

# Laboration 2

## Funktioner, if och loop

– övningar/uppgifter

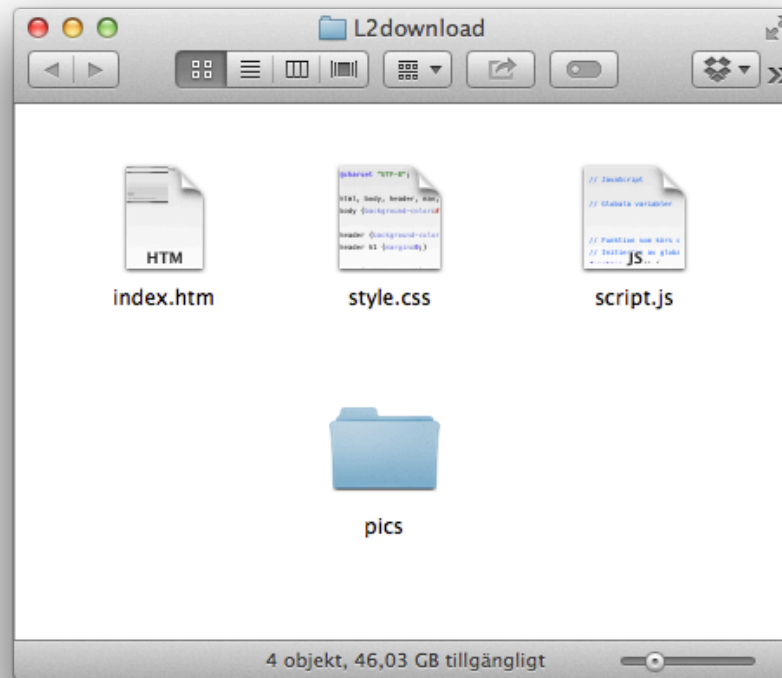
1M322 Webbteknik 2, 7,5hp

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 tre filer och ytterligare en mapp med ett antal bildfiler.



## 2. Globala variabler och init-funktionen

Programmet kommer byggas upp stegvis i ett antal övningar. Du ska skapa olika funktioner för de olika knapparna.

En del av koden visas i bilder i övningarna. Men för andra delar av koden, får du se tillbaks på föregående laboration eller exempel i föreläsningar och kursboken, för att se hur du ska skriva.

All programkod skriver du i filen *script.js*.



script.js

Du behöver en del globala variabler och en del initiering av dem i *init*-funktionen. Detta kommer också byggas ut i de kommande övningarna, men du börjar här med de första variablerna och initieringarna.

### Globala variabler

- Deklarera två globala variabler (dvs i början av filen, före funktionerna).
- Kalla variablerna *inputElem* och *msgElem*.

### Funktionen *init*

- Variabeln *inputElem* ska vara en array med referenser till de tre textfälten.
  - Skapa först en tom array.
  - I positionerna som indexeras av 1, 2 och 3 lägger du in referenser till de tre textfälten.
    - En array börjar ju med index 0, men vi lämnar den första positionen tom och använder index 1 till 3, så det blir samma nummer som i de id som används i textfälten.
- I variabeln *msgElem* lägger du in en referens till *div*-elementet med id "message".
- Lägg också in en programsats, så att funktionen *showFruit* anropas, då man klickar på knappen med id "btn1".

```
inputElem = []; // Skapa en ny Array
inputElem[1] = document.getElementById("input1");
inputElem[2] = document.getElementById("input2");
inputElem[3] = document.getElementById("input3");
```

## 3a. Visa bild med frukt

Du ska nu skapa funktionen *showFruit*, som ska avläsa ett tal i det första textfältet. Sedan ska bildfilen som motsvarar talet visas i *img*-taggen.

