me</button>

// EventListener genom JS = REKOMMENDERAT
function someFunc() { ... }
btn.addEventListener('click', someFunc);
```

Sektionen **<body>** skall bestå av huvudområdena **<header>**, **<main>** samt **<footer>**, och vad som skall ingå för var och en av dessa följer härnäst.

Försök i möjligaste mån att använda dedikerade sektionstaggar för innehåll istället för den generiska **<div>**. Du finner alla tillgängliga element för HTML5 på följande [källa](#) [4].

3.1.1 <header> & <footer>

<header> skall innehålla en rubrik (**label**) för applikationen tillsammans med en passande bild.



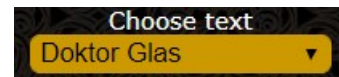
<footer> skall finnas längst ned på sidan och innehålla copyright-information samt en klickbar länk för att kontakta författaren via mail. Det skall dessutom finnas en tydlig avgränsare som skiljer denna del från övrigt innehåll.



För att göra en klickbar länk som anropar systemets mailklient så kan du titta på
<a href="mailto:..."

3.1.2 <main> : Inställningar & Textväljare

Högst upp i sektionen <main> (under <header>) skall det finnas **inställningar** samt **textväljare**. Det förstnämnda behöver du bara implementera för högre betygsnivåer, men en textväljare skall finnas även för basimplementationen. Denna skall bestå av två komponenter; en lista från vilken användaren kan ange vilken text som skall föreligga för spelet samt en överskrift (**label**) som centreras ovanför listan. När användaren klickar i listan så skall denna expandera och visa rubriken för alla tillgängliga texter och när en ny text blir vald så skall denna ersätta den nuvarande i textväljaren, samtidigt som textdetaljerna uppdateras med aktuellt innehåll (se nästa stycke). Listan skall innehålla minst tre texter!

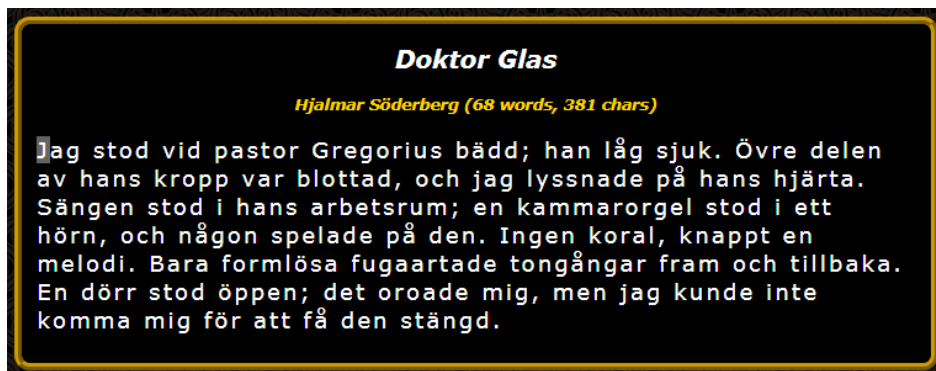


För en smidig lösning på textväljaren så kan du kika på elementet <select> och hur eventet **onchange** kan [registreras från JavaScript](#) [5].

3.1.3 <main> : Textdetaljer

Textdetaljer skall placeras under inställningarna / textväljaren och består av textkomponenter som återger detaljerna från den valda texten; **textrubrik**, **författare** samt **textinnehållet**. Dessa detaljer skall alltså dynamiskt anpassas till förändringar i textväljaren! Runt textdetaljerna skall en ram (**border**) finnas och såväl *rubrik* som *författare* skall centreras inom textområdet medan *textinnehållet* skall vara vänsterjusterat.

Ingen del av textdetaljerna får vara modifierbar, så om du exempelvis använder <textarea> för textinnehållet så måste denna anges som **readonly**.



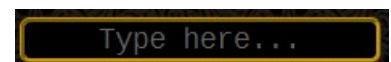
En **markör (highlighter)** skall visuellt återge progressionen av spelet, d.v.s. vilket tecken i *textinnehållet* som användaren närmast skall skriva. Vid ett nytt spel skall denna markör befinna sig på textinnehållets första tecken.

Det kan vara lite av en utmaning att implementera denna **highlighter**. Ett tips är att via JavaScript lägga varje tecken inom ett eget och sedan ange bakgrundsfärg!

Det kan vara lämpligt att lagra textrelaterad information som objekt i en array!

