

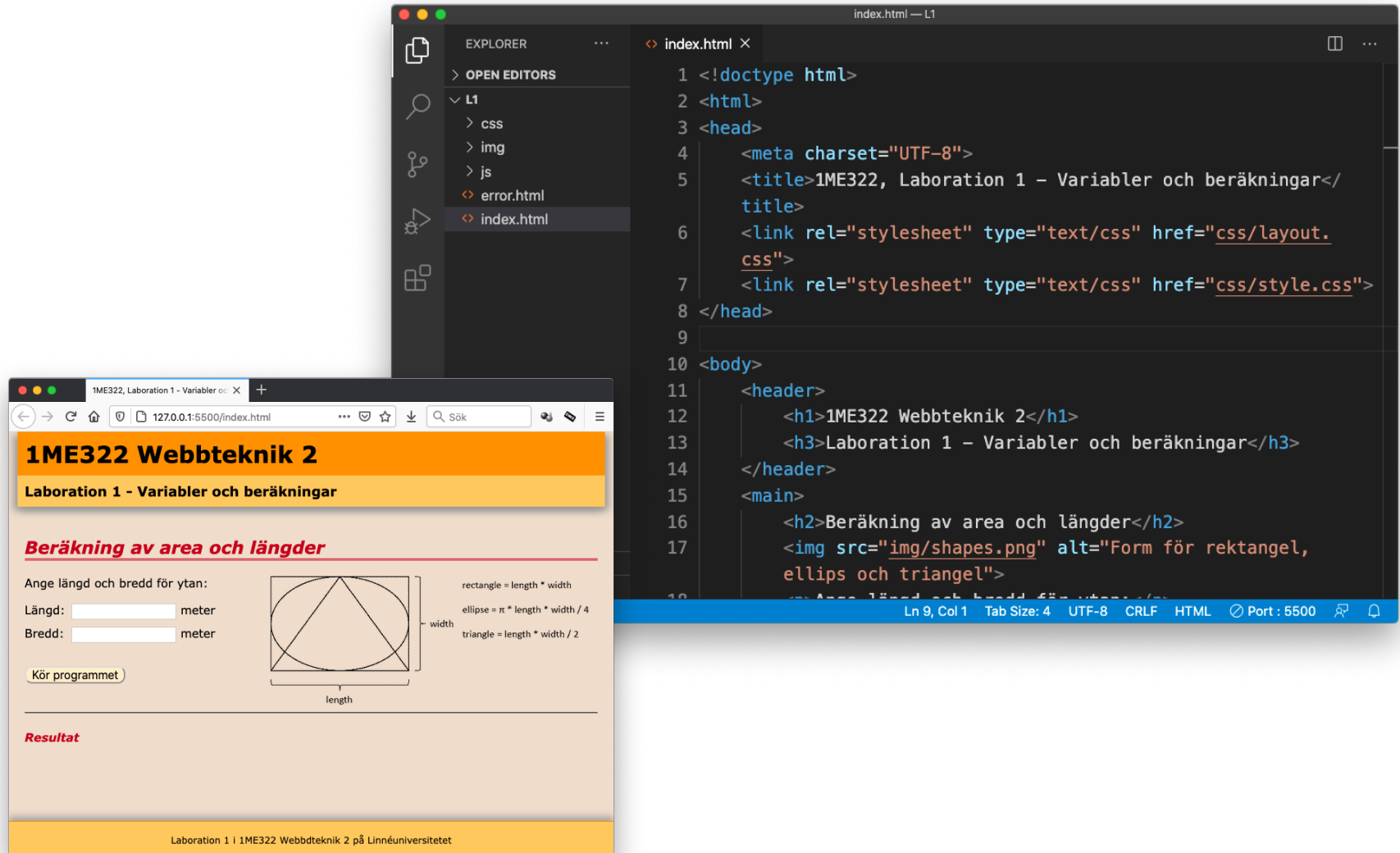
Laboration 1

Ett första program, variabler samt debugging

1M322 Webbteknik 2, 7,5hp
Medieteknik

1. Öppna mappen L1

Öppna mappen *L1* i Visual Studio Code (VSC) och öppna filen *index.html* i Live Server.



