

# Övningar - Funktioner fortsättning

## Introduktion

Som så mycket annat i programmering kan allting göras på flera olika sätt. Funktioner kan även skapas på flera olika sätt. Ingen av sätten är “rätt” men de fungerar på lite olika sätt och ett av dessa sätt föredras av många JavaScript-användare:

```
var myFunc = function(){  
    //Do Stuff  
};
```

Sättet vi skapar en `function` här är genom att skapa en anonym funktion och lagrar den funktionen i en variabel.

Skillnaden här blir dock att den funktionen ovan följer kodens flöde (uppfifrån och ner). Vi kan inte kalla på funktionen innan vi har skapat den. Men om vi deklarerar på det tidigare sättet flyttas funktionen högst upp i vår kod (Inom global scope):

```
function myFunc(){  
    //Do Stuff  
}
```

Detta gör så att vi kan kalla på funktionen innan den är skriven i vår kod. Det finns olika användningsområden för båda sätten som vi kommer gå in på senare under kursens gång.

## The Horror

Nedan är en helt onödig funktion som `console.loggar` “The Horror! The Horror!”. Varför denna funktion finns kommer vi aldrig få veta men vi ska nu använda den till denna övning.

```
function theHorror() {  
    console.log('The horror! The Horror!');  
}
```

1. Du ska göra om funktionen till en anonym funktion som sparas i en variabel som heter `theHorror`.
2. Testa att kalla på funktionen både före och efter den är deklarerad och se vilken av anropen som alertar något. (Kolla längst till höger i konsolen för att se vilken rad som körs i din kod)
3. Du ska sedan även göra om funktionen till en anonym funktion som kallar på sig själv när din sida laddas. (Kolla genomgången för tips)
4. Hur skulle vi kunna skicka med ett argument in i den anonyma funktionen?

## Even and Odd

Använd while-loopen du skapade från i tisdags enligt uppgiften:

Skriv en while-loop med ett condition som gör så att endast siffror som är jämna tal skrivs ut till konsolen.

Denna loop ska du nu göra om till en funktion som tar 2 parametrar. \* Första parametern ska vara siffran som loopen ska räkna ner ifrån, alltså hur många värden funktionen ska gå igenom. \* Andra parametern ska vara om funktionen ska skriva ut jämna eller ojämna värden (even or odd)

## Calcy

Skriv en funktion som tar 2 parametrar. Parametrarna ska vara två heltal. Funktionen ska multiplicera heltalen utan att använda \*-operatören.

## String checker

Till uppgifterna nedan kan det vara bra att kolla igenom: **String Methods @ w3Schools.com**

Skriv en funktion som tar in en parameter. Parametern ska vara en string. Funktionen ska sedan returnera strängens längd på detta sätt:

```
"Strängen du matade in är 14 tecken lång"
```

## String converter

Skapa en funktion som heter `convertString`, funktionen ska ta två parametrar. Den första parametern ska vara en sträng som skickas med, typ: "Goodbye World". Den andra parametern som skickas med ska bestämma om strängen ska konverteras till bara stora bokstäver eller bara små bokstäver.

## String extractor

Skapa en funktion som tar en parameter, argumentet som skickas in ska vara en sträng. Funktionen ska sedan returnera det sista tecknet i strängen som skickas in. (substr samt length kan användas här)