

Laboration 6

Formulär och stränghantering

– övningar/uppgifter

1M322 Webbteknik 2, 7,5hp

Medieteknik



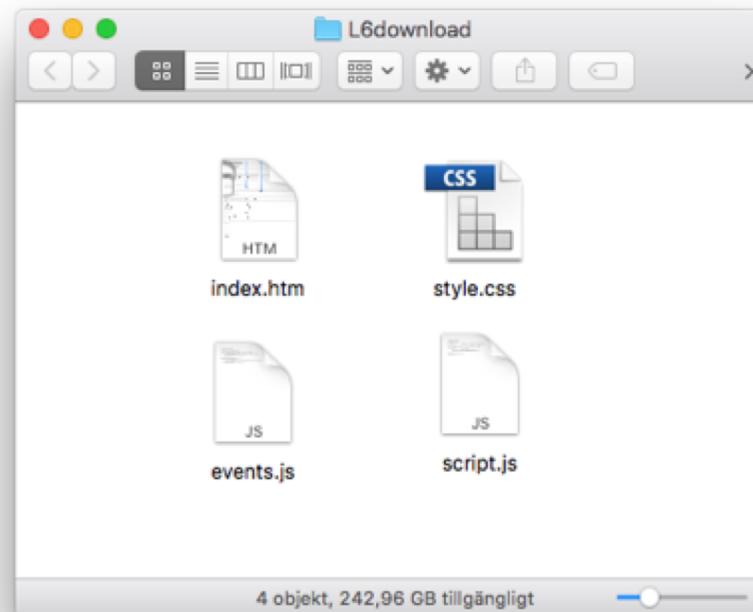
1. Ladda ner arbetsdokument

Till övningarna i denna laboration finns det ett antal filer som du kan ladda ner i en zip-fil.

Länk till zip-filen finns på laborationens webbsida.

Då du packat upp zip-filen, får du en mapp med fyra filer.

Det är i filen script.js som du ska skriva koden i övningarna.



2. Orientera dig i de nedladdade filerna

HTML- och CSS-filerna är i stort sett desamma som skapades i laboration 8 i kursen Webbteknik 1. Filerna innehåller kod för ett formulär för en rumsförfrågan på ett påhittat pensionat. Gå främst igenom HTML-filen *index.htm* och bekanta dig med koden, så du förstår hur det är strukturerat.

Filen *events.js* är densamma som i tidigare laborationer och exempel.

Det är i filen *script.js* som du ska lägga till kod i övningarna. I den nedladdade filen finns en början till en *init*-funktion.

Tillägg i övningarna

De tillägg som ska göras i övningarna är:

- Aktivera / inaktivera olika delar av formuläret, beroende på vilken rumstyp som väljs.
- Beräkna total kostnad. Varje gång användaren gör ett val som påverkar kostnaden, ska en ny beräkning göras.
- I fältet för ort ska bokstäverna ändras till versaler.
- I fälten för postnummer och telefonnummer ska innehållet kontrolleras med hjälp av reguljära uttryck.
- Kampanjkoden ska också kontrolleras med ett reguljärt uttryck. För varje nytt tecken som skrivs, ska det ges en feedback genom bakgrundsfärgen.

Pensionat Sjöstugan

Förfrågan om rumsbokning

Val av rum

Enkelrum (600 kr)
 Dubbelrum (800 kr)
 Familjerum (950 kr) — Antal personer (anges endast för familjerum)

Datum och antal nätter

* Ankomstdatum (MM-rrr-ÅÅÅÅ)
* Antal nätter

Tillägg

Internet (40 kr)
 Parkering (80 kr)
 Sjöutsikt (100 kr) - ej tillgängligt i familjerum

Kostnader per dygn.

Total kostnad

Dina val ger en kostnad på 0 kr.

Kampanj

Om du har en kampanjkod, anger du den här

Kund

* Namn
* Gatuadress
* Postnummer
* Ort
Telefon
* E-post

3a. Kontrollera om familjerum är valt eller ej

Nu ska du skriva en funktion som ska aktivera eller inaktivera olika delar av formuläret, beroende på vilken rumstyp som väljs. Om familjerum väljs ska menyn för antal personer aktiveras och alternativet sjöutsikt ska inaktiveras. För de andra två rumstyperna ska det vara tvärtom.