2. Länk till scriptfil

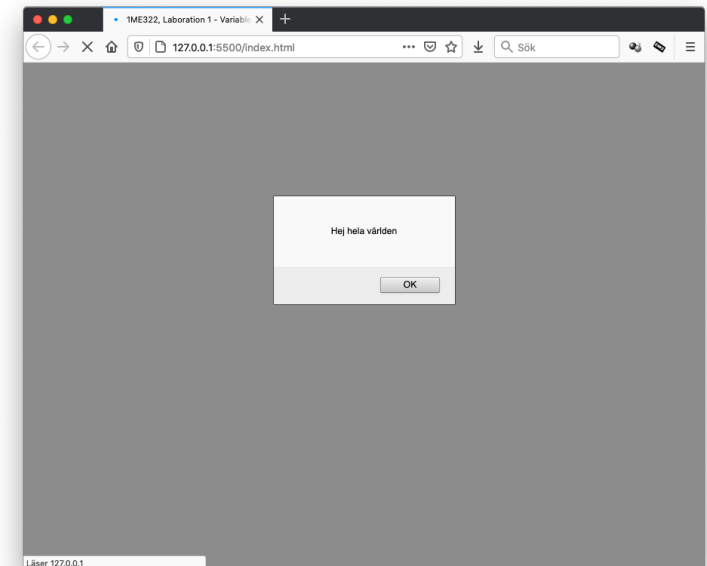
Du ska nu skapa en fil för JavaScript-koden och länka in den i HTML-filen.
För att testa att det funkar, ska du skriva ut ett meddelande med alert.

- Skapa en ny fil i mappen *js* och kalla den *script.js*.
- Lägg in kommandot `alert("Hej hela världen");` i filen.
 - Det är det enda som just nu ska ligga i filen, för att testa den.
- Öppna filen *index.htm*.
- I *head*-delen lägger du in ett *script*-element som refererar till filen *script.js*.

```
alert("Hej hela världen");
```

Testa i webbläsaren

- Varje gång du klickar på "reload"-knappen, får du rutan med meddelandet.
- Observera att det är tomt bakom rutan. Rutan kommer innan resten av webbsidan lagts upp.



3a. En init-funktion

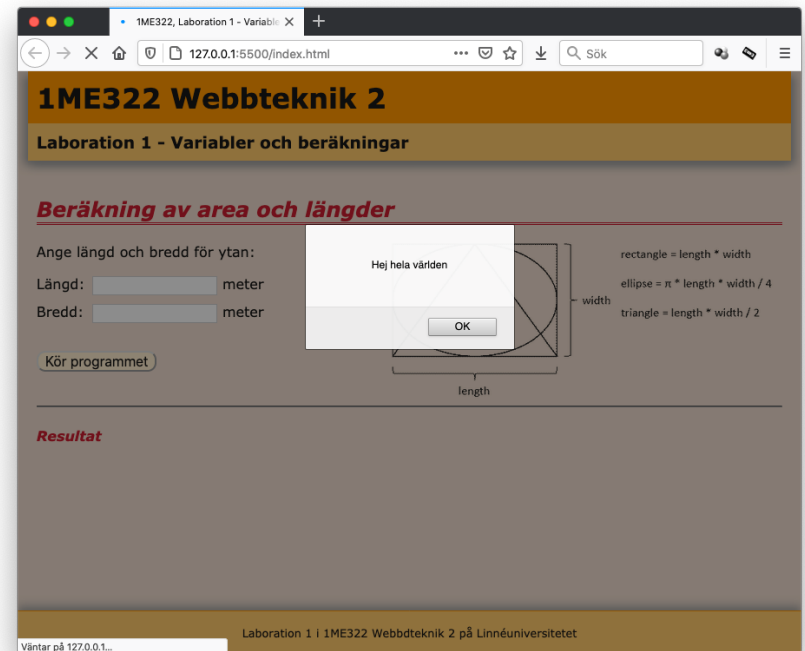
Nu ska du i script-filen lägga in en funktion som anropas, då webbläsaren läst in hela webbsidan. Du kommer skapa en kod som liknar den som finns i exemplen föreläsning 1, fast du gör det här i två steg.

