

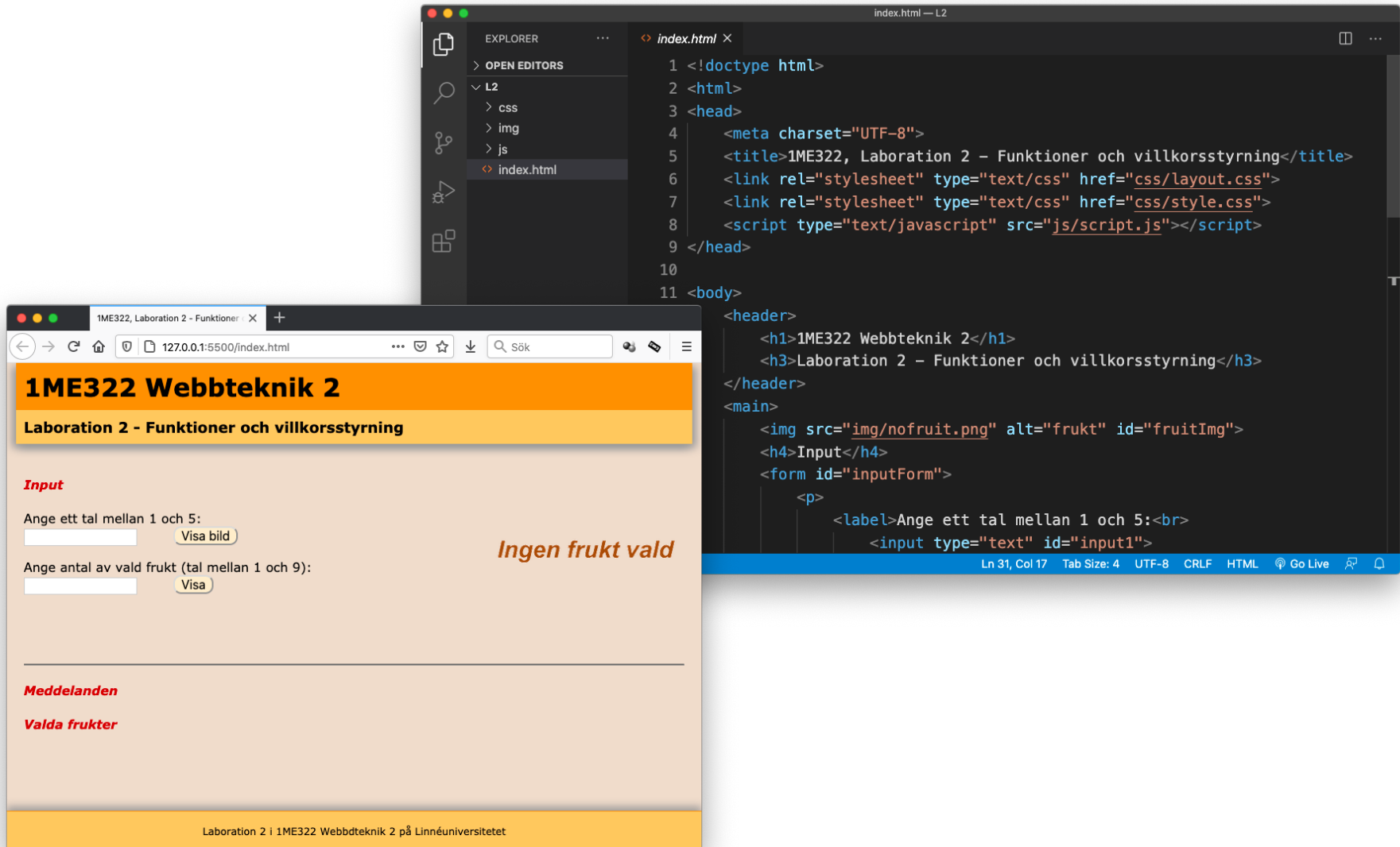
# Laboration 2

## Funktioner och villkorssatser

1M322 Webbteknik 2, 7,5hp  
Medieteknik

## 1. Öppna mappen L2

Öppna mappen *L2* i Visual Studio Code (VSC) och öppna filen *index.html* i Live Server.