### Funktionen *showFruit*

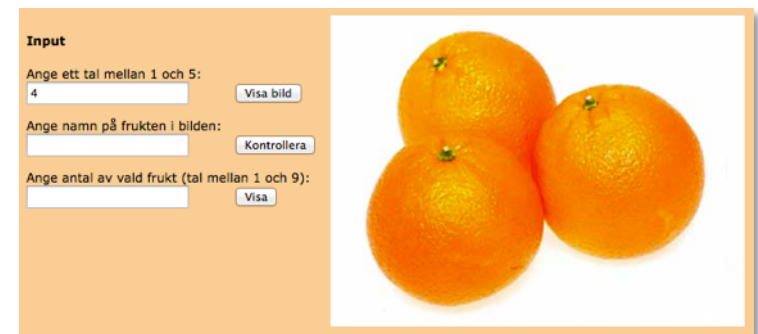
- Skapa en funktion kallad *showFruit*.
- I funktionen deklarerar du två lokala variabler, *nr* och *fruitUrl*.
- Lägg in en sats för att avläsa textfältet och spara värdet i variabeln *nr*.
- Bilderna ligger i mappen *pics* och heter *fruit1.jpg*, *fruit2.jpg*, etc. Så sätt samman en textsträng med en referens till den bildfil som bestäms av värdet i *nr*. Spara textsträngen i variabeln *fruitUrl*.
- Referera till elementet med id "*fruitImg*" och *src*-attributet. Lägg där in *fruitUrl*, för att byta ut bilden i *img*-taggen.
  - Med "lägg in" menas en tilldelning. Du har referensen till vänster om likhetstecknet och det som ska läggas in i den skriver du till höger om likhetstecknet.

```
"pics/fruit" + nr + ".jpg"
```

### Testa

- Testa programmet i webbläsaren.
- Öppna också webbkonsolen, då du testat ditt program, så att du får eventuella felmeddelanden där.

Om din kod inte fungerar, tittar du på tipsen för felsökning på nästa sida.



Skriver man in 4 och klickar på knappen "Visa bild", ska bilden med apelsiner visas.

# Om det inte fungerar

Om din kod fungerar, behöver du inte göra det som står på denna sida, men läs gärna igenom detta ändå.

Om din kod inte fungerar, ges här lite tips till felsökning.

## Debuggern

- Kontrollera i Webbkonsolen att Fel är markerat i JS-menyn. Ladda om sidan och prova igen, för att se om du får något felmeddelande.
- Om du får ett felmeddelande, rättar du felet och provar igen.

## Stega dig fram i programmet

Genom att tillfälligt lägga in en *alert*-sats på olika ställen och skriva ut en variabel eller en text, kan du se om du kommer dit i koden och om variabeln innehåller ett korrekt värde.

- Lägg in *alert("Hej");* precis i början av funktionen *showFruit*.
- Då du klickar på knappen "Visa bild" ska det dyka upp en ruta med texten "Hej".
  - Om det inte gör det, så har du ett fel i *init*-funktionen, där du kopplar funktionen *showFruit* till knappen. Kontrollera då den koden och se om du lagt till *onclick* och om du skrivit *showFruit* på likadant sätt som i funktionen, med stora och små bokstäver.
- Om du får fram "Hej", vet du att du kommer in i funktionen. Då kan du ta bort *alert*-satsen och istället lägga in en *alert(nr);* efter den rad där du läser in talet från textfältet.
- Då du skriver in ett tal och klickar på knappen ska det talet dyka upp i *alert*-rutan.
  - Om det inte gör det, så är det fel vid inläsningen. Kontrollera att du skrivit *inputElem* korrekt, indexerat med 1 och lagt till *.value* i slutet.
- Om du får fram talet, tar du bort *alert*-satsen och lägger till *alert(fruitUrl);* efter den rad där den variabeln tilldelas. Kontrollera att du får ut rätt textsträng i *alert*-rutan, då du klickar på knappen.
  - Om du inte får det, jämför du det du skrivit i koden med det som anges i övning 3a.
- Om du ändå inte får fram bilden, då du klickar på knappen, är det fel i raden där du byter ut bilden. Kontrollera då att du stavat *getElementById* rätt, att du har rätt id och att du avslutar med *.src* samt att du har variabeln *fruitUrl* efter likhetstecknet.

**Om det i alla fall inte fungerar, får du be om hjälp.**

## 3b. Visa bild med frukt

För att kunna visa korrekta bilder, ska användaren skriva en siffra mellan 1 och 5 i textfältet. men det är ju möjligt att också skriva något annat. Så du ska nu lägga till *if*-satser, för att kontrollera det som läses in från textfältet. Detta är nu nytt, så här visas hela koden för *if*-satserna.

