Övningar

[0 Grundläggande övningar](#_4f3odgmd7grm)

[1 Program structure](#_h945kikrm01v)

[A. Fahrenheit 451](#_jlcjw2npxd1m)

[B. Vem är du?](#_2xdzfgeoj6z5)

[C. Gissa numret](#_kd5zchqwn2gg)

[D. Summera](#_rruttx23fhh7)

[E. Vända på en sträng](#_pxlxjbshjgdy)

[F. Antal vokaler](#_hmh39gqqw7yi)

[G. Primtal](#_wasg7b8qlhfv)

[2 Funktioner](#_r8d7btca3qwp)

[A Övningar på repl.it](#_6equxx9w15dt)

[3 Objekt och listor](#_sqofz75o5xx9)

[1 [VG] Statistik för länder (higher order functions för listor)](#_pcts4y1t7j92)

[2 [VG] Rekursiv sorteringsmetod](#_tyutmi4sk49r)

[3 [VG] Länkad lista](#_wtwut2zrvap)

[4 Uppgradera lab 1](#_42c8wux8ympu)

[4 DOM-manipulation](#_v3ypzyvhbuwm)

[1 Blandat](#_uakfy650mad3)

[2 Ny console.log](#_qsk3rz64bh9f)

[3 Manipulera HTML-listor](#_nb3bst8p9hur)

[1 addToHtmlList](#_hfu1pxu1ar05)

[2 reverseHtmlList](#_ak1ieed8rtvl)

[3 VG: moveElement](#_ylblql5u26s2)

[4 VG: createTableFromList](#_9rym33lzjojc)

[5 Events](#_gmswqx3vse34)

# 0 Grundläggande övningar

Här är uppgifter som tränar på att skriva enklare funktioner. Några är mer komplicerade så gå vidare till nästa om du kör fast.

**Skriv en funktion som:**

1. Lägger ihop två tal
2. Multiplicerar tre tal
3. Räknar ut medelvärdet av två tal
4. Räknar ut ränta. Funktionen ska ta två parametrar. Den första ska vara hur mycket pengar som finns på kontot, den andra ska vara räntesatsen. T.ex. motsvarar 1.05 fem procents ränta.
5. Räknar ut hur mycket pengar man har på ett bankkonto efter några år. Funktionen ska ta tre parametrar. Den första ska vara hur mycket pengar det är på kontot från början. Den andra ska vara räntesatsen och den tredje ska vara hur många år man räknar. (Tips: använd funktionen från förra uppgiften)
6. Kapitaliserar ett användarnamn. Man ska till exempel kunna skriva: capitalize(david) och få "DAVID" som returvärde.
7. Avrundar ett tal till två decimaler. Tips: man kan använda Math.round(x) för att avrunda ett tal till närmast heltal.
8. Omvandlar en veckodag till ett nummer. Exempel: veckodagTillNummer('måndag') ska bli 1.
9. Säger hur många dagar en månad har. Funktionen ska ta en parameter, som ska vara en siffra. Siffran 1 motsvarar januari, 2 februari och så vidare. Exempel: howManyDays(3) ska bli 31, eftersom mars har 31 dagar. (Låtsas att februari har 28 dagar)
10. Returnerar de första tre tecknen i en sträng. Exempel: firstThreeChars('Bratislava') ska bli 'Bra'.  
    Använd funktionen substring(startindex, endindex), som plockar ut en del av en sträng. Exempel: 'programmering'.substring(3,7) blir 'gram'.
11. Returnerar det mittersta tecknet i en sträng. Exempel: middleChar('abcde') ska bli 'c' och middleChar('1234') ska bli '3'.
12. Plockar ut året från en sträng i datumformat. Funktionen ska ta en parameter, som ska vara en sträng. Strängen ska alltid ha 10 tecken och följa mönstret 'YYYY-MM-DD'. Man ska kunna skriva year('2016-11-02') och få '2016' som resultat.
13. Skriv en funktion för att plocka ut månaden, och en funktion för att plocka ut dagen, precis som föregående uppgift gjorde för året. Exempel: month('2016-11-02') → 11 och day('2016-11-02') → 2.
14. Räknar ut hur många månader det är kvar till din födelsedag från ett datum. Funktionen ska ta ett datum som parameter.
15. Kontrollerar om en sträng i datumformat är ett riktigt datum. Strängen ska följa mönstret 'YYYY-MM-DD' som funktionerna ovan. Exempel: isProperDate('2016-11-02') ska bli true eftersom det är ett riktigt datum, men isProperDate('2016-13-32') ska bli false.  
    Tips: använd String.substring för att plocka ut en del av en sträng. Använd Number(x) för att omvandla ett värde till datatypen Number.
16. Tar en sträng i datumformat och räknar ut vad nästa dag har för datum.
17. Tar ett veckonummer för 2016 och säger vilket datum veckan börjar och slutar.  
    (Tips: nu är det vecka 44, som började i måndags den 31/10. Använd funktionen från förra övningen.)