Händelsehanterare på knapparna

Oavsett vilken av de tre knapparna för rumstyp som man klickar på, ska funktionen för aktivering/inaktivering anropas.

- I *init*-funktionen lägger du in en *for-loop*, där du går igenom alla *formElem.roomType*, dvs radioknapparna.
- I loopen lägger du händelsen *click* och funktionen *checkIfFamilyRoom* på aktuell knapp (som indexeras med *i*) med hjälp av funktionen *addListener*.

Funktionen *checkIfFamilyRoom*

- Skapa en ny funktion och kalla den *checkIfFamilyRoom*.
- I funktionen lägger du in en *if-sats*, där du kontrollerar egenskapen *checked* för den sista knappen, dvs den med index 2.
- Om *checked* är *true* ska följande göras:
 - Menyn för personer aktiveras:
 - Sätt egenskapen *disabled* för menyn *persons* (dvs *formElem.persons*) till *false*.
 - Menyn blir då aktiv.**
 - Gå med *parentNode* upp till det omgivande elementet och ändra textfärgen till svart (#000). Se kod i figur.

```
formElem.persons.parentNode.style.color = "#000";
```

- Alternativet Sjöutsikt inaktiveras:
 - Sätt egenskapen *disabled* för det tredje alternativet *addition* till *true*.
 - Kryssrutan blir då inaktiv.**
 - Gå med *parentNode* upp till det omgivande elementet och ändra textfärgen till grå (#999).
- I *if-satsens else-del* lägger du in en kopia av ovanstående rader, fast ändra *false* till *true* och tvärtom.
Ändra också färgkoderna så #000 blir #999 och tvärtom.

Testa

- Öppna webbsidan i webbläsaren och testa.
- Klicka på de olika rumstyperna och kontrollera att menyn för antal personer och alternativet sjöutsikt aktiveras/inaktiveras.

The screenshot shows a user interface for selecting a room type. There are three radio buttons: 'Enkelrum (600 kr)', 'Dubbelrum (800 kr)' (which is selected), and 'Familjerum (950 kr)'. Below the radio buttons is a dropdown menu labeled 'Antal personer' with the value '3' selected. A note at the bottom right says '(anges endast för familjerum)'.

3b. Kontrollera om familjerum är valt eller ej

Då du öppnar sidan eller laddar om den är fel alternativ aktiva, så du ska nu se till att kontrollera knapparna för rumstyp, redan då sidan laddats in.

Funktionen *init*

- Sist i funktionen lägger du in ett anrop av *checkIfFamilyRoom*.
 - Då anropas funktionen även innan användaren klickat på någon knapp.

Testa

- Testa i webbläsaren ladda om sidan eller stäng den och öppna igen.
- Kontrollera att aktivering/inaktivering av alternativen stämmer med vald rumstyp.

4a. Beräkna total kostnad

Du ska nu skriva en funktion för att beräkna total kostnad av de val som görs. Priset för rummen finns i *value*-attributet i radioknapparna i HTML-koden. För enkelrum står det t.ex. "enkelrum,600". För att få fram priset som ska användas i beräkningen ska *value* avläsas och sedan ska värdet efter kommatecknet tas fram.

Funktionen *calculateCost*

- Skapa en ny funktion och kalla den *calculateCost*.
- Du kommer behöva följande fem lokala variabler, så börja med att deklarera dem: *i*, *elemValue*, *roomPrice*, *nightsIndex* och *nrOfNights*.

Du börjar nu med att ta fram rumspriset för valt rum:

- I funktionen lägger du in en *for*-loop där du går igenom alla *roomType* (dvs radioknapparna).
- I loopen lägger du in en *if*-sats där du kontrollerar *checked* för aktuell knapp.
 - Om *checked* är *true* ska följande göras:
 - Avläs *value*-attributet från knappen och spara i variabeln *elemValue*.
 - Ta fram rumspriset (se bild här intill):
 - Dela upp *elemValue* i en array med *split(",")*.
 - Lägg till en indexering med [1], för att få det andra värdet i arrayen.
 - Konvertera till ett tal med *Number* och spara i variabeln *roomPrice*.
 - Lägg också in en *break*-sats, för att avbryta loopen.
 - Har vald knapp hittats, behöver man inte gå runt mer i loopen.
- Efter loopen lägger du in *alert(roomPrice)*, för att kunna testa.
 - Men innan du kan testa, måste du också lägga in händelsehanterare, så funktionen anropas. Det gör du i övningen på nästa sida.

```
roomPrice = Number(elemValue.split(",")[1]);
```

4b. Beräkna total kostnad

Funktionen `calculateCost` ska anropas varje gång användaren gör ett val som påverkar priset, så det ska läggas till flera händelsehanterare på flera objekt.