### Kontrollera om det är ett tal

- Efter programsatsen där du läst in *nr* från textfältet, lägger du in en *if*-sats, där du kontrollerar om *nr* är ett tal eller ej. Använd funktionen *isNaN*.
  - Om det inte är ett tal, skriver du i elementet för meddelanden att man ska skriva siffror.
  - Avsluta sedan funktionen med *return*.

```
if (isNaN(nr)) {  
    msgElem.innerHTML = "Du måste skriva ett tal med siffror.";   
    return;  
}
```

### Kontrollera att talet ligger inom rätt intervall

- Lägg till ytterligare en *if*-sats, för att kontrollera om talet ligger utanför intervallet 1 till 5, dvs är mindre än 1 eller större än 5.
  - Skriv i så fall ut ett meddelande och avsluta med *return*.

```
if (nr < 1 || nr > 5) {  
    msgElem.innerHTML = "Du måste skriva ett tal med siffror.";   
    return;  
}
```

### Konvertera till ett heltal

Även om det är ett tal i rätt intervall, så kan det ju vara på fel form. Skriver man t.ex. in 3.45, så är det ett tal mellan 1 och 5, men det blir fel, om man sedan använder det till att sätta samman textsträngen för filnamnet. Så du ska nu konvertera till ett heltal, genom att ta bort en eventuell decimaldel. Det gör du med funktionen *parseInt*.

- Efter *if*-satserna lägger du till en rad där *nr* konverteras till ett heltal.
- Skriv sedan in det korrigerade värdet i textfältet, så att användaren också kan se det.

```
nr = parseInt(nr);  
inputElem[1].value = nr;
```

### Testa

- Testa programmet i webbläsaren.

Om din kod inte fungerar, kollar du felmeddelanden i debuggern eller stegar dig fram med alert, så som det beskrevs efter övning 3a.

# 4. Kommentarer

Det är alltid en god sed att skriva kommentarer i programmet, så det ska du nu också göra.

## **Förklara vad en funktion gör eller är till för**

- Ovanför funktionen *showFruit* lägger du in en rad med en kommentar som förklarar vad funktionen gör.

## **Förklara vad variabler är till för**

- Efter varje deklaration av en variabel med *var*, skriver du en kort kommentar som förklara vad variabeln är till för.
- Detta gör du både för de globala variablerna i början av filen och de lokala variabler som deklareras inuti funktionen *showFruit*.

I de kommande övningarna fortsätter du sedan med att skriva kommentarer varje gång du skapar en ny funktion eller deklarerar en ny variabel.

# 5a. Kontrollera fruktnamn

I det andra textfältet ska användaren skriva namnet på den frukt som visas i bilden. Detta ska göras i en ny funktion som ska kopplas till den andra knappen, men du behöver också ett par nya variabler.

## Globala variabler

Du behöver två nya globala variabler. En array med alla fruktnamn och en variabel med numret på vald frukt.

- Inför två nya globala variabler med namnen *fruitNames* och *fruitNr*.

## Initiering av variablerna och knappen

- I *init*-funktionen lägger du till en rad där frukternas namn läggs in i en array som sparas i *fruitNames*.
  - I första positionen skriver du "ingen frukt". Då hamnar frukterna i position 1 till 5, vilket stämmer med det nummer som användaren skriver in i det första textfältet, för att välja frukt.
- Lägg in värdet 0 i *fruitNr*.
  - Variabeln *fruitNr* ska sedan användas för att indexera *fruitNames* och från början är ingen frukt vald.
- Lägg också in en programsats, så att funktionen *checkName* anropas, då man klickar på den andra knappen.

```
fruitNames = ["ingen frukt", "äpple", "banan", "citron", "apelsin", "päron"];  
fruitNr = 0;
```

## Uppdatering av variabeln *fruitNr*

- Sist i funktionen *showFruit* lägger du in en programsats, så att *fruitNr* blir lika med *nr*.
  - Variabeln *nr* är ju en lokal variabel i *showFruit* och försvinner, då man lämnar funktionen. Men *fruitNr* är en global variabel och där sparas värdet, så att det sedan kan användas i en annan funktion.



## 5b. Kontrollera fruktnamn

Nu ska du skriva funktionen *checkName*.

### Funktionen *checkName*


- Skapa en ny funktion sist i filen och kalla den *checkName*.
- Inför en lokal variabel, som du kallar *name*.
- Läs in det andra textfältet och spara innehållet i variabeln *name*.
  - Använd ingen konvertering med *Number*. Det ska ju nu vara en text som läses in.
- I en *if*-sats jämför du om *name* är lika med *fruitNames[fruitNr]*.
  - Om de är lika skriver du "*Rätt namn.*" i fältet för meddelanden (som du refererar med *msgElem*).
  - Annars skriver du "*Fel namn.*".