# 1 Program structure

## A. Fahrenheit 451

Skriv ett program som översätter temperaturen 451 Fahrenheit till Celsius. Skapa två variabler, en för temperaturen i Fahrenheit och en för Celsius. Skriv ut på konsolen vad resultatet blir. Formeln finns på [den här sidan](http://www.manuelsweb.com/temp.htm).

## B. Vem är du?

Skriv ett program som frågar vad användaren heter, kommer ifrån och favoritfärg. Skriv ut informationen till konsolen i en fullständig mening. Exempel "Välkommen *David* från *Göteborg* med favvofärg *blått*".  
Tips: skriv var x = prompt('Fråga'); för att visa en dialogruta för användaren.

## C. Gissa numret

Skriv ett program som spelar ett spel med användaren. Programmet ska generera ett slumptal mellan 1 och 100. Sedan ska användaren gissa på ett tal. Om talet är större eller mindre än det hemliga så ska programmet skriva ut det. Om användaren gissar fel ska programmet fortsätta fråga efter gissningar, tills användaren gissar rätt. Då ska poängen skrivas ut, som är antalet gissningar.  
Använd funktionen Math.random() som genererar ett slumptal mellan 0 och 1 samt funktionen Math.ceil(nummer) som avrundar ett tal uppåt till närmsta heltal.  
Tips: Math.ceil(Math.random(100)\*100) . (markera för att se tipset)

## D. Summera

Skriv ett program som räknar ut summan av de 100 första heltalen.

## E. Vända på en sträng

Skriv ett program som låter användaren skriva in en sträng och skriver ut vad strängen blir baklänges. Om användaren skriver 'Student' ska programmet skriva ut 'tnedutS' på konsolen. Använd *while* eller *for*.  
Tips: skriv sträng.charAt(X) för att plocka ut bokstaven på plats X i strängen.  
Exempel: 'Student'.charAt(2) == 'u'.

## F. Antal vokaler

Skriv ett program som låter användaren skriva in en sträng och skriver ut hur många vokaler det finns i strängen. (Det räcker med engelska vokaler: a, e, i, o, u, y)

Tips: du kanske kan använda switch?

## G. Primtal

Skriv ett program som frågar användaren om ett tal och kontrollerar om det är ett primtal. Ett primtal är ett heltal som bara är delbart med sig självt och 1. De första primtalen är 2, 3, 5, 7. Men 8 är inget primtal eftersom 8 / 2 == 4.

Obs! Det går att hitta färdiga lösningar på internet. Men om du plockar en färdig lösning så lär du dig mindre än om du försöker göra en själv.

# 2 Funktioner

## A Övningar på repl.it

Gör övningarna på <https://repl.it/classroom/invite/uAFMQT> vars namn börjar med "v2".

# 3 Objekt och listor

Det är blandat med G- och VG-övningar.

## 1 [VG] Statistik för länder (higher order functions för listor)

Använd higher order functions för listor (forEach, filter, map, reduce, find, some och every) för att lösa följande uppgifter. For-loopar är absolut förbjudna!

Börja med att lägga till tre länder till datamängden som finns här: [Länk till data](https://docs.google.com/document/d/1PUAykLaRNDB6EeVA1TJCaHsOgs-_ndaG3B8_gWFt4RY/edit?usp=sharing). Skriv sedan funktioner som tar fram lösningen på följande problem. Jämför dina lösningar med en granne för att se om du har gjort rätt. Se till att ni har samma version av listan, t.ex. genom att jämföra countryStatistics.length innan ni testkör.

1. Skriv ut namnen på alla länder i databasen. Använd forEach eller reduce.
2. Hur många bor i hela (den kända) världen?
3. Hur många bor i Europa?
4. Vilket land har minst befolkning i världen? Minst i Asien?
5. Vad är medelbefolkningen för länder i Afrika?
6. Vilken kontinent har flest invånare?
7. Finns det något land som har mer än 50000000 invånare?
8. Har alla länder i Europa över 1234567 invånare?
9. Vilka länder har mellan 8 och 15 miljoner invånare?
10. Vilken kontinent har flest länder med över 50 miljoner invånare?
11. Av de länder som har minst befolkning i sin världsdel, vilket land har störst befolkning?

## 2 [VG] Rekursiv sorteringsmetod

Skriv en funktion som sorterar en lista rekursivt.

Om en lista har 1 eller färre element, då är den sorterad.

Om en lista har fler än 1 element, dela den på mitten och sortera varje del för sig. Slå sedan ihop de sorterade listorna.

## 3 [VG] Länkad lista

Implementera en *länkad lista* som ett objekt i JavaScript.

En länkad lista är en vanlig datastruktur. Den går ut på att man har ett antal element, som kallas för *noder*. En länkad lista har en *startnod*. Varje nod i listan är ett objekt som har en egenskap *next*, som hänvisar till nästa element i listan. Om en nod är det sista elementet ska egenskapen *next* vara *null*. Dessutom har varje nod en egenskap *value*.

Listan ska ha en egenskap *length* och metoderna *push*, *pop* och *get*. Funktionen motsvarar när man skriver lista[x] för en array, get(x) returnerar värdet som noden på position x i listan har. Implementera den länkade listan utan att använda arrayer.