Funktionen `init`

- I den loop som du har i funktionen `init`, lägger du till ytterligare en rad med `addListener`.
 - Första parametern ska vara densamma som i den `addListener` du redan har, dvs aktuell radioknapp.
 - Händelsen ska vara `click` och funktionen ska vara `calculateCost`.
 - Varje gång användaren klickar på en knapp för rumstyp, kommer alltså både funktionen `checkIfFamilyRoom` och `calculateCost` anropas.

Fler objekt:

Även tilläggen i kryssrutorna och val av antal nätter i menyn, kommer påverka priset. Så lägg in händelsehanterare för dem också, även om du ännu inte tagit med dessa delar i beräkningen av priset (men det kommer i nästa övning).

- Efter den första loopen, lägger du in ytterligare en loop, där du går igenom alla `addition`, dvs alla kryssrutor.
 - I loopen lägger du på aktuell kryssruta in händelsen `click` och funktionen `calculateCost`.
- Efter loopen lägger du till ytterligare en rad med `addListener` för elementet `nights`, dvs menyn med antal nätter.
 - Här lägger du på händelsen `change` och funktionen `calculateCost`.

Testa

- Testa i webbläsaren.
- Klicka på olika rumstyper och kontrollera att du får fram rätt pris i `alert`-rutan.
- När du sett att det fungerar, tar du bort `alert`-satsen i funktionen `calculateCost`.

4c. Beräkna total kostnad

Nu ska du också ta fram de tillägg som valts och lägga till kostnaden för dem till rumspriset.

Funktionen `calculateCost`

- Lägg till ytterligare en `for`-loop, där du går igenom alla `addition`, dvs alla kryssrutor.
 - I loopen lägger du in en `if`-sats, där du kontrollerar kryssrutans `checked`. Samtidigt kontrollerar du också att `disabled` för kryssutan inte är `true`.

```
if (formElem.addition[i].checked && !formElem.addition[i].disabled)
```

- I `if`-satsen ska följande göras:
 - Avläs `value`-attributet från kryssutan och spara i variabeln `elemValue`.
 - På samma sätt som i föregående loop, delar du upp `elemValue` i en array och indexerar dig till det andra alternativet. Konvertera till `Number` och lägg till värdet till `roomPrice`. Använd alltså `+=`.
 - I denna loop ska du inte ha någon `break`-sats, eftersom flera kryssrutor kan vara vald, så alla måste gås igenom i loopen.
- Efter loopen lägger du in `alert(roomPrice)`, för att kunna testa.

Testa

- Testa i webbläsaren.
- Klicka på olika rumstyper och tillägg. Kontrollera att du får fram rätt pris i `alert`-rutan.
- När du sett att det fungerar, tar du bort `alert`-satsen i funktionen `calculateCost`.

4d. Beräkna total kostnad

Nu ska du också ta fram hur många nätter som valts och ta med det i beräkningen av det totala priset.

Funktionen `calculateCost`

- I slutet av funktionen lägger du till följande tre rader (som också visas i figuren):
 - Avläs `selectedIndex` ur menyn `nights` och spara det i variabeln `nightsIndex`.
 - Ta sedan fram `value`-attributet ur den `option` som indexeras av `nightsIndex`, konvertera till `Number` och spara i variabeln `nrOfNights`.
 - Slutligen multiplicerar du `nrOfNights` med `roomPrice` och skriver ut i elementet för totala kostnaden (`totalCostElem`).

```
nightsIndex = formElem.nights.selectedIndex;
nrOfNights = Number(formElem.nights.options[nightsIndex].value);
totalCostElem.innerHTML = nrOfNights * roomPrice;
```

Funktionen `init`

- Sist i funktionen `init` lägger du till ett anrop av `calculateCost`.
 - Då webbsidan lästs in, görs då en första beräkning med de val som är gjorda från början.