### Testa

- Testa programmet i webbläsaren.

Om din kod inte fungerar, kollar du felmeddelanden i debuggern eller stegar dig fram med alert, så som det beskrevs efter övning 3a.

**Input**  
Ange ett tal mellan 1 och 5:  
   
Ange namn på frukten i bilden (endast små bokstäver):  
   
Ange antal av vald frukt (tal mellan 1 och 9):



**Meddelanden**  
Rätt namn.

# 5c. Kontrollera fruktnamn

Man kan ju skriva in ett namn och klicka på knappen Kontrollera, innan man valt en frukt. Du ska nu testa om en frukt är vald, innan namnet kontrolleras.

## Kontroll om en frukt är vald

- I funktionen *checkName* lägger du in en *if*-sats i början av funktionen (efter deklaration av variabeln).  
I denna nya *if*-sats kontrollerar du om *fruitNr* är lika med 0.
  - I så fall skriver du ut ett meddelande om att man först måste välja frukt.
  - Avsluta sedan funktionen med *return*.

## Testa

- Testa programmet i webbläsaren.

Om din kod inte fungerar, kollar du felmeddelanden i debuggern eller stegar dig fram med alert, så som det beskrevs efter övning 3a.

**Input**

Ange ett tal mellan 1 och 5:

Ange namn på frukten i bilden (endast små bokstäver):

Ange antal av vald frukt (tal mellan 1 och 9):

*Ingen frukt vald*

**Meddelanden**

Du måste först välja en frukt.

# 6a. Funktion för att läsa in ett tal från ett textfält

I det tredje textfältet ska det skrivas in ett tal mellan 1 och 9. Det görs på samma sätt som då du läser in det första textfältet i funktionen *showFruit*, fast det är en annan input-tagga och en annan övre gräns för talet. Du ska nu flytta ut koden för detta ur *showFruit* och skapa en särskild funktion för detta. Den funktionen kan sedan anropas både då man läser in tal från det första och från det andra textfältet.

## Funktionen *getNr*

- Skapa en ny funktion och kalla den *getNr*.
- Lägg in en lokal variabel, som du kallar *nr*.
- I funktionen *showFruit* kopierar du raderna där *nr* läses in från textfältet, de båda *if*-satserna samt de båda raderna där du konverterar med *parseInt* och sedan skriver in *nr* i textfältet igen. Dvs koden som visas i figuren här intill.
- Lägg in denna kod i funktionen *getNr*.

```
nr = Number(inputElem[1].value);
if (isNaN(nr)) {
  msgElem.innerHTML = "Du måste skriva ett tal med siffror.";
  return;
}
if (nr < 1 || nr > 5) {
  msgElem.innerHTML = "Du måste skriva ett tal med siffror.";
  return;
}
nr = parseInt(nr);
inputElem[1].value = nr;
```

## Parametrar och ändringar i koden

- Lägg in *elemNr* och *high*, som två parametrar i funktionen *getNr*.
  - Med *elemNr* specificeras vilket textfält som ska läsas och med *high* anges den övre gränsen för talet.
    - Värden för dessa parametrar ska alltså sedan ges, då funktionen anropas.
- I koden ändrar du indexet för *inputElem* från 1 till *elemNr*.
  - Det finns både i början och i slutet av koden.
- I den andra *if*-satsen byter du ut 5 mot *high*.

```
function getNr(elemNr, high)
```

## 6b. Funktion för att läsa in ett tal från ett textfält

Värdet från textfältet läses in till den lokala variabeln *nr*.

Detta ska nu returneras, så man kan ta emot det då *getNr* anropas.