## 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*.



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 tre globala variabler (dvs i början av filen, före funktionerna).
- Kalla variablerna *input1Elem*, *input2Elem* och *msgElem*.

### Kod i funktionen *init*

- Variablerna *input1Elem* och *input2Elem* ska vara referenser till de två textfälten.
- I variabeln *msgElem* lägger du in en referens till *div*-elementet med id "*message*".

```
input1Elem = document.getElementById("input1");  
input2Elem = document.getElementById("input2");  
msgElem = document.getElementById("message");
```

- Lägg också in en programsats, så att funktionen *showFruit* anropas, då man klickar på knappen med id "*btn1*".

# 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*.
- Lägg in en sats för att avläsa textfältet och spara värdet i variabeln *nr*.
  - Konvertera till datatypen *Number*, för att det ska fungera i *switch*-satsen som du sedan ska skriva.
- Lägg in en *let*-sats och definiera en variabel kallad *url*.
- Skriv en *switch*-sats, där du kontrollerar *nr*. Ta med *case*-satser för värdena 1 till 5, där du lägger in adressen till en bild i variabeln *url*. I *default*-satsen lägger du in bilden "nofruit.png" i *url*.
- Referera till elementet med id "*fruitImg*" och *src*-attributet. Lägg där in *url*, 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.

```
let nr = Number(input1Elem.value);
```

```
case 1: url = "img/apple.png"; break;
```

## Testa programmet i webbläsaren

- Öppna också webbkonsolen, då du testar ditt program, så att du ser eventuella felmeddelanden där.

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

**Input**

Ange ett tal mellan 1 och 5:

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



*Skriver man in 3 och klickar på knappen "Visa bild", ska bilden som du la in i url i "case 3"-satsen 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 Verktyg för webbutvecklare (fliken Konsol) 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 *input1Elem* korrekt och lagt till *.value* i slutet.
- Om du får fram talet, tar du bort *alert*-satsen och lägger till *alert(url);* efter den *switch*-satsen. Kontrollera att du får ut rätt textsträng i *alert*-rutan, då du klickar på knappen.
  - Om du inte får det har du fel i koden för *switch*-satsen. Jämför du det du skrivit i koden med det som anges i övning 3a. Jämför också med exempel i föreläsning och bok, för att se hur *switch*-satsen ska skrivas.
- 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 *url* 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.
  - Avbryt 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 avbryt funktionen med *return*.

```
if (nr < 1 || nr > 5) {  
    msgElem.innerHTML = "Du måste skriva ett tal mellan 1 och 5";  
    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 i *switch*-satsen, för att bestämma *url*. 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);  
input1Elem.value = nr;
```

### Testa programmet i webbläsaren

Om din kod inte fungerar, kollar du felmeddelanden i debuggern eller stegar dig fram med alert, så som det beskrevs i tipsen på föregående sida.

# 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 *let* eller *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 deklarerar 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. Dela upp i fler funktioner

Nu ska du dela upp *showFruit* i flera funktioner, dels för att strukturera koden och dels för att en del ska kunna återanvändas, då du sedan skriver koden för det andra textfältet och knappen.

## Funktionen *getUrl*

- Skapa en ny funktion sist i filen och kalla den *getUrl*.
- Inför en parameter, som du kallar *nr*.
- Flytta *let*-satsen där du inför variabeln *url* samt hela *switch*-satsen från *showFruit* till *getUrl*.
  - Det är alltså en del av den kod som du skrev i övning 3a.
- Lägg också in en *return*-sats som returnerar *url* i slutet av funktionen.

```
function getUrl(nr) {  
  let url;  
  switch (nr) {  
    case 1: url = "img/apple.png"; break;  
    case 2: url = "img/pear.png"; break;  
    case 3: url = "img/orange.png"; break;  
    case 4: url = "img/banana.png"; break;  
    case 5: url = "img/pineapple.png"; break;  
    default: url = "img/nofruit.png";  
  }  
  return url;  
} // End getUrl
```

