

# JavaScript

## Programming Language



## WEBBPROGRAMMERING MODUL 4 - JAVASCRIPT

Mahmud Al Hakim  
[mahmud@alhakim.se](mailto:mahmud@alhakim.se)

# FRONTEND TEKNOLOGIER

**HTML**



**CSS**



**JS**



HTML  
**Struktur**

CSS  
**Utseende**

JavaScript  
**Beteende**

# VAD ÄR JAVASCRIPT?

JavaScript är ett skriptspråk som används tillsammans med HTML och CSS för att lägga till beteende och interaktivitet på webben.

Koden körs direkt i webbläsaren och behöver inte kompileras som till exempel är fallet för andra programspråk såsom C# och Java.

Tolkning av JavaScript-kod sker i besökarens webbläsare till skillnad från skriptspråk som körs på serversidan t.ex. PHP.

# LITE HISTORIA

JavaScript såg dagens ljus december 1995 i samband med att Netscape släppte sin webbläsare Netscape Navigator 2 som stödde skriptspråket **Livescript**.

Livescript utvecklades av Brendan Eich, som var anställd vid Netscape och sedan grundade Mozilla år 1998.

Efter ett samarbete med Sun (som utvecklade Java) ändrades namnet till JavaScript. Kort efteråt kom Microsoft med sin variant som kallades **Jscript**.

JScript och Javascript skilde sig markant på inbyggda objekt och funktioner vilket ledde fram till behovet av en gemensam standard för skriptspråket, standarden kom att kallas för ECMA-262, även känt som **Ecmascript**.



# ETT FÖRSTA EXEMPEL

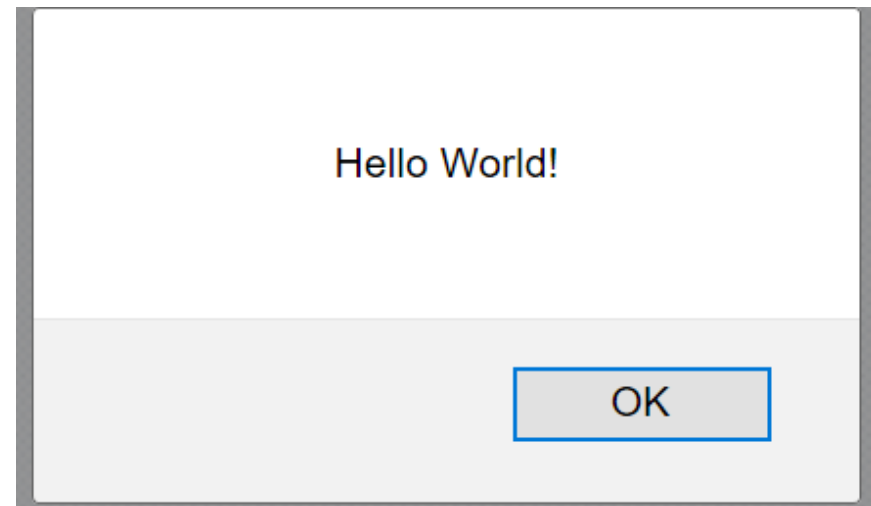
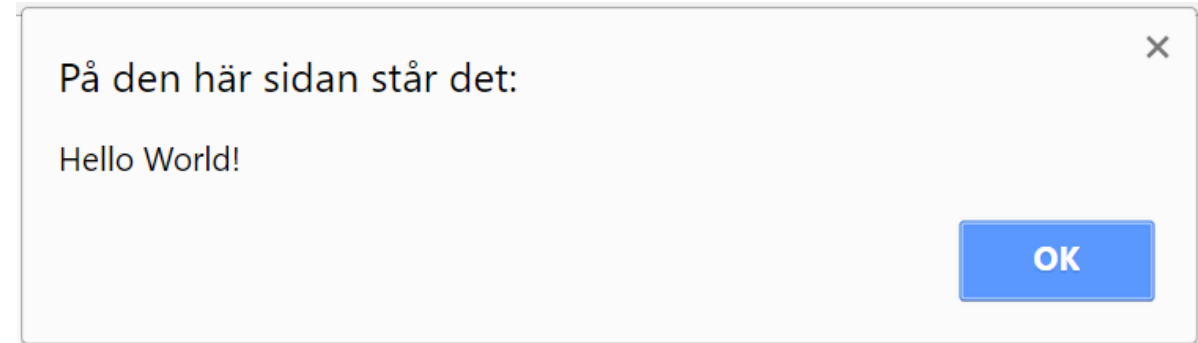
`alert()` är en JavaScript funktion (metod) som gör att en varningsruta poppar upp på skärmen.

Besökaren måste klicka på "OK"-knappen på för att fortsätta.

Typiskt användningsområde är om du vill försäkra dig att information kommit fram till besökaren.

# HELLO WORLD!

```
<html>
<head>
<title>Exempel 1</title>
</head>
<body>
<script>
    alert("Hello World!");
</script>
</body>
</html>
```



# EXTERNA JAVASCRIPT FILER

Externa JavaScript-filer kan användas av flera dokument.

