



Inlämningsuppgift 2

Betyg: U/G

Syfte

Syftet med den andra inlämningen är att fortsätta göra sig bekant med både programmeringsmiljön men ännu mer med självaste programmeringen. I denna inlämning kommer ni att arbeta med koncept som är viktiga för att kunna strukturera ett program - funktioner. Dessa används för att kunna dela in kod i logiska delar samt för att kunna skapa delar som är återanvändbara, vilket är en väldigt viktig del av programmering. Utöver detta så blir det fortsatt repetition av det som tidigare introducerats i den första inlämningen.

Inför denna uppgiften rekommenderas det starkt att läsa kapitel 3, 4 och 5 i boken Eloquent JavaScript.

Obs där det står “returnerar” (eller dylikt) så menas detta att ni använder er av

`return` i funktionen, dvs. **inte** `console.log()`

Uppgiften

Denna uppgift är indelad i mindre deluppgifter, det rekommenderas att försöka hålla sig till den ordning som presenteras (då svårighetsgraden ökar med varje).

Uppgifterna kan lösas på många olika vis, det finns alltså inte alltid ett korrekt svar utan ibland kan det finnas ett flertal. När ni söker information via webben efter lösningar eller hjälp på vägen så rekommenderas det starkt att oavsett vad ni hittar - försök att förstå hur det fungerar - experimentera och utforska koden med hjälp av webbkonsollen.

Tips 1: testa gärna att klistra in er kod på **JSHint** för att se om den kan ge förslag på förbättringar. Det är en sida som följer vissa standarder för hur en bör skriva kod på ett korrekt vis.

Tips 2: testa gärna att använda er av tjänsten **Dirty Markup** som hjälper er formatera er kod.



Glöm inte att läsa igenom kravspecifikation innan ni börjar, så ni inte missar någonting eller får komplettera på grund av slarvfel.

Uppgift 1

Skapa funktionerna `max` och `min` som returnerar det högsta respektive lägsta värdet av två parametrar. Det är **inte** tillåtet att använda sig av `Math.max` och `Math.min`. Ta en titt på kodexemplet nedan för att få mer information om användningen av dessa funktioner:

```
max(5, 10); // hade skickat tillbaka 10
max(7, 7); // hade skickat tillbaka 7
min(12, 24); // hade skickat tillbaka 12
min(30, 18); // hade skickat tillbaka 18
```

För att pröva era funktioner kan ni manuellt skriva in dem i webbkonsollen (förutsatt att ni skapat dom i er JavaScript-fil) - t.ex. `max(5, 12);`.

Uppgift 2

Skapa funktionen `range` som returnerar en array innehållande siffrorna `0` till `n` där `n` är den parameter som skickades med, det vill säga att anropet `range(5);` hade gett oss arrayen `[0, 1, 2, 3, 4]`. För att pröva er kod kan ni testa kodexemplet nedan när ni skapat er funktion:

```
var testArray = range(10);
console.log(testArray);
```

Tips: tänk på att i funktionen kommer ni behöva använda er av en `for`-loop i kombination med parametern `n`.

Uppgift 3

Skapa funktionen `sum` som returnerar summan av alla siffror i en array. Det vill säga att anropet `sum([5, 10, 15]);` hade skickat tillbaka siffran `30`. För att pröva er funktion kan ni utgå från kodexemplet nedan:



```
var numbers = [5, 10, 15, 20, 25];  
var sumOfNumbers = sum(numbers);  
console.log(sumOfNumbers);
```

Tips: för att öka en variabel (lägga till) med siffror kan ni göra på följande vis:

```
// Börja med värdet 10  
var n = 10;  
// Lägg till 10  
n += 10;  
// Lägg till 5 (samma som ovan fast skrivet på ett alternativt v  
n = n + 5;  
  
console.log(n);
```

Uppgift 4

Skapa funktionen `countCharacter` som räknar antal upprepningar av en bokstav från en sträng. Det vill säga att anropet `countCharacter("Jane Doe", "e");` hade skickat tillbaka siffran `2` för det finns två stycken `"e"` i `"Jane Doe"`. För att pröva att er funktion fungerar kan ni utgå från kodexemplet nedan:

```
var testString1 = "Jane Doe";  
var testString2 = "Abracadabra";  
  
console.log(countCharacter(testString1, "e")); // => 2  
console.log(countCharacter(testString2, "a")); // => 4 (obs. lit
```

Tips: att gå igenom bokstäver från en sträng kan göras på samma vis som element i en array, det vill säga:

```
// Vår sträng  
var exampleName = "John Doe";  
// Gå igenom varje bokstav från strängen  
for (var i = 0; i < exampleName.length; i++) {  
    // Skriv ut en bokstav i taget
```



```
        console.log(exampleName[i]);  
    }  
}
```

Uppgift 5

Skapa funktionen `palindrome` som kontrollerar om en sträng är det samma som den är baklänges, om så är fallet ska denna funktion returnera värdet `true` annars `false`. Ta en titt på kodexemplet neda för att få mer information om hur funktionen ska fungera:

```
palindrome("sirap - paris"); // skickar tillbaka "true"  
palindrome("lorem ipsum"); // skickar tillbaka "false"
```

Tips: skapa originalsträngen baklänges och jämför den sedan med originalet så vet vi om strängen är ett palindrom.

