

## Tips till läraren

I denna fil beskrivs några vanliga misstag som eleverna ibland kan göra.

### Vid kursstart när eleven ska ladda ner filer till sin egen dator

- Kan inte komma åt internet.  
*Lösning:* Ha med några USB-stickor som du kopierat filerna till i förväg. Kopiera filerna till elevens dator från en sådan USB-sticka istället.
- Kan komma åt internet, men hittar inte den rätta webbsidan för boken.  
*Lösning:* Kontrollera att eleven skrivit rätt adress.  
Den korrekta adressen för elevboken står på baksidan av boken.  
Den är [www.studentlitteratur.se/39785](http://www.studentlitteratur.se/39785).  
(För lärarboken är adressen [www.studentlitteratur.se/39883](http://www.studentlitteratur.se/39883).)

### När eleven ska köra filerna som laddats ner

- Har laddat ner filerna men det händer inget när man klickar på någon av filerna `demo.html` eller `start.html`.  
*Lösning:* Eleven klickar förmodligen på filerna i den ouppackade zip-filen. Packa upp zip-filen, eller markera alla filerna och kopiera dem till en ny egen mapp.

### När eleven ska köra ett program (nedladdat eller eget)

- Försöker köra något program genom att klicka på JavaScript-filen, men får ett konstigt felmeddelande.  
*Lösning:* Alla JavaScript-program måste (vilket beskrivs i boken) köras med hjälp av filen `start.html`. Klicka på den istället och skriv programmets namn.
- Klickar på filen `start.html`. En webbläsare startas men det fungerar inte som det ska.  
*Lösning:* Kontrollera att eleven inte använder den gamla webbläsaren Internet Explorer. Använd helst Google Chrome. Edge går också bra.
- Klickar på filen `start.html` men själva filen visas. Man får ingen ruta med en fråga.  
*Lösning:* Högerklicka på filen `start.html` och välj alternativet *Öppna med* och klicka på Google Chrome.
- Kör filen `start.html`, men den hittar inte filen man skrivit namnet på.  
*Lösning:* Filen `start.html` och den fil man vill köra måste ligga i *samma* mapp. Om man har lagt filerna som ska köras i en egen mapp måste man se till att klicka på filen `start.html` i den egna mappen. (Om det inte finns någon `start.html` fil där, kan man kopiera den dit.)  
Tips: Om eleven kör Windows kan man lägga genvägar till de olika startfilerna på skrivbordet. Då behöver man inte bläddra i mapparna.
- Har skrivit ett eget program och sparat det med hjälp av texteditorn. Klickar på `start.html` i *samma* mapp, men den säger att den inte hittar programmet.  
*Lösning 1:* Filen som eleven skapat måste ha ett namn som slutar på `.js`. Kontrollera detta och döp om filen om det behövs.  
*Lösning 2:* Kontrollera att namnet på filen som eleven skapat inte innehåller några mellanslag (blanka tecken).

## När eleven kör ett program och det inte fungerar som det ska

- Inget händer när man försöker köra programmet.  
*Lösning 1:* Kontrollera att det inte finns några felmarkeringar i texteditorn Caret.  
*Lösning 2:* Använd *Verktyg för programmerare* så som beskrivs i kapitel 4 och se om det finns några felmarkeringar som Caret missat. Ett vanligt fel är att man stavat fel på något namn, t.ex. skrivit liten bokstav istället för stor.
- Alla frågor eller inmatningsrutor visas inte.  
*Lösning:* Eleven har förmodligen kopierat och klistrat in kod för att få fler inmatningsrutor, men har glömt att ändra variabelnamnet (t.ex. fraga1 till fraga2) på *alla* ställen.

## När eleven ska lösa uppgift 3.5 på sidan 37

I första tryckningen av elevboken finns tyvärr ett tryckfel i uppgift 3.5. Filen man ska leta efter heter `hej.js`, inte `typer.js` som det står i boken.

## Enkla programmeringsfel som nybörjare ofta gör

- Stavar fel på namn. Gäller speciellt stora och små bokstäver.
- Läger in mellanslag (blanka tecken) inne i variabelnamn.
- Skriver citationstecken runt variabelnamn.
- Försöker använda å, ä och ö i variabelnamn.
- Sätter klamrar felaktigt. Kolla att det finns lika många vänster- och höger-klamrar och att de står på rätt ställen. Detta gäller speciellt när man har **if**-satser och loopar.
- Har inte ordning på citationstecken. Dessa måste stå i början och slutet av varje text.
- Placerar sådana programrader som ska utföras när man klickar på knappen eller trycker på Enter utanför funktionen `klicka`.
- Skriver decimalkomma istället för decimalpunkt.
- Läger satserna i fel ordning.
- Deklarerar om en variabel som redan finns inne i en **if**-sats eller en loop.  
Exempel.

```
let a = 0;
if (i > 0) {
  let a = a + 1;    // OBS! Inget let här. Då skapas en ny variabel
}
```

- Tilldelar en text till `resultat.innerHTML` två gånger.  
Exempel.

```
resultat.innerHTML = "Text 1";
resultat.innerHTML = "Text 2";
```

Här kommer den andra texten att ersätta den första texten. Alltså kommer alltid "Text 2" att visas. Vill man visa båda texterna gör man så här istället:

```
resultat.innerHTML = "Text 1" + " " + "Text 2";
```

- Använder inte **else** på rätt sätt när det är en flervalssituation.

Exempel:

```
if (i == 0) {  
    resultat.innerHTML = "noll";  
}  
if (i > 0) {  
    resultat.innerHTML = "större än noll";  
}  
else {  
    resultat.innerHTML = "mindre än noll";  
}
```

Här kommer alltid texten "mindre än noll" att visas, även om variabeln i skulle vara lika med noll.

*Lösning:* Lägg in **else** även i den första **if**-satsen.

```
if (i == 0) {  
    resultat.innerHTML = "noll";  
}  
else if (i > 0) {  
    resultat.innerHTML = "större än noll";  
}  
else {  
    resultat.innerHTML = "mindre än noll";  
}
```

- Skriver ett semikolon direkt efter villkoret i en **if**- eller **while**-sats.

Exempel:

```
if (i == 0); // OBS! Inget semikolon här  
{  
    ...  
}
```