JavaScript-kod skrivs en separat fil med filnamnstillägget .js  
(t.ex. script.js)

Script-taggen med attributet src infogar skriptet i HTML-dokumentet:

```
<script src="script.js"></script>
```

OBS! Man får inte länka till externa JavaScript-filer och skriva kod direkt i script-taggen samtidigt.

# EXTERN JAVASCRIPT-FIL

exempel-02.html

```
<html>
<head>
<title>Exempel 2</title>
</head>
<body>
<script src="script.js"></script>
</body>
</html>
```

script.js

```
alert("Hello World!");
```



# SATSER (STATEMENTS)

En sats (statement) är ett uttryck som följer de skrivregler (**syntax**) som finns för språket.

En sats i JavaScript avslutas vanligtvis med ett **semikolon ;**

**Semikolon** talar om för JS-tolken att satsen är slut;

Exempel på en sats

```
alert("Hello World!");
```

# OBS! JAVASCRIPT ÄR CASE SENSITIVE

JavaScript är Case Sensitive (skiftlägeskänsligt)  
d.v.s. språket är känslig för stora och små bokstäver!

Exempel:

```
Alert("Hello World!");
```

```
✖ Uncaught ReferenceError:  
Alert is not defined  
at javascript.js:1
```

# JAVASCRIPT KOMMENTARER

```
<script>
```

```
// Detta är en kommentar i JavaScript
```

```
/* Detta är kommentar  
över en eller flera rader  
*/
```

```
</script>
```

# JAVASCRIPT ÄR ETT OBJEKTORIENTERAT PROGRAMMERINGSSPRÅK!

**Det finns ett antal inbyggda objekt i JavaScript**

Exempel:

Objektet document innehåller all information om den nedladdade sidan.

**Objekt har metoder (methods) som utför något på själva objektet**

Exempel:

Metoden getElementById() hämtar ett element med hjälp av attributet id som kan kopplas till alla HTML-Element. OBS! Värdet på id måste vara unikt.

**Objekt har även olika egenskaper (properties)**

Exempel:

Egenskapen innerHTML hämtar eller sätter innehållet i ett HTML-element.

# METODEN getElementById() och EGENSKAPEN InnerHTML

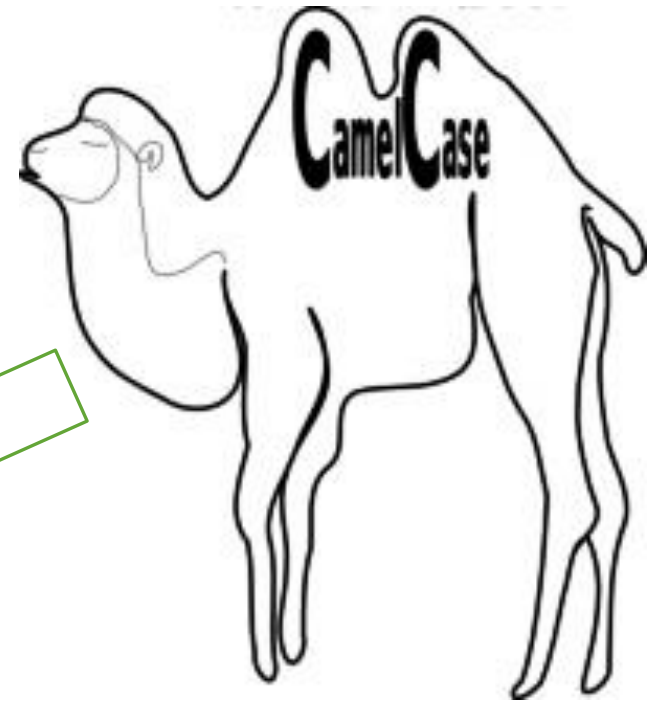
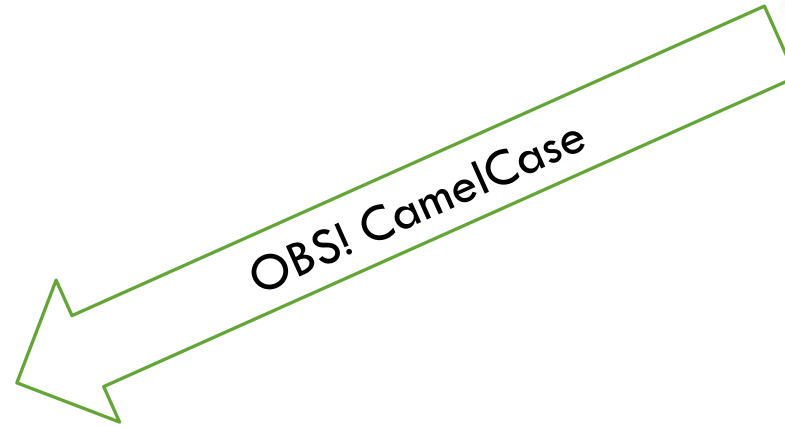
```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML =
```

```
"JavaScript kan manipulera innehåll";
```

```
</script>
```



# VARIABLER

En variabel är en platshållare för ett värde som ändras under programmens gång.

Variabelnamn måste börja med en bokstav, understreck eller dollartecken.

