**Funktioner och objekt-orientering i Javascript**

**Metoder**

En funktion är en bit kod som ofta utför en specifik uppgift.

En fördel med en funktion är att den kan användas flera gånger.

**Enkelt exempel:**

**function** plussaTal **()**

**{**

**var** tal **=** tal1 **+** tal2**);**

**document.**write**(**"Talet är " **+** tal**);**

**}**

För att det ska hända som sker i funktionen måste man anropa den.

plussaTal**();**

**Invärde**

En funktion kan också ha invärden som man skickar till funktionen. I exemplet ovan blir ju talet alltid 15. Ofta skickar man ett värde till en funktion.

Man kan se det som en maskin där man matar in t.ex. siffror för att sedan få ett resultat.

plussaTal**(**5**,** 20**);**

**function** plussaTal **(**tal1**,** tal2**)**

**{**

**var** tal **=** tal1 **+** tal2**);**

**document.**write**(**"Talet är " **+** tal**);**

**}**

**Returvärde**

Ibland vill man skicka tillbaka ett värde från en funktion istället för att t.ex. skriva ut något i funktionen. När värdet kommer tillbaka lägger man det t.ex. i en variabel och

Det kan t.ex. vara en funktion som gör en komplicerad uträkning. Sedan kan man själv välja vad man ska göra med resultatet, t.ex. skriva ut det eller använda det i en formel.

alert("summan blir” + plussaTal**(**5**,** 20**));**

**function** plussaTal **(**tal1**,** tal2**)**

**{**

**var** tal **=** tal1 **+** tal2**);**

**document.**write**(**"Talet är " **+** tal**);**

**}**

**Objekt och objektorientering**

Nästan allt i Javascript är objekt men det som brukar kallas objekt är ett sätt att samla olika sorters data.

När man kör ett program uppifrån och ner bruka detta kallas för procedurell programmering.

För att göra programmeringen lättare uppfann man ett begrepp som kallas för Objektorienterad programmering.

Istället för att köra koden uppifrån och ner skapar man objekt i minnet som man sedan hanterar.

Objekt är ofta en sak som anställd, bok eller liknande.

Man brukar använda begreppet **klass** som är som en mall för hur objekten ska se ut.

Begreppet klass finns inte i Javascript som egentligen är funktionsorienterat istället för objektorienterat.

Man kan dock använda objektorientering även i Javascript.

Klassen (I Javascript objektet) har sedan egenskaper (properties)

**Märke Färg Typ**

Saab Grön Sedan

Ferrari Röd Sportbil

Range Rover Svart Jeep

**Märke**: Saab

**Modell**: 900

**Färg**: Blå

Märke

Modell

Färg

Märke

Färg

Typ

**Märke**: Ferrari

**Modell**: F40

**Färg**: Röd

**Märke**: Bugatti

**Modell**: Veyron

**Färg:** Svart

Vi skapar objektet (som är mallen).

Så en **”klass”** i Javascript kallas objekt.

**function** Employee **()**

**{**

**this.**name**;**

**this.**profession**;**

**}**

Sedan en instans av det objektet (vad som i andra språk kallas just objekt)

**var** person **=** **new** Employee**();**

person**.**name **=** "Kalle"**;**

**document.**write**(**"Den nye heter " **+** person**.**name**);**

Man kan även som i andra språk ha en s.k. construktor. Det innebär att man sätter värden direkt när objektet skapas.

Vi skapar först en funktion (objekt) där vi kan sätta värden.

**function** Employee **(**name**,** profession**)**

**{**

**this.**name **=** name"**;**

**this.**profession **=** profession**;**

**}**

När vi skapar en instans (objekt) av klassen sätter vi värden direkt. Man konstruerar hela objektet direkt.

**var** person **=** **new** Employee**(**"Kalle"**,** "Utvecklare"**);**

**document.**write**(**"Den nye heter " **+** person**.**name**);**