## Funktionen *showFruit*

- Ändra satsen där bilden ska visas, så att du till höger om likhetstecknet har ett anrop av *getUrl*. Skicka med *nr* som parameter.
  - Observera att det som skickas som parameter är innehållet i den variabel *nr* som finns i funktionen *showFruit*.

```
document.getElementById("fruitImg").src = getUrl(nr);
```

Testa programmet i webbläsaren



# 5b. Dela upp i fler funktioner

Även kontrollen av indata ska läggas i en separat funktion.

## Funktionen *checkNr*

- Skapa en ny funktion sist i filen och kalla den *checkNr*.
- Inför två parametrar, som du kallar *nr* och *high*.
- Flytta de båda *if*-satserna samt satsen med *parseInt* från *showFruit* till *checkNr*.
  - Det är alltså all kod, utom den sista raden, som du skrev i övning 3b.
- Ändra också villkoret och texten i den andra *if*-satsen, så att du använder parametern *high* istället för talet 5.
- Lägg till *null* efter *return* i *if*-satserna.
  - Då vi använder funktionen ska vi sedan kontrollera det returnerade värdet, för att se om indata var korrekt eller ej.
- Lägg också in en *return*-sats som returnerar *nr* i slutet av funktionen.

```
if (isNaN(nr)) {  
    msgElem.innerHTML = "Du måste skriva ett tal med siffror";  
    return null;  
}  
if (nr < 1 || nr > high) {  
    msgElem.innerHTML = "Du måste skriva ett tal mellan 1 och " + high;  
    return null;  
}  
nr = parseInt(nr);
```

## Funktionen *showFruit*

- Ändra den första raden, så att *checkNr* anropas.

```
let nr = checkNr(input1Elem.value,5);
```

  - Den första parametern är värdet i textfältet och den andra parametern är den övre gränsen för tillåtna värden.
  - Vi behöver inte längre konvertera indata till *Number*, eftersom det ändå görs med *parseInt* i funktionen *checkNr*.
  - Det returnerade värdet sparas i variabeln *nr*, som här har samma namn som variabeln i *checkNr*. Men det är två olika variabler, så de skulle kunna ha olika namn.
- Efter ovanstående rad lägger du in en *if*-sats, där du kontrollerar om *nr* är lika med *null*. I så fall ska *return* utföras.
  - Du kontrollerar alltså det värde som returnerades från *checkNr*. Om indata är fel, returnerar ju funktionen *null*. I så fall ska du lämna funktionen *showFruit* med en *return*-sats.
  - I annat fall går programmet vidare med de återstående raderna och byter ut bilden.

Testa programmet i webbläsaren

fortsättning på  
nästa sida ...

## 5b. Dela upp i fler funktioner

*... fortsättning från föregående sida*

Om man först skriver in något felaktigt i textfältet, ges ett meddelande om det. Då man sedan skriver in ett korrekt värde, ligger meddelandet kvar. Så du ska nu se till att det tas bort.

### Funktionen *checkNr*

- Lägg in en rad i början av funktionen där du skriver en tom textsträng, två citattecken utan något mellan dem, i *msgElem*.

```
msgElem.innerHTML = "";
```

Testa programmet i webbläsaren

## 6a. 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 andra 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 samt en global variabel som håller reda på vilken frukt som valts.

### Globala variabler och initiering

- Inför de globala variablerna *selfFruitsElem* och *selfFruitNr*.
- I *init*-funktionen tar du fram en referens till elementet med id "*selectedFruits*" och sparar i *selfFruitsElem*.
- Variabeln *selfFruitNr* sätter du till 0.
- Lägg också in en programsats, så att funktionen *addFruits* anropas, då man klickar på den knappen med id "*btn2*".