Exempel

```
var namn;
```

```
var $;
```

Variabler deklarerar  
med nyckelordet **var**

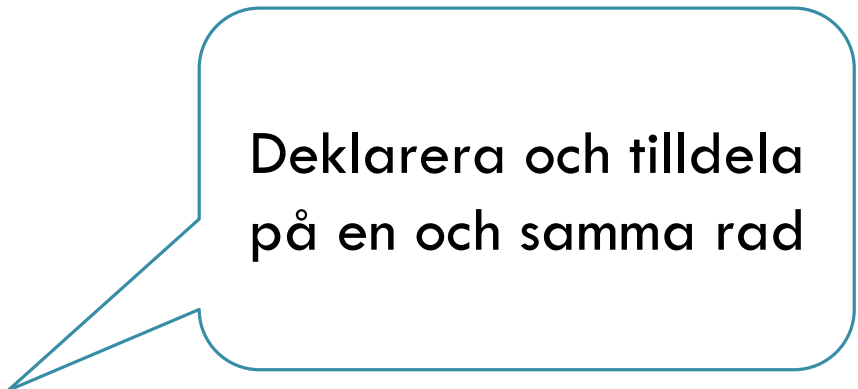
**namn** och **\$**  
är giltiga variabelnamn

# DEKLARERA OCH TILLDELA VARIABLER

Variabler i JavaScript deklareras med nyckelordet **var**

En variable tilldelas ett värde med tecknet "lika med" =

```
<script>  
var fornamn;  
fornamn = "Mahmud";  
var efternamn = "Al Hakim";  
</script>
```



Deklarera och tilldela  
på en och samma rad

# NYCKELORD I JAVASCRIPT

I JavaScript finns det ett antal reserverade ord (nyckelord), som inte får användas för t.ex. variabelnamn.

Exempel på reserverade ord i JavaScript  
var, this, true, false, function, return, for

Tips!

Här finns en lista på alla nyckelord

[http://www.w3schools.com/js/js\\_reserved.asp](http://www.w3schools.com/js/js_reserved.asp)



# DATATYPER I JAVASCRIPT

**Object:** Objekt med tillhörande egenskaper och metoder.

**String:** Text (sträng) som skrivs innanför citationstecken.

**Number:** Ett positivt eller negativt heltal eller ett flyttal (decimaltal).

**Boolean:** kan anta värden true (sant) och false (falskt).

**Undefined:** betyder ej definierad.

**Null:** betyder saknar ett värde just nu (men har definierats förut).

# VARIABLER OCH DATATYPER

```
<p id="total"></p>
```

```
<script>
```

```
var pris = 95.50;
```

```
var antal = 5;
```

```
var total = pris * antal;
```

```
var el = document.getElementById("total");
```

```
el.innerHTML = total.toLocaleString() + "kr";
```

```
</script>
```

# KONSTANTER

**const** skapar konstanter (single-assignment)

```
const x = "foo";  
x = "bar";           // Fel! x är en konstant
```

✖ ▶ Uncaught TypeError: Assignment  
to constant variable.  
at test.html:3

# ARBETA MED FÄLT I JAVASCRIPT (ARRAYS)

<h2>En array kan lagra flera värden i en enda variabel</h2>

<p id="fname"></p><p id="lname"></p>

<p id="person"></p><p id="person2"></p>

<script>

var person = ["Mahmud", "Al Hakim", 1973];

document.getElementById("fname").innerHTML = person[0];

document.getElementById("lname").innerHTML = person[1];

document.getElementById("person").innerHTML = person;

document.getElementById("person2").innerHTML = person.join(" ");

</script>

# ARITMETISKA (MATEMATISKA) OPERATORER

Operator	Benämning
+	Plus
-	Minus
*	Multiplikation
/	Division
%	Modulus (restoperator)

# JÄMFÖRELSEOPERATORER

Operator	Benämning
<	mindre än
>	större än
<=	mindre än eller lika med
>=	större än eller lika med
==	lika med
!=	ej lika med

# LOGISKA OPERATORER

Operator	Benämning
!	NOT (inte)
&&	AND (och)
	OR (eller)

# ENSIDIGA OPERATORER

## Ökningsoperatorn ++

Addera ett med hjälp av operatorn ++

```
var a=1;
```

```
a++;
```

## Minskingsoperatorn --

Minska ett med hjälp av operatorn --

```
var b=1;
```

```
b--;
```



# OPERATORER - EXEMPEL

```
<p id="demo"></p>
```

```
<script>
```

```
var a = 10, b = 10;
```

```
var el = document.getElementById("demo");
```

```
el.innerHTML = "a < b = " + (a < b) + "<br>";
```

```
el.innerHTML += "a > b = " + (a > b) + "<br>";
```

```
el.innerHTML += "a == b = " + (a == b) + "<br>";
```

```
el.innerHTML += "a <= b = " + (a <= b) + "<br>";
```

```
el.innerHTML += "a >= b = " + (a >= b) + "<br>";
```

```
</script>
```

# STRÄNGHANTERING

Med hjälp av objektet String kan du manipulera och hantera text i JavaScript

Du kan t.ex. beräkna antal tecken i en sträng, omvandla text till gemener eller versaler.