3.1.4 <main> : Inmatning & Start-/stoppkontroll

Precis nedanför textdetaljerna skall det finnas ett inmatningsfält där användaren skriver sin input. Denna komponent skall ha en default-text som anger detta ändamål och som enbart visas när komponenten inte har fokus. Vidare skall texten vara centrerad och löpande tömmas efter varje ord, d.v.s. att fältet blir tomt efter att användaren skrivit ett mellanslag.

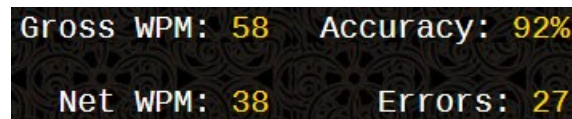


Nedanför inmatningsfältet skall det finnas en komponent (**button**) som styr start och stopp av spelet. Två lämpliga bilder skall användas som bakgrund för denna komponent och skiftas beroende på spelets status; pågående spel skall ha en bild för stopp medan en bild för start används för ej pågående spel. När spelet startas så skall all information kring tidigare spel (highlighter, statistik osv) nollställas.



3.1.5 <main> : Statistik

Löpande under pågående spel skall statistik skrivas ut i ett område nedanför inmatningsfältet och start-/stoppkontrollen. Information som skall visas är **WPM** (*words per minute*), **korrekthet i procent** samt **antal felaktigheter**. Två typer av *WPM* skall visas, en grov (**gross**) som inte tar hänsyn till felaktigheter och en netto (**net**) som inräknar dessa fel. Statistiken skall dynamiskt uppdateras under spelets gång.



Gross WPM: 58 Accuracy: 92%
Net WPM: 38 Errors: 27

För att beräkna WPM så skall du utgå från ordlängd på 5 tecken enligt formeln;

```
var grossWPM = (typed_entries / 5) / elapsed_minutes;  
var netWPM = grossWPM - (errors / min);
```

För att beräkna passerad tid så kan du använda `Date().getTime()` som returnerar aktuell tid i millisekunder!

3.2 Betygskrav

För betyget **E** skall din lösning motsvara kraven från basimplementationen, såsom beskriven ovan. För högre betyg så skall ytterligare funktionalitet tillföras applikationen. För varje nytt steg i betygstrappan förutsätts att tidigare betygskrav är uppfyllda, d.v.s. för att få betyget **A** så måste alla tidigare krav även de vara uppfyllda.

3.2.1 Betyg D

Textdetaljerna skall utökas med antal ord / tecken som vald text innehåller och presenteras tillsammans med författaren inom parentes! (*dessa skall kalkyleras vid inläsning och får inte finnas hårdkodade*).

Hjalmar Söderberg (68 words, 381 chars)

Minst två olika teckensnitt skall laddas och anges från CSS (Tips: se [@font-face](#) [1]).

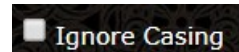
Animering; det skall finnas en repeterande animering på sidrubriken (i [<header>](#)) där du själv väljer huruvida detta skall gälla färg, storlek och / eller position. Detta skall styras från CSS och du skall använda dig av [@keyframes](#) [2].

3.2.2 Betyg C

Korrekt inmatade tecken skall ges en annan färg än kvarvarande och felaktiga inmatningar skall lämna en **röd** teckenfärg i huvudtexten.

Jag stod vid pastor Gregorius bädd; han låg sjuk. Övre delen
av hans kropp var blottad, och jag lyssnade på hans hjärta.

Ett alternativ för att ignorera versaler / gemener skall finnas i inställningarna.



Ett passande ljud skall spelas upp vid felaktiga inmatningar.

3.2.3 Betyg B

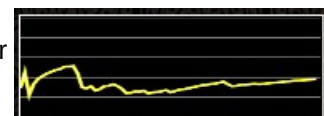
Alla textstycken skall sparas i en separat **XML**-fil (du kan använda den **XML** som följer med projektet). För att läsa in data från XML så kan du kika på [XMLHttpRequest\(\)](#) [3], vilket dock kräver att filen levereras via en webbserver. Om du använder en IDE såsom **WebStorm** så tillhandahålls en sådan där, men du kan även publicera din lösning på dina studentsidor för att detta skall fungera! (Även **JSON** är tillåtet att användas).

Språkval skall finnas bland inställningarna (svenska / engelska), där innehållet av textväljaren anpassas till valt språk. Det skall finnas minst tre texter för båda språken.



