

## Funktioner

Om du har glömt bort ser en funktion ut på detta sätt:

```
function functionName(parameter){  
    return parameter;  
}
```

Det som är innanför paranteserna är en **parametrar**. De parametrar som skickas med kan användas inom funktionen utan att deklarerats och vi kan använda parametern inom vår funktion. Funktionen är dock bara HUR programmet ska funka och funktionen gör ingenting förrän vi kallar på funktionen och skickar med värden. Detta betyder att vi kan skapa kod som KAN användas men används bara när vi bestämmer oss för att kalla på koden.

Om vi ska skicka med värden skickar vi med dem som **argument**. De argument vi skickar med till funktionen blir funktionens parametrar. Om vi ska kalla på en funktion samt skicka med ett argument gör vi på följande sätt:

```
function functionName(parameter){  
    console.log(parameter)  
}  
  
functionName('Jesper');  
//Output: Jesper
```

## Multiply

1. Skapa en namngiven funktion: multiply, som tar in 2 *parametrar*. I funktionen, multiplicera dessa nummer med varandra och console.loggar resultatet av multiplikationen. Du får använda vilka namn du vill på parametrarna.
2. Kalla på din funktion multiply med siffrorna 8,4 som *argument*

Om du gjort rätt ska du få siffran 32.

## Calculate

1. Skapa en funktion som heter calculate och som istället tar 3 parametrar och sedan multiplicerar de två första parametrarna med varandra och delar värdet med den tredje parametern, alltså: `param1 * param2 / param3`.
2. Console.logga svaret
3. Kalla sedan på din funktion med valfria värden.

## Maxing, relaxing, acting all cool

1. Skriv en funktion som heter **highestNumber** som tar två tal som parametrar.
2. Funktionen ska sedan jämföra vilket av talet som är störst och console.logga det största talet.
3. Kalla på din funktion med två valfria värden.

## Maxing, not relaxing

1. Använd funktionen som du skapade i förra övningen.
2. Båda parametrarna kan dock vara samma värde. Lägg till kod som `console.log`gar ut “Samma värde” om båda parametrarna är likadana.
3. Vi kan även skicka med andra värden än t.ex. heltal. Vi skulle kunna skicka med en sträng, detta vill vi dock inte. Se till så att din funktion `console.log`gar “Inte ett nummer” om något av värdena är något annat än ett nummer. Detta kan man göra med den inbyggda funktionen: `isNaN()`

## Refaktorera

Koden ovan är dock inte optimal. För det mesta vill vi att funktioner endast returnerar värden. En funktions uppgift ska ju vara att returnera värden så att vi kan återanvända vår kod. När vi har massa `console.log` på det här sättet blir koden svårare att återanvända så vi vill nu skriva om våra funktioner så att de endast returnerar värden.

**Refaktorera** dina funktioner som du tidigare skapade:

- De två första funktionerna (`multiply` och `calculate`) ska **returnera** det slutgiltiga värdet av beräkningarna.
- Den tredje funktionen `highestNumber` ska **returnera** det högsta värdet av de två värden som skickas in som parametrar. Om dock värdet inte är ett nummer ska funktionen returnera `false` och om de båda parametrarna är samma värde ska funktionen returnera “*Samma värde*”

Spara värdena som returneras på följande sätt:

```
var multiplyAnswer = multiply(5,5);
```

Samt `console.log`ga resultaten.

## Mjau Machine Reborn

Eftersom **alla** gillar katter så ska vi refaktorera vår Mjau Machine.

Du ska skriva en funktion som ska ha samma funktionalitet som vår ursprungliga Mjau Machine men nu i form av en funktion.

Den ska ta antalet mjau som en parameter och köras tills `mjau == 0`.