Du kan även omvandla en sträng till en array

Tips

[http://www.w3schools.com/jsref/jsref\\_obj\\_string.asp](http://www.w3schools.com/jsref/jsref_obj_string.asp)

# STRÄNGHANTERING — EXEMPEL

```
<p id="demo"></p>
```

```
<script>
```

```
var name = "Mahmud Al Hakim";
```

```
var el = document.getElementById('demo');
```

```
el.innerHTML = "Antal tecken: " + name.length;
```

```
el.innerHTML += "<br>Konvertera till versaler: " + name.toUpperCase();
```

```
el.innerHTML += "<br>Konvertera till gemener: " + name.toLowerCase();
```

```
</script>
```

# OMVANDLA EN STRÄNG TILL EN ARRAY

```
<p id="demo"></p>
<script>
var name = "Mahmud Al Hakim";
var array = name.split(" ");
var el = document.getElementById('demo');
el.innerHTML =array[0];
</script>
```

# ESCAPE-TECKEN

Escape-tecken kallas ibland för skiftningstecken och i vissa sammanhang för jokertecken. De ser olika ut och används lite olika, beroende på vad det är man vill göra.

- \ " Visar dubbelt citationstecken
- \ ' Visar enkelt citationstecken
- \\ Visar tecknet backslash

# ESCAPE-TECKEN — EXEMPEL

```
var x = 'It\'s alright <br>';  
var y = "We are the so-called \"Vikings\" from the north."  
var el = document.getElementById('demo');  
el.innerHTML = x + y;
```

## Tips

[https://www.w3schools.com/js/js\\_strings.asp](https://www.w3schools.com/js/js_strings.asp)

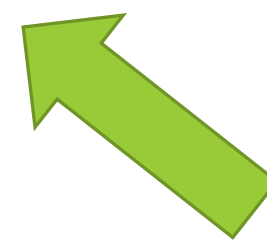
# STRING INTERPOLATION

```
var name = "Mahmud";
```

```
var text = `Hej ${name}! <br>
```

JavaScript, as of the ECMAScript 2015 ES6 standard,  
supports string interpolation using backticks

This feature is called template literals`



*Backtick*

Tips!

[https://en.wikipedia.org/wiki/String\\_interpolation#JavaScript](https://en.wikipedia.org/wiki/String_interpolation#JavaScript)

# FUNKTIONER

Funktioner är ett bra sätt att ordna kod i olika separata block.

Funktioner lagras helst i en separat js-fil.

```
function namn(Parametrar) {  
    // 0 eller flera satser  
    return (Returvärde);  
}
```



# FUNKTIONER — ETT FÖRSTA EXEMPEL

```
<head>
<script>
function hej() {
    alert("Welcome to JavaScript");
}
</script>
</head>

<body>
<script>
    hej();
</script>
</body>
```

En funktion i  
<head>

Funktionsanrop i  
body

# PARAMETRAR OCH ARGUMENT

```
<head>
```

```
<script>
```

```
function hej(msg) {
```

```
    alert(msg);
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<script>
```

```
    hej('Detta är ett argument som skickas till funktionen hej');
```

```
</script>
```

```
</body>
```



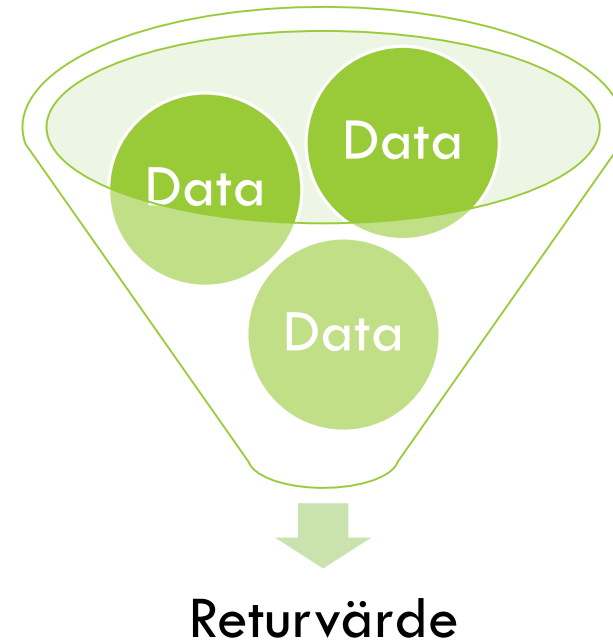
En parameter



Ett argument

# RETURVÄRDE

Att returnera något i en funktion betyder att man ber funktionen bearbeta data och skicka tillbaka ett resultat.