- I filen *script.js* skriver du en funktion som du kallar *init*.
- Flytta *alert*-satsen med din hälsning till funktionen, så att den ligger mellan klammrarna för funktionen.
- Efter funktionen skriver du *window.onload = init;*
 - Detta gör att *init* utförs, då hela webbsidan är inladdad i fönstret.

```
function init() {  
    alert("Hej hela världen");  
} // End init  
window.onload = init;
```

Testa i webbläsaren

- Klicka på "reload"-knappen och se att webbsidan först kommer fram och därefter rutan med meddelandet.



3b. En testfunktion

Nu ska du skapa ytterligare en funktion och koppla den till knappen på webbsidan.


- I *script.js* lägger du till en funktion som du kallar *areaCalculations*.
 - Du ska senare lägga till kod för att utföra beräkningar.
- Flytta *alert*-satsen från *init* till *areaCalculations*.
- I *init*-funktionen skriver du in de fyra rader som finns i rutan här intill.
- De tre variablerna deklarerar du ovanför funktionen.
 - De måste ligga där, för att du sedan även ska kunna använda dem i den andra funktionen.

Du refererar här till *id*-attribut i HTML-koden. Så titta även i filen *index.htm*, och se vilka element det refereras till.

```
var input1Elem, input2Elem, resultElem;

function init() {
    input1Elem = document.getElementById("input1");
    input2Elem = document.getElementById("input2");
    resultElem = document.getElementById("result");
    document.getElementById("runBtn").onclick = areaCalculations;
} // End init
window.onload = init;

function areaCalculations() {
    alert("Hej hela världen");
} // End areaCalculations
```



Testa i webbläsaren

- Klicka på knappen, så ska meddelandet med *alert* dyka upp.

I exemplen i föreläsning 1 ser du också att det finns kommentarer vid variablerna och funktionerna, som förklarar vad de är till för. Man bör alltid ta med förklarande kommentarer i sin kod, så lägg in sådana kommentarer i din kod också.

4. Läs input och skriv på sidan

Nu ska du skriva in kod i funktionen *areaCalculations*, för att avläsa textfälten och skriva i *div*-elementet på sidan.

- I funktionen *areaCalculations* tar du nu bort *alert*-satsen och lägger istället in det följande.
- Deklarera två variabler, *length* och *width*.
- Läs in det första textfältet och spara i variabeln *length*.
Det andra textfältet sparar du i variabeln *width*.
- Det ska vara tal som ska skrivas i textfälten, så konvertera till den datatypen med *Number*.
 - Vi gör dock nu ingen kontroll av att användaren verkligen skriver tal. Det kommer i laboration 2.
- Skriv ut variabeln *length* på webbsidan, för att testa att det fungerar att avläsa fältet och skriva på sidan.

```
function areaCalculations() {  
    var length;    // Längd i meter  
    var width;     // Bredd i meter  
  
    length = Number(input1Elem.value);  
    width = Number(input2Elem.value);  
  
    resultElem.innerHTML = length;  
} // End areaCalculations
```

Testa i webbläsaren

- Skriv något tal i det första textfältet och klicka på knappen. Det du skrev i textfältet ska då skrivas på sidan under rubriken Resultat.

Längd: meter

Bredd: meter

Resultat

4

5. Uttryck

Nu ska du beräkna arean för en rektangel med de mått som användaren skriver i textfälten.