Uppgift 6

Skapa ett objekt och spara det i variabeln `person`. Detta objekt ska ha följande attribut med respektive värde:

- `firstname`, en sträng innehållande ert förnamn
- `lastname`, en sträng innehållande ert efternamn
- `age`, en siffra innehållande er ålder
- `family`, en array innehållande namn på era familjemedlemmar (vilka och hur många ni väljer är valfritt)

Uppgift 7

Skapa funktionen `printPerson` som tar emot ett objekt som parameter och skriver ut dess attribut (formatet på utskriften får ni bestämma själva). Ni kan utgå från att objektet är strukturerat/innehåller samma attribut som det i uppgift 6. För att förstå mer hur det är tänkt att fungera kan ni ta en titt på kodexemplet nedan:

```
var person1 = { /* alla attribut */ };  
  
// Anropet till printPerson() hade kunnat ge följande utskrift  
printPerson(person1);
```



```
// "Fullname: Jane Doe, Age: 25"
// "Family: John, Eliza, Elise"
```

Uppgift 8

Skapa funktionen `createBox` som tar emot två parametrar (`height` och `width`) och returnerar ett objekt innehållande attributen `height` och `width` . För att förstå mer hur det är tänkt att fungera kan ni ta en titt på kodexemplet nedan:

```
var box = createBox(15, 20);
// Skriv ut attributen
console.log(box.height); // skickar tillbaka 15
console.log(box.width); // skickar tillbaka 20
```

Uppgift 9

Skapa funktionen `Triangle` som tar emot två parametrar (`height` och `width`) och returnerar ett objekt innehållande attributen `height` och `width` samt en metod (funktion) med namnet `area` . När denna metoden anropas ska arean räknas ut och returneras (för att räkna ut arean kan ni använda formeln `height * width / 2`). För att förstå mer hur det är tänkt att fungera kan ni ta en titt på kodexemplet nedan:

```
var tri = Triangle(12, 14);

console.log(tri.height); // skickar tillbaka 12
console.log(tri.width); // skickar tillbaka 14
// Observera att vi anropar "area()"
console.log(tri.area()); // skickar tillbaka 84
```

Det rekommenderas **starkt** att ni använder er av nyckelordet `this` inuti metoden `area` för att räkna ut triangelns area.

Uppgift 10

Skapa funktionen `attributes` som tar emot ett objekt som en parameter och returnerar en array innehållande alla namnen på objektets attribut. Ta en titt på



kodexemplet nedan för att förstå vad som menas med detta:

```
var testObject1 = {
  width: 15,
  height: 20
};

var attrsFromObject1 = attributes(testObject1);
console.log(attrsFromObject1); // skickar tillbaka ["width", "he

var testObject2 = {
  a: 1,
  b: 2,
  c: 3
};

var attrsFromObject2 = attributes(testObject2);
console.log(attrsFromObject2); // skickar tillbaka ["a", "b", "c
```

Tips: använd er av loopen `for (var attribute in obj) { ... }` för att gå igenom alla attributen.

Kravspecifikation

- På första raden i er fil ska ni använda raden `"use strict";` för att göra webbläsaren mer strikt i sin tolkning av er kod
- Samtliga uppgifter ska markeras med en kommentar innan uppgiften innehållande uppgiftsnummret, dvs. `// Uppgift 1.` för första uppgiften osv.
- Det får inte visas några felmeddelande i webbkonsollen
- All JavaScript-kod ska placeras i en egen fil och länkas in till ert HTML-dokument
- All JavaScript-kod måste ha godtycklig indentering, dvs. allting får inte vara vänsterställt eller allt för svårt att läsa

Inlämning

Glöm inte kontrollera att ni skickat med svar på alla uppgifter och att ni följt kravspecifikationen.

När du är färdig med din uppgift ska du ladda upp denna som en `.zip`-fil innehållande alla dina filer på Canvas (på samma sätt som inlämningsuppgift 1). Döp



denna enligt formatet `inl2_Förnamn_Efternamn.zip` . Du ska även ladda upp dessa filer på dvwebb och därefter ska du även inkludera länken dit.

Lycka till!

Detta projekt är publicerat som **öppen källkod** på **GitHub**.

Användning av allt material är fritt att tillgå genom **The MIT License (MIT)**.