# ARBETA MED FUNKTIONER

Skapa en ny fil med namnet funktioner.js

Skriv följande funktion i filen

```
function moms25(pris) {  
    return pris * 1.25;  
}
```

# ANROPA FUNKTIONER

```
<script src="funktioner.js"></script>
<p id="demo"></p>
<script>
var pris = 100;
var el = document.getElementById('demo');
el.innerHTML = moms25(pris);
</script>
```

## Övningar

1. Lägg till två funktioner som beräknar momssatserna 12% och 6%
2. Kan du lösa detta med en funktion istället för tre?

# FUNKTIONEN DATE()

```
<p id="demo"></p>
```

```
<script>
```

```
var d = new Date();
```

```
document.getElementById("demo").innerHTML = d.getFullYear();
```

```
</script>
```

# IMMEDIATELY-INVOKED FUNCTION EXPRESSION IIFE (SJÄLVANROPANDE FUNKTIONER)

Kallas även "self-invoked anonymous function"

```
(function () {  
    var d = new Date();  
    document.getElementById("datum").innerHTML = d.getFullYear();  
})();
```

# ÖVNINGAR

Testa flera metoder som finns i objektet Date

[https://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](https://www.w3schools.com/jsref/jsref_obj_date.asp)

Undersök metoden setTimeout

[https://www.w3schools.com/js/js\\_timing.asp](https://www.w3schools.com/js/js_timing.asp)

**Skapa en enkel live klocka i JavaScript**



# ÖVNINGAR

## Skapa följande funktioner:

1. En funktion som beräknar en rektangels omkrets
2. En funktion som beräknar en rektangels area
3. En funktion som beräknar en kvadrats omkrets
4. En funktion som beräknar en kvadrats area

## Testa alla funktioner.

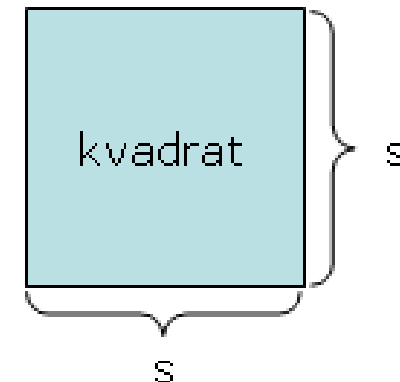
$$\text{Arean} = b * h$$

$$\text{Omkretsen} = 2b + 2h$$



$$\text{Arean} = s^2$$

$$\text{Omkretsen} = 4s$$



# ARBETA MED OBJEKT

```
var s1 = new String("Hello");  
console.log(typeof s1);
```

```
// Skapa ett String-objekt  
// object
```

```
var n1 = new Number(10);  
console.log(typeof n1);
```

```
// Skapa ett Number-objekt  
// object
```

# SKAPA OBJEKT MED "CONSTRUCTOR NOTATION"

```
var obj    = new Object();    // Skapa ett nytt objekt
obj.name   = "Mahmud";        // En egenskap
obj.hello  = function () {    // En metod
    console.log('Hej ' + obj.name);
}
obj.hello();                    // Metodanrop
```

# SKAPA OBJEKT MED "LITERAL NOTATION"

```
var person = {                                // Skapa ett objekt med { }
  name: "Mahmud",                             // En egenskap
  hello: function () {                       // En metod
    console.log('Hej ' + this.name);
  }
};

person.hello();                               // Metodanrop
```

# THE GLOBAL OBJECT



# THIS I FUNKTIONER

```
function plus(a, b) {  
  return (  
    console.log(a + b),  
    console.log(this),  
    console.log(arguments)  
  )  
}  
plus(1, 2);
```



top

3

► Window {fr

► (2) [1, 2,

# THIS I METODER (INUTI OBJEKT)

```
var calc = {  
  plus: function (a, b) {  
    return (  
      console.log(a + b),  
      console.log(this),  
      console.log(arguments)  
    )  
  }  
}  
calc.plus(1, 2);
```



top

3

► {plus: f}

► (2) [1, 2,

# ARBETA MED KLASSER

```
class Person {  
    constructor(firstName, lastName) {  
        this.firstName = firstName;  
        this.lastName = lastName;  
    }  
    fullname() {  
        return this.firstName + " " + this.lastName;  
    }  
}
```



# SKAPA EN INSTANS AV KLASSEN PERSON (SKAPA ETT OBJEKT)

```
var p1 = new Person("Mahmud", "Al Hakim");  
console.log(p1);  
console.log(p1.firstName);  
console.log(p1.lastName);  
console.log(p1.fullname());
```

# DOM

DOM står för Document Object Model

DOM gör så att alla element i ett HTML-dokument kan behandlas som fristående objekt.

DOM lägger in dokument i en väldefinierad trädstruktur.

Trädet har ett rotelement och alla objekt i modellen kallas för noder.

De vanligaste nodtyperna är: Document, Element, Attribut och Text.

Noderna är relaterade till varandra som en familj.

En nod kan ha en förälder, syskon och barn!