- Deklarera variabeln *area* under de andra variablerna i början av funktionen.

```
var length;    // Längd i meter
var width;     // Bredd i meter
var area;      // Yta i kvadratmeter
```

- Efter raderna där du läste in *length* och *width* lägger du in ett uttryck för att beräkna arean. Resultatet sparas i variabeln *area*.
- Ändra utskriften till att skriva ut *area* tillsammans med någon lämplig text och HTML-kod, för att formatera det.

```
// Area för en rektangel
area = length * width;
resultElem.innerHTML = "<p>Rektangelns area är " + area + " m<sup>2</sup></p>";
```

Testa i webbläsaren

- Skriv in några värden i textfälten och klicka på knappen.

Ange längd och bredd för ytan:

Längd: meter

Bredd: meter

Resultat

Rektangelns area är 12 m²

6. Fler uttryck

Nu ska du beräkna arean för en ellips.

- Sist i funktionen *areaCalculations* lägger du till kod för att beräkna den nya arean.
 - Variabeln *area* återanvänds nu och får ett nytt värde.
 - I formeln för en ellips area ingår π , men vi approximerar det här till 3.14.
- Skriv ut *area* tillsammans med lämplig text.
- Använd nu operatorn +=
 - Då lägger du till den nya texten, utan att ta bort den gamla.

```
// Area för en ellips
area = 3.14 * length * width / 4;
resultElem.innerHTML += "<p>Ellipsens area är " + area + " m<sup>2</sup></p>";
```

Testa i webbläsaren

- Värdet för arean blev nu ett flyttal med några decimaler. Beroende på vad du skrivit i textfälten, kan det bli många decimaler, men vi struntar i det nu. Det går avrunda till heltal med *parseInt*, men vi tar det i nästa laboration.

Ange längd och bredd för ytan:

Längd: meter

Bredd: meter

Kör programmet

Resultat

Rektangelns area är 12 m²

Ellipsens area är 9.42 m²

7. Ännu ett uttryck

Nu ska du beräkna arean för en rektangel där längden ökas med 5.

- Lägg till kod för beräkning av uttrycket och utskrift av det sist i funktionen.
 - Observera att man här behöver en parentes, för att först öka längden, innan arean beräknas.

```
// Areal om bredden är bredden ökas med 5
area = (length + 5) * width;
resultElem.innerHTML += "<p>Längden + 5 ger rektangelns area " + area + " m<sup>2</sup></p>";
```

Testa i webbläsaren

- Den nya arean skrivs ut längst ner.

Ange längd och bredd för ytan:

Längd: meter

Bredd: meter

Kör programmet

Resultat

Rektangelns area är 12 m²

Ellipsens area är 9.42 m²

Längden + 5 ger rektangelns area 27 m²

8. Egna tillägg

Nu har du gått igenom ett antal övningar, där du fått koden given. Du har alltså sett i bilderna hur du ska skriva koden. Nu kommer ett par uppgifter, där du inte får koden. Det beskrivs vad du ska göra, men det är upp till dig att avgöra hur det ska göras. Du skriver alltså kod för beräkningarna och skriver ut resultat på webbsidan. Vid behov inför du också nya variabler.

8a. Beräkning av rektangelns area

- Beräkna rektangelns area då längden ökas med 50% och bredden ökas med 3 meter.
- Skriv sedan ut arean tillsammans med lämplig text under de övriga utskrifterna i programmet.

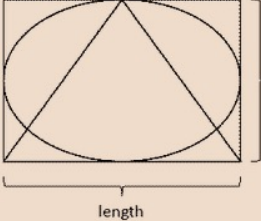
8b. Beräkning av triangelns area

- Beräkna en triangelns area i kvadratfot.
 - Formeln för triangelns area ser du i bilden på webbsidan.
 - 1 meter är 3,28 fot.
- Skriv sedan ut arean tillsammans med lämplig text under de övriga utskrifterna i programmet.

Ange längd och bredd för ytan:

Längd: meter

Bredd: meter



rectangle = length * width
ellipse = $\pi * \text{length} * \text{width} / 4$
triangle = length * width / 2

Resultat

Rektangelns area är 12 m²

Ellipsens area är 9.42 m²

Längden + 5 ger rektangelns area 27 m²

Då längden ökas med 50% och bredden med 3 meter blir rektangelns area 36 m².