### Returvärde

- Sist i funktionen returnerar du *nr* med *return*.
- Det finns också redan två andra *return*-satser i *if*-satserna i koden. Förut räckte det med att endast ha *return*, för att avbryta funktionen och programmet, men nu ska du där returnera *null*. Då kan man efter anropet av *getNr* kontrollera om resultatet blev *null* eller ej, för att se om man läst in en korrekt siffra.
- Byt de båda *return* till *return null*.

```
return nr;
```

```
return null;
```

### Anropa *getNr* i *showFruit*

- Den kod som du kopierade från *showFruit*, tar du nu bort. Istället lägger du där in ett anrop av *getNr*.
  - Det är det första textfältet som ska läsas in, så den första parametern ska vara 1 och den övre gränsen för talet är 5, så den andra parametern ska vara 5.
- Det värde som returneras från *getNr*, sparar du i variabeln *nr*.

```
nr = getNr(1,5);
```

### Kontrollera det returnerade värdet

- Efter anropet av *getNr* lägger du in en *if*-sats, för att kontrollera om *nr* är skilt från *null*.
- I så fall har du fått en korrekt siffra och de tre raderna för byte av bild ska utföras.

```
if (nr != null) {  
    fruitUrl = "pics/fruit" + nr + ".jpg";  
    document.getElementById("fruitImg").src = fruitUrl;  
    fruitNr = nr;  
}
```

### Testa

- Testa i webbläsaren att det nu fortfarande fungerar att skriva in ett tal i det första textfältet och få fram bilden för frukten.

Om din kod inte fungerar, kollar du felmeddelanden i debuggern eller stegar dig fram med alert, så som det beskrevs efter övning 3a.

## 7a. Lägg till frukter i en lista av valda frukter

Den sista funktionen du ska skriva i denna laboration, är en funktion som ska avläsa det tredje textfältet. Sedan ska du lägga till bilder av vald frukt i *div*-elementet med id "*selectedFruits*". Du behöver då ytterligare en global variabel med en referens till det *div*-elementet.

### Globala variabler och initiering

- Inför den globala variabeln *selfFruitsElem*.
- I *init*-funktionen tar du fram en referens till elementet med id "*selectedFruits*" och sparar i *selfFruitsElem*.
- Lägg också in en programsats, så att funktionen *addFruits* anropas, då man klickar på den tredje knappen.

### Funktionen *addFruits*

- Skapa en ny funktion och kalla den *addFruits*.
- Deklarera två variabler, *amount* och *imgList*.
  - Variabeln *amount* ska vara det tal som läses in från textfältet och *imgList* ska vara en textsträng med HTML-kod som ska läggas in i elementet "*selectedFruits*".
- Lägg in en *if*-sats, som kontrollerar att en frukt är vald, annars avbryter du med *return*.
  - Denna *if*-sats blir likadan som i funktionen *checkName*.
- Avläs sedan textfältet genom att anropa *getNr*. Det är nu det 3:e textfältet och övre gränsen är 9. Spara det returnerade värdet i variabeln *amount*.
- I en *if*-sats kontrollerar du om *amount* är skilt från *null*.
  - I så fall ska det som beskrivs i nästa övning utföras.

# 7b. Lägg till frukter i en lista av valda frukter

Här skriver du den kod som ska finnas i *if*-satsen i föregående övning.

## Kod i *if*-satsen

- I variabeln *imgList* ska du lägga till kod för *img*-taggar med vald frukt (dvs den som bestäms av *fruitNr*).
- Det ska vara så många *img*-taggar som bestäms av det inlästa talet (som nu finns i *amount*).  
Så detta ska göras i en *for*-loop, där du lägger till en *img*-tagg för varje varv i loopen.
- Före loopen måste du då initiera *imgList* som en tom sträng (två citationstecken utan något emellan).
- Efter loopen lägger du till (med +=) *imgList* till elementet som bestäms av *selfFruitsElem*.
  - Om man sedan väljer en annan frukt och lägger till den, kommer listan att byggas på med fler och fler bilder.
- I början av funktionen deklarerar du också variabeln *i* med var.

```
imgList = "";  
for (i=0; i<amount; i++) {  
    imgList += "<img src='pics/fruit' + fruitNr + '.jpg' alt='frukt'>";  
}  
selfFruitsElem.innerHTML += imgList;
```

## Testa

- Testa i webbläsaren.


Om din kod inte fungerar, kollar du felmeddelanden i debuggern eller stegar dig fram med alert, så som det beskrevs efter övning 3a.

**Input**

Ange ett tal mellan 1 och 5:











Ange namn på frukten i bilden (endast små bokstäver):





Ange antal av vald frukt (tal mellan 1 och 9):



**Meddelanden**

**Valda frukter**





## 8. Publicera och testa

Då du är klar med ditt program publicerar du det i *Web publishing* i FirstClass, på samma sätt som du publicerade labb 1.

### 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 *lab2* 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 *lab2*).

### Testa

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