# DOM träd av objekt

<html>

<head>

<title>Dokument</title>

</head>

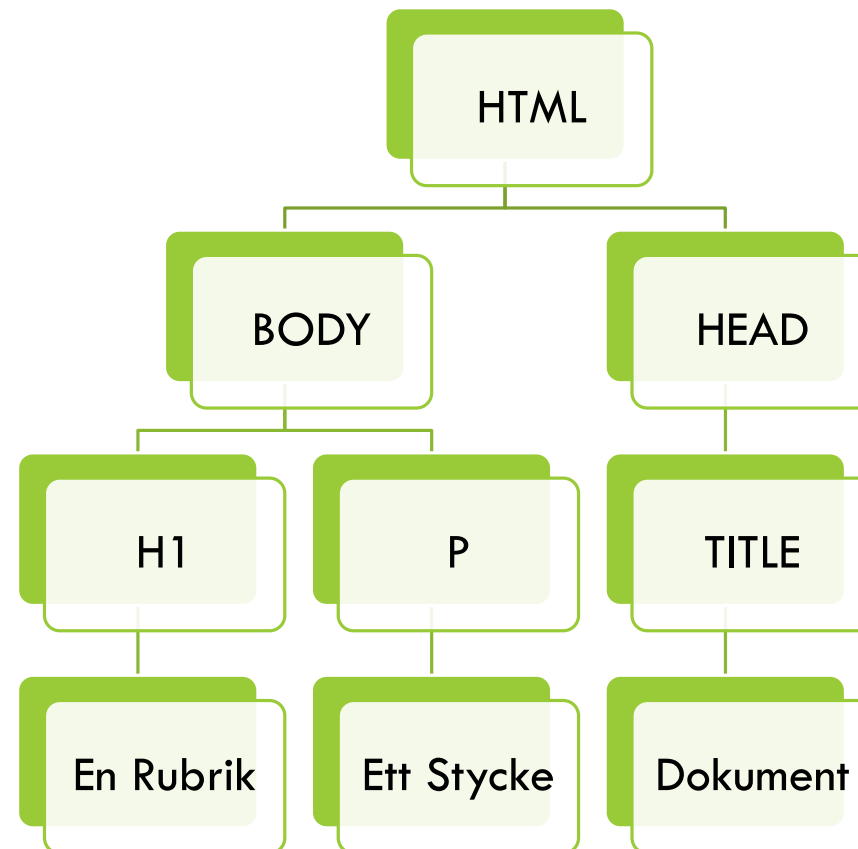
<body>

<h1>En rubrik</h1>

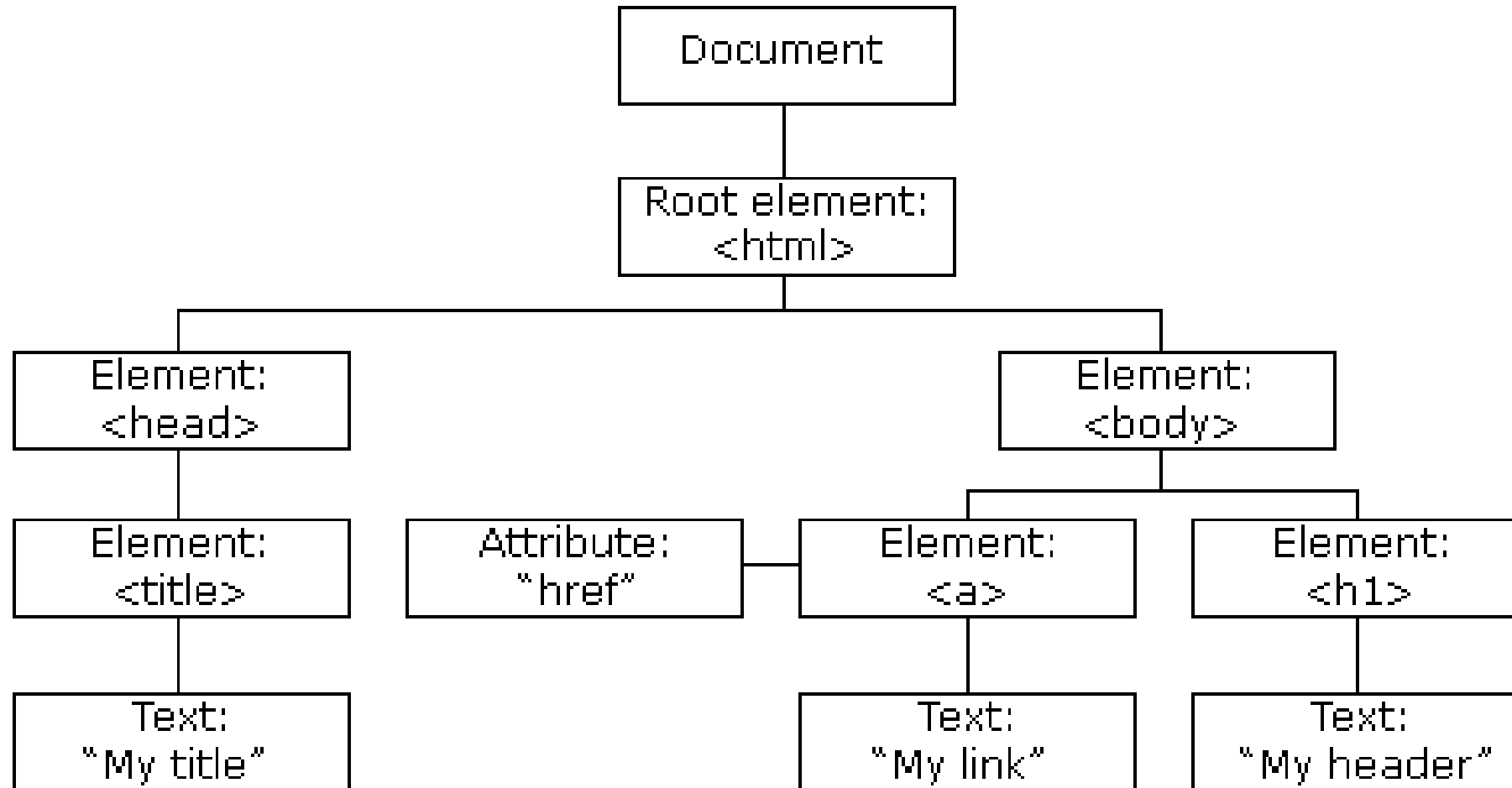
<p>Ett stycke</p>

</body>

</html>



# THE HTML DOM TREE OF OBJECTS



Bildkälla: [http://www.w3schools.com/js/js\\_htmlDOM.asp](http://www.w3schools.com/js/js_htmlDOM.asp)

# HÄNDELSE (EVENTS)

Med JavaScript kan du få webbläsare att reagera på vissa händelser.