Triangelns area blir 64.5504 kvadratfot.

9a. Finn fem fel

Att hitta fel (dvs "debugging") är en viktig del av programmering.

Här ska du öva på att hitta några fel i ett program, som är en del av det program som du själv skapat i föregående övningar, men du kanske inte stötte på de fel som är inlagda här.

- Öppna filen *error.html* i VSC och Live Server.
- Öppna Verktyg för webbutvecklare och fliken Konsol i Firefox.
 - Välj kommandot *Webbläsarverktyg->Verktøy för webbutvecklare* i menyn *Verktøy* i Firefox.
 - En ruta öppnas då i den undre halvan av fönstret. Om inte fliken *Konsol* är vald, klickar du på den.
- Öppna också filen *error.js* i VSC, så att du kan rätta felen där.
- Det första felmeddelande som debuggern ger är på rad 25, kolumn 16. Felmeddelandet är att det är ett oväntat "token", då den träffar på *width*.
 - Felet är dock att det saknas en operator för multiplikation.
 - Rätta felet och ladda om sidan i webbläsaren (den bör laddas om automatiskt av Live Server).
- Det andra felet är på rad 29. Felmeddelandet säger att det förväntades ett uttryck, men det kom *<*. I koden står det *area +* och då förväntar sig webbläsaren att uttrycket ska fortsätta.
 - Felet här är att det första citationstecknet kring *"</p>"* saknas.
 - Rätta felet och kontrollera i webbläsaren.
 - Nu kommer inga nya felmeddelande i debuggern.

I de följande punkterna kallar jag detta verktyg för "debuggern".

*fortsättning på
nästa sida ...*

9b. Finn fem fel

... fortsättning från föregående sida

- Skriv in några värden i textfälten och klicka sedan på knappen. Då får du ett nytt felmeddelande i debuggern. Felet är på rad 23 och felmeddelandet säger att *number* inte är definierat.
 - Felet är att det är felskrivet. Det ska vara *Number* med stort *N*.
 - Rätta felet.
- Kör programmet igen, så får du nästa felmeddelande på rad 28. Felmeddelandet säger att *widht* inte är definierat.
 - Här är det också en felstavning, där *h* och *t* är omkastat. Det ska vara *width*.
 - Rätta felet.
- Kör programmet igen, genom att klicka på knappen. Nu får du inga fler felmeddelanden och utskriften kommer ut.

fortsättning på nästa sida ...

9c. Finn fem fel

... fortsättning från föregående sida

- Men vänta nu! Rubriken på övningen är ju "Finn **fem** fel" och hittills har det ju endast varit fyra fel. Så vad är det femte felet?
- Arean för ellipsen blir alltid 3, oavsett vilka värden som skrivs in i webbläsaren.
 - Felet är alltså i uträkningen av arean på rad 31. Där står det *area = 3,14...* etc.
 - Arean sätts till 3. Sedan kommer ett kommatecken och ett nytt uttryck. Detta är inte språkligt fel, utan man kan räkna ut flera uttryck åtskilda av komma.
 - Man kan t.ex. skriva *a=5, b=7*; för att ge två variabler värden samtidigt.
 - Men felet här är att det inte ska vara komma, utan en punkt, för att få korrekt värde för pi.
 - Flyttal i JavaScript skrivs med decimalpunkt och inte decimalkomma.
 - Rätta felet.
- Kör programmet igen och kontrollera att det nu blir rätt resultat.

Experimentera nu med programmet genom att själv införa fel i det.
Se vilka fel debuggern kan upptäcka och vad felmeddelandena blir.

- Inför några egna fel i programkoden.
- Kontrollera om debuggern ger felmeddelande och vad det säger.
- Rätta felet, ladda om och fortsätt tills du inte får några fler felmeddelanden.