## 4 Uppgradera lab 1

Ersätt funktionen createMovieObject från Lab 1 med en *konstruktor* vid namn Movie som skapar ett filmobjekt.

Funktionen rateMovie ska vara en metod till Movie. Eftersom den är en metod behövs inte parametern *movie* längre.

Ersätt listan *database* med ett objekt med namn Database. Objekten ska fortfarande lagras i en lista, men listan ska ligga som en egenskap i objektet.

Funktionerna getBestMovie, getWorstMovie, getMoviesThisYear och getMoviesByGenre ska göras om till metoder. Eftersom de är metoder behövs inte parametern *database* längre i någon av dem.

# 4 DOM-manipulation

## 1 Blandat

<https://lejonmanen.github.io/dom.html>

Kom ihåg att du kan få fram HTML-koden för en webbsida genom att högerklicka och välja "visa källa" / "view source".

## 2 Ny console.log

Nu när vi börjar med DOM-manipulation vill vi inte använda konsolen så mycket. Men vi vill fortfarande kunna skriva ut vad som händer i programmet. Skapa en funktion som gör samma sak som console.log, fast skriver ut resultatet till ett HTML-element i stället.

Funktionen ska förutsätta att det finns ett HTML-element med id="console" i dokumentet.

Inuti det elementet ska du lägga till ett <pre> element. Det ska ha svart bakgrund och ljusgrön textfärg. Använd JavaScript för att sätta färgerna. Varje gång användaren anropar din funktion ska du lägga till en ny rad med text i elementet.

Tips: använd '\n' för att lägga till en radbrytning i ett <pre> element.

## 3 Manipulera HTML-listor

### 1 addToHtmlList

Skriv en funktion som lägger till saker till en HTML-lista. (ul eller ol) Funktionen ska ha listans id som parameter.

Exempel: addToHtmlList('listId', 'text');

Ska bli: <ul id='listId'> <!-- gamla element finns kvar --> <li>text</li> </ul>

### 2 reverseHtmlList

Skriv en funktion som vänder upp och ner på en HTML-lista. Använd removeChild och appendChild eller replaceChild.

Exempel: reverseHtmlList('exempel');

<ul id="exempel"> <li>1</li> <li>2</li> <li>3</li> </ul>

ska ändras till:

<ul id="exempel"> <li>3</li> <li>2</li> <li>1</li> </ul>

### 3 VG: moveElement

Skriv en funktion som flyttar det första elementet i en lista och lägger det sist i en annan lista.

Exempel: moveElement(originListId, destinationListId)

Tips: element.firstChild är en property som ger det första barnet till ett element.

### 4 VG: createTableFromList

Skriv en funktion som lägger in en lista med text (eller HTML) i en tabell.

createTableFromList(parentElementId, itemsPerRow, list)

Testa funktionerna med följande kod.

HTML:

<body id="body">

<ul id="first"></ul>

<ul id="second"></ul>

</body>

JavaScript:

addToHtmlList('first', 'apa');

addToHtmlList('first', 'björn');

addToHtmlList('first', 'delfin');

addToHtmlList('second', 'clementin');

addToHtmlList('second', 'dadel');

addToHtmlList('second', 'fikon');

reverseHtmlList('first');

moveElement('first', 'second');

createTableFromList('body', 3, ['wild', 'words', 'went', 'wilfully', 'wherever', 'wily', 'whales', 'would', 'waste', 'water']); // 10 elements

# 5 Events

Från och med nu arbetar vi med sådant som vi kan vilja visa upp. Lägg gärna upp dina svar på övningarna på din Github-sida.

Du kan använda JS Bin när du utvecklar sidorna.

1. Skriv HTML och JavaScript som visar en button. När man klickar på den ska du skriva ut till en div vad event.type och event.target är. Använd JS Bin eller gör en egen sida.
2. Gör en sida med två knappar. På den ena ska det stå "Start" och på den andra "Stopp". Stopp-knappen ska vara inaktiverad, använd element.disabled=true  
   När man klickar på Start ska start-knappen inaktiveras och stopp-knappen aktiveras. När man klickar på Stopp ska det bli tvärtom.
3. Gör en sida med flikar. Ett system med flikar är när man har två eller flera div-element som finns på samma plats på sidan, men bara ett är synligt åt gången. Det ska finnas knappar eller andra element, ett per div, som man klickar på för att göra motsvarande element synligt. Använd element.style.display för att göra saker synliga och osynliga.
4. Gör en sida med element som ändrar för- eller bakgrundsfärg när musen rör sig över dem.
5. VG: Gör ett element som ändrar storlek på texten inuti sig när muspekaren befinner sig över elementet, utan att hela elementet tar större plats.
6. Gör en sida med ett textfält. Skriv ut i en div vilka knappar användaren håller nere.
7. Gör ett textfält som validerar tal. När textfältet tappar fokus ska du kontrollera om texten i textfältet är ett giltigt tal. Du ska skriva ut resultatet i anslutning till textfältet, med grön text om det är ett giltigt tal och röd text annars.
8. Gör ett textfält som gör om texten medan man skriver, så att den gör om små bokstäver till stora. (se övning 0.6 ovan)