3.2.4 Betyg A

Ett diagram, ritat i en [<canvas>](#), som dynamiskt ritar ut hastighetsstatistik, skall läggas till i statistikområdet. Du kan utgå från att användaren inte skriver fortare än 100 WPM och behöver inte skriva ut skalan för diagrammet, utan är istället till för att visa användaren hur dennes hastighet utvecklats under det pågående spelet med upp- och nedgångar.



4 Utvecklingsmiljö

För webbutveckling uteslutande på klientsidan går det ofta alldeles utmärkt att använda sin favoritordbehandlare, kanske tillsammans med något plugin, och använda den *light-weight debugging* som erbjuds via webbläsaren. Det finns dock mer kompetenta miljöer som kan erbjuda fler verktyg för utvecklaren såsom **code completion**, **refaktorering**, integration av **versionshanteringssystem** osv. Den IDE som vi rekommenderar dig att använda är [JetBrains WebStorm](#) som förutom nämnda funktionaliteter även inkluderar en inbyggd webbserver, vilket underlättar delar av utvecklingen där filer behöver levereras via **server requests** (som exempelvis [XMLHttpRequest](#) för att uppfylla kraven gällande betyget **B**). *WebStorm* har även en inbyggd **FTP-klient** som du kan använda för att enkelt publicera din lösning på dina studentsidor. Som vanligt har du tillgång till alla *JetBrains* produkter via din studentmail, men för dem som inte registrerat sig redan kan så göra via [denna sida](#)!

Om du av någon anledning inte vill använda *WebStorm* för utvecklingen så finns det även andra utvecklingsmiljöer, med varierande grad av funktionalitet. Du kan även som sagt arbeta i en avskalad miljö (Atom, Sublime Text osv) och då använda exempelvis [FileZilla](#) för att publicera din lösning på dina studentsidor.

Överväg [JetBrains PHPStorm](#) då den innehar all funktionalitet som *WebStorm* har, samt fullt stöd för PHP som kommer användas i nästa webbkurs!

4.1 SFTP med WebStorm / PHPStorm

För att kunna publicera din webbapplikation bland dina studentsidor så behöver du göra följande inställningar i WebStorm:

Tools → Deployment → Configuration... och lägg till en ny konfiguration (plustecknet) där du under **Type** väljer **SFTP** och namnger med vad du vill.

Under respektive flik (**Connection** / **Mappings**) anger du följande värden (ersätter **studentid** med ditt **id**):

Connection		Mappings	
SFTP host:	minafiler.miun.se	Local path:	'sökväg till lokala projektet'
Port:	22	Deployment path:	/public_html/'specifik katalog'
Root path:	/home/studentid		
User name:	studentid		
Auth type:	Password		
Password:	'ditt studentlösen'		

Sedan kommer du åt denna fjärran värd via menyn **Tools → Deployment**:

Browse Remote Host låter dig navigera bland katalogstrukturen med fulla läs- / skrivrättigheter (grönmarkerad katalog är den förvalda mappen som du angav i konfigurationen)! **Upload here** / **to...** låter

dig ladda upp projektet till angiven katalog, medan alternativen av **Sync** utför nödvändig synkronisering av förändringar.

Webbsidan kommer du sedan åt via följande url (ersätt **studentid** med din **id** och **vald_katalog** med den katalog som projektet ligger i):

studenter.miun.se/~studentid/vald_katalog/

Referenser

- [1] MDN web docs, "@font-face", 2017. [Online]
Tillgänglig: <https://developer.mozilla.org/en-US/docs/Web/CSS/%40font-face>
[Åtkomst: 20 November 2017]
- [2] MDN web docs, "@keyframes", 2017. [Online]
Tillgänglig: <https://developer.mozilla.org/en-US/docs/Web/CSS/@keyframes>
[Åtkomst: 20 November 2017]
- [3] MDN web docs, "XHR (XMLHttpRequest)", 2017. [Online]
Tillgänglig: [https://developer.mozilla.org/en-US/docs/Glossary/XHR_\(XMLHttpRequest\)](https://developer.mozilla.org/en-US/docs/Glossary/XHR_(XMLHttpRequest))
[Åtkomst: 20 November 2017]
- [4] MDN web docs, "HTML elements reference", 2017. [Online]
Tillgänglig: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>
[Åtkomst: 27 November 2017]
- [5] MDN web docs, "change", 2017. [Online]
Tillgänglig: <https://developer.mozilla.org/en-US/docs/Web/Events/change>
[Åtkomst: 27 November 2017]