Du kan göra dina webbsidor mer levande och interaktiva genom att de förändras på något sätt beroende på vad användaren gör.

Exempel. Användaren fyller i ett textfält, trycker på en knapp eller helt enkelt när hemsidan laddas och visas i en webbläsaren.

# NÅGRA VANLIGA HÄNDELSER

**onLoad:** När webbsidan är färdigladdad.

**onClick:** När man klickar på ett objekt.

**onMouseOver:** När muspekaren rör sig över ett objekt (t.ex. över bild eller en länk).

**onMouseOut:** När muspekaren lämnar ett objekt

Tips

[http://www.w3schools.com/jsref/dom\\_obj\\_event.asp](http://www.w3schools.com/jsref/dom_obj_event.asp)

# EXEMPEL PÅ ONLOAD

```
<html>

<head>
  <title>OnLoad</title>
</head>

<body onload="alert('Hej');">
</body>

</html>
```

# EXEMPEL PÅ ONCLICK

```
<html>
```

```
<head>
```

```
<title>OnClick</title>
```

```
</head>
```

```
<body>
```

```
<button onClick="alert('Hej');">Klicka här!</button>
```

```
</body>
```

```
</html>
```



# ONCLICK — EXEMPEL 2

<body>



```

```

```

```

```

```

</body>

# ONMOUSEOVER OCH ONMOUSEOUT

```

```



Bildkälla: <http://www.iconarchive.com/show/shop-cart-icons-by-fasticon.html>

# THIS I EVENTS

```
<body>
```

```

```

```
</body>
```

# EVENTS OCH CSS

```
<p id="demo">Lorem ipsum dolor sit amet</p>
```

```
<button
```

```
onclick="document.getElementById('demo').style.display='none' ">
```

Dölj element med JavaScript Events och CSS

```
</button>
```

## Övning

Skapa en annan knapp för att  
visa elementet!

Kan du lösa detta med en enda  
knapp (samma knapp visar och  
döljer ett element!)

# ÖVNING

## Skapa en enkel bildgalleri enligt följande

<http://webacademy.se/webmaster/ovningar/javascript/tavla/tavla.html>



Peka här nere för  
att ändra den  
stora bilden

# INMATNINGSFÄLT (PROMPT)

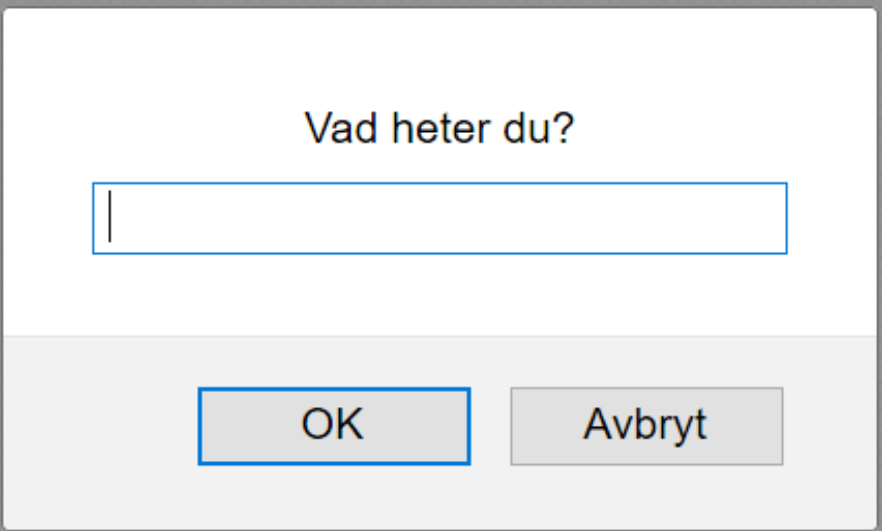
```
<p id="demo">Hej </p>
```

```
<script>
```

```
var namn=prompt("Vad heter du?");
```

```
document.getElementById('demo').innerHTML += namn;
```

```
</script>
```



Vad heter du?

OK Avbryt

# ÖVNING

Skapa ett skript som läser in två godtyckliga tal via inmatningsfält (prompt).  
Multiplicera dessa tal och visa resultatet.