Testa

- Testa i webbläsaren.
- Välj olika rumstyper, olika tillägg och olika antal nätter. Kontrollera att totalpriset uppdateras efter varje val.

The screenshot shows a web-based form for calculating room costs. It consists of several input fields and sections:

- Val av rum**: A group of radio buttons for room types: Enkelrum (600 kr), Dubbelrum (800 kr) (selected), and Familjerum (950 kr). Below it is a dropdown for "Antal personer" with the value "3".
- Datum och antal nätter**: Fields for "Ankomstdatum" (date) and "Antal nätter" (number of nights) set to "3".
- Tillägg**: A group of checkboxes for add-ons: Internet (40 kr), Parkeringsplats (80 kr) (selected), and Sjöutsikt (100 kr) - ej tillgängligt i familjerum (selected). Below this is a note: "Kostnader per dygn."
- Total kostnad**: A summary box stating: "Dina val ger en kostnad på 2940 kr."

5. Kontrollera fält med ort

Nu ska du skriva en funktion för fältet med ort. Då användaren skrivit en ort och lämnar fältet, ska alla bokstäver konverteras till versaler.

Funktionen *init*

- Lägg till en rad med *addListener* där du på elementet *city* i formuläret lägger på händelsen *blur* och funktionen *checkCity*.
 - Funktionen *checkCity* anropas då, när man lämnar fältet.

Funktionen *checkCity*

- Avläs *value* i fältet *city* och spara i en lokal variabel kallad *city*.
- Konvertera innehållet i variabeln *city* till versaler med funktionen *toUpperCase* och spara resultatet återigen i variabeln *city*.
- Lägg in det nya innehållet i variabeln *city* i fältet *city* i formuläret.

Testa

- Testa i webbläsaren.
- Skriv en ort med gemener i fältet för ort. Klicka sedan någonstans utanför fältet.
Bokstäverna ska då konverteras till versaler.

A screenshot of a web form. On the left, there is a label "Ort" with an asterisk (*) next to it. To its right is an input field containing the lowercase text "växjö". The input field has a blue border and a light gray background.

A screenshot of the same web form after the user has clicked outside the input field. The input field now contains the uppercase text "VÄXJÖ". The input field has a white background and a light gray border.

6. Kontrollera fält med postnummer och telefonnummer

Fälten med postnummer och telefonnummer ska kontrolleras med hjälp av reguljära uttryck. Detta ska göras på exakt samma sätt som i exempel *js6-ex4-2* i föreläsning *F6*, så du kan kopiera en del kod därifrån.

Funktionen *init*

- Lägg till händelsehanterare på fälten *zipcode* och *telephone*.
Händelsen ska vara *blur* och funktionerna ska vara *checkZipcode* respektive *checkTelephone*.
- Från exempel *js6-ex4-2* kopierar du koden i *init*-funktionen med variablerna *re* och *errMsg* och klistrar in i din *init*-funktion.
 - Ta sedan bort reguljära uttryck och felmeddelanden, så att du endast har kvar de två som gäller postnummer och telefonnummer.
- Deklarera också *re* och *errMsg* som globala variabler.

Funktionen *checkField*

- Från exempel *js6-ex4-2* kopierar du också hela funktionen *checkField*.
- Klistra in den sist i din script-fil.

Funktionen *checkZipcode*

- Skapa en ny funktion som du kallar *checkZipcode*.
- I funktionen lägger du in ett anrop av *checkField*.
 - Som första parametrar skickar du med en referens till formulärets fält *zipcode*
 - Som andra parameter skickar du med *0* (dvs index till det första reguljära uttrycket och felmeddelande).

Funktionen *checkTelephone*

- Skapa en ny funktion som du kallar *checkTelephone*.
- I funktionen lägger du in ett anrop till *checkField*, men skickar nu med en referens till fältet *telephone* och *1*.

Testa

- Testa i webbläsaren.
- Skriv olika felaktiga former för postnummer och telefonnummer i fälten och kontrollera att felmeddelandet dyker upp intill fälten.
- Skriv sedan korrekta former för värdena och kontrollera att du inte får något felmeddelande.