### Funktionen *showFruit*

För att sedan veta vilken frukt som är vald, ska du i funktionen *showFruit* spara numret i en global variabel.

- Lägg in en rad sist i funktionen, där du lägger in *nr* i *selfFruitNr*.

```
selfFruitNr = nr;
```

# 6b. Lägg till frukter i en lista av valda frukter

## Funktionen *addFruits*

- Skapa en ny funktion och kalla den *addFruits*.
- Lägg in en *if*-sats, som kontrollerar att en frukt är vald, annars avbryter du med *return*.
- Avläs det andra textfältet genom variabeln *input2Elem* och skicka med dess värde samt gränsen 9 som parametrar till funktionen *checkNr*. Det returnerade värdet sparar du i variabeln *amount*.
  - Denna rad blir likadan som den första raden i *showFruit*, fast med variabeln *amount* och gränsen 9.
- I en *if*-sats kontrollerar du om *amount* är *null*. I så fall avslutar du funktionen med *return*.
  - Annars går programmet vidare och utför det som beskrivs nedan.

```
let amount = checkNr(input2Elem.value,9);
```

## Kod för att visa bilder

- Anropa *getUrl* med parametern *selFruitNr* och spara resultatet i variabeln *fruitUrl* i en *let*-sats.
- I variabeln *imgList* ska du lägga till kod för *img*-taggar med vald frukt.
- Det ska vara så många *img*-taggar som bestäms av det inlästa talet (som 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 *selFruitsElem*.
  - Om man sedan väljer en annan frukt och lägger till den, kommer listan att byggas på med fler och fler bilder.

```
let fruitUrl = getUrl(selFruitNr);
let imgList = "";
for (let i = 0; i < amount; i++) {
    imgList += "<img src='" + fruitUrl + "' alt='frukt'>";
}
selFruitsElem.innerHTML += imgList;
```

## Testa programmet i webbläsaren

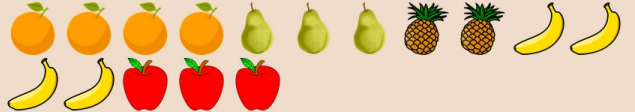
**Input**

Ange ett tal mellan 1 och 5:

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

**Meddelanden**

**Valda frukter**



# 7. Städa koden och lägg till kommentarer

Då du nu är klar med programmet ska du se över det och eventuellt snygga till koden samt lägga in kommentarer för funktioner och variabler.

## Indentering, tomrader, m.m.

- Kolla så att alla indenteringar är prydliga. Alla satser inom ett block, t.ex. i en funktion, i en loop eller en if-sats, ska vara inskjutna, så att man tydligt ser vad som ingår i blocket
- Funktionerna bör vara åtskiljda med en tomrad eller en kommentar som tydligt markerar gränsen.
- Ibland kan tomrader användas mellan olika avsnitt av koden, för att markera att de hanterar olika delar, men i övrigt ska det inte finnas några omotiverade tomrader.

## Kommentarer

- Alla funktioner ska ha en kommentar som beskriver vad de gör.
- Alla konstanter och variabler ska då de införs med *const*, *var* eller *let*, ges en kort kommentarer som förklarar vad de är till för.
  - Undantag är loopvariabler som inte behöver förklaras.
  - Då variabeln sedan används igen, behövs ingen kommentar.

```
let count = 0; // Totalt antal frukter
...
count = count + 5;
...
for (let i = 0; i < 9; i++) {
    ...
}
```

```
// Förklaring av funktionen
function qwerty() {
    ...
    for (...) {
        ...
    }
    if (...) {
        ...
    }
    else {
        ...
    }
} // End qwerty
```

```
// Förklaring av funktionen
function aaa() {
    ...
} // End aaa

// Förklaring av funktionen
function bbb() {
    ...
} // End bbb
```

```
// Förklaring av funktionen
function aaa() {
    ...
} // End aaa
// -----
// Förklaring av funktionen
function bbb() {
    ...
} // End bbb
```