7a. Kontrollera kampanjkod

Nu ska kampanjkoden kontrolleras. Det ska också ske med ett reguljärt uttryck, fast feedbacken ska vara annorlunda. Så därför kan inte samma funktion som i föregående övning användas.

Medan man skriver i fältet ska bakgrundsfärgen vara röd, då innehållet ej stämmer med det reguljära uttrycket. Då innehållet är ett korrekt mönster, ska bakgrundsfärgen bli grön. Klickar man sedan utanför fältet, ska bakgrundsfärgen tas bort.

Det är då tre händelser och det blir tre funktioner.

Funktionen *init*

- Sist i funktionen *init* lägger du till tre rader med *addListener*.
 - Alla händelsehanterare ska kopplas till fältet *campaigncode* i formuläret.
 - Den första ska vara händelsen *focus* och funktionen *startCheckCampaign*.
 - Den andra ska vara händelsen *keyup* och funktionen *checkCampaign*.
 - Den tredje ska vara händelsen *blur* och händelsen *endCheckCampaign*.

Vi börjar här med den första och sista funktionen.

Funktionen *startCheckCampaign*

- Skapa en ny funktion med namnet *startCheckCampaign*.
- I funktionen byter du bakgrundsfärg på *this* till #F99.
- Lägg också in en rad med *this.select()*;
 - Då markeras hela innehållet, när man klickar i fältet.

```
this.style.backgroundColor = "#F99";
this.select();
```

Funktionen *endCheckCampaign*

- Skapa en ny funktion med namnet *endCheckCampaign*.
- Sätt bakgrundsfärgen för *this* till en tom sträng (två citattecken utan något mellan dem).
 - Då tas den tidigare bakgrundsfärgen bort.
- Ändra också innehållet i textfältet till endast versaler för bokstäver.
 - Det gör du på samma sätt som du gjorde i funktionen *checkCity*.

7b. Kontrollera kampanjkod

Nu ska du skapa funktionen `checkCampaign`. Denna funktion anropas på händelsen `keyup`, så den anropas för varje nytt tecken som användaren skriver in. Det är här kontrollen mot det reguljära uttrycket ska ske.

Funktionen `checkCampaign`

- Skapa en ny funktion och kalla den `checkCampaign`.
- Deklarera en lokal variabel som du kallar `re` och lägg in ett reguljärt uttryck för kampanjkoden.
 - En kampanjkod består av tre bokstäver, ett bindestreck, två siffror, ett bindestreck, en bokstav och en siffra, t.ex. ABC-12-D3.
 - Skriv alltså ett reguljärt uttryck för detta och spara i variabeln `re`.
Uttrycket finns alltså inte givet här, utan du måste själv formulera det.
 - Uttrycket ska också vara "case-insensitive", så det ska inte spela någon roll om man skriver små eller stora bokstäver.
- Testa sedan uttrycket mot innehållet i textfältet och sätt bakgrundsfärgen till grön eller röd. Se kod i bilden.

```
if (re.test(this.value)) this.style.backgroundColor = "#6F9";  
else this.style.backgroundColor = "#F99";
```

Testa

- Skriv en kod i textfältet.
- Så länge mönstret är fel, ska bakgrundsfärgen vara röd. Då mönstret är rätt, blir bakgrundsfärgen grön.

Kampanj

Om du har en kampanjkod, anger du den här

Kampanj

Om du har en kampanjkod, anger du den här

8. Publicera och testa

Programmet är därmed klart och du publicerar nu, som vanligt, ditt program i *Web publishing* i FirstClass, på samma sätt som du publicerade i föregående laborationer.

Kommentarer

- Gå först igenom ditt program och kontrollera att du skrivit en förklarande kommentar för varje variabel och varje funktion.

Publicera ditt program

- Lägg upp ditt program i en mapp kallad *lab6* i mappen *dold* i *Web Publishing* i FirstClass.
- Skapa en länk från ingångssidan i din portfolio (*index.htm* i *Web Publishing*) till den första sidan i ditt program (*index.htm* i *lab6*).

Testa

- Ta fram din portfolio i webbläsaren och kontrollera att länken fungerar samt att allt i ditt program fungerar.