# VILLKORSATSER (SELEKTIONER)

```
if(villkor){  
    // En eller flera satser  
}
```

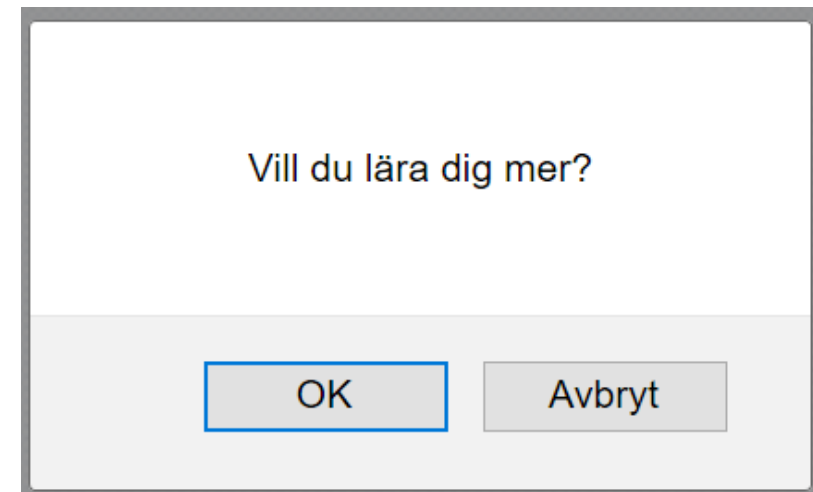


# IF-SATSEN — EXEMPEL

```
<script>
var age;
age = prompt("Ange din ålder");
if(age < 18) {
    alert("Du får inte köpa cigaretter");
}
</script>
```

# CONFIRM-RUTAN

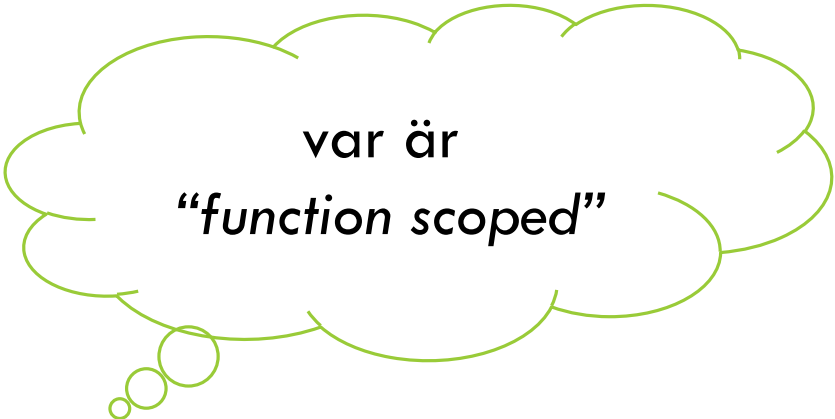
```
<script>  
var ok = confirm("Vill du lära dig mer?");  
if (ok == true) {  
    alert("Kunskap ger färdighet!");  
}  
</script>
```



# IF-ELSE-SATSEN

```
<script>
var age;
age = prompt("Ange din ålder");
if (age < 18) {
    alert("Du får inte handla här");
}
else {
    alert("Välkommen in");
}
</script>
```

# MER OM VAR



var är  
*"function scoped"*

Variabler som deklarerats med var lever i funktionens räckvidd (scope)

Exempel

```
var test = 123;  
if (true) {  
    var test = 456; // Borde vi inte få en ny lokal variabel?  
}  
console.log(test); // 456
```

# VAR OCH LET



let är  
*"block scoped"*

Variabler som deklareraras med let lever i blockets räckvidd (scope)

Exempel

```
let test = 123;  
if (true) {  
    let test = 456; // en ny lokal variabel skapas här!  
}  
console.log(test); // 123
```

# ITERATIONER

Iterationer kallas ibland för loopar eller upprepningar.

Det finns flera olika sätt att bilda iterationer i JavaScript.

1. While
2. Do-while
3. For

# WHILE-SATSEN

```
var a = 1;
while (a < 10) {
    document.write(a);
    a++;
}
```

Tips! Läs mer om metoden write

[https://www.w3schools.com/jsref/met\\_doc\\_write.asp](https://www.w3schools.com/jsref/met_doc_write.asp)

Nackdelen med write. Testa detta:

[https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_doc\\_write4](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_doc_write4)

# DO-WHILE-SATSEN

```
var a = 1;
do {
    document.write(a);
    a++;
}
while (a < 10)
```



# FOR-SATSEN

```
for (startvärde ; villkor ; uppdateringsvärde) {  
    // en eller flera satser  
}
```

**Startvärdet** är det värde variabeln har från början.

**Villkoret** kan vara sant eller falskt och upprepas tills dess att det inte längre är sant.

**Uppdateringsvärdet** anger hur variabeln skall förändras medans upprepningen sker.

# FOR-SATSEN

```
for (a = 1; a <= 10; a++) {  
    document.write(a);  
}
```

## Övningar

1. Skapa en for-loop som visar alla udda tal mellan 1 och 10
2. Skapa en for-loop som visar alla jämna tal mellan 1 och 10

# NÄSTLADE LOOPAR

```
var x = 1;
while (x <= 10) {
    for (y = 1; y <= 10; y++) {
        document.write(y, " &times; ", x, "=", y * x, "<br>");
    }
    document.write("<br>");
    x++;
}
```

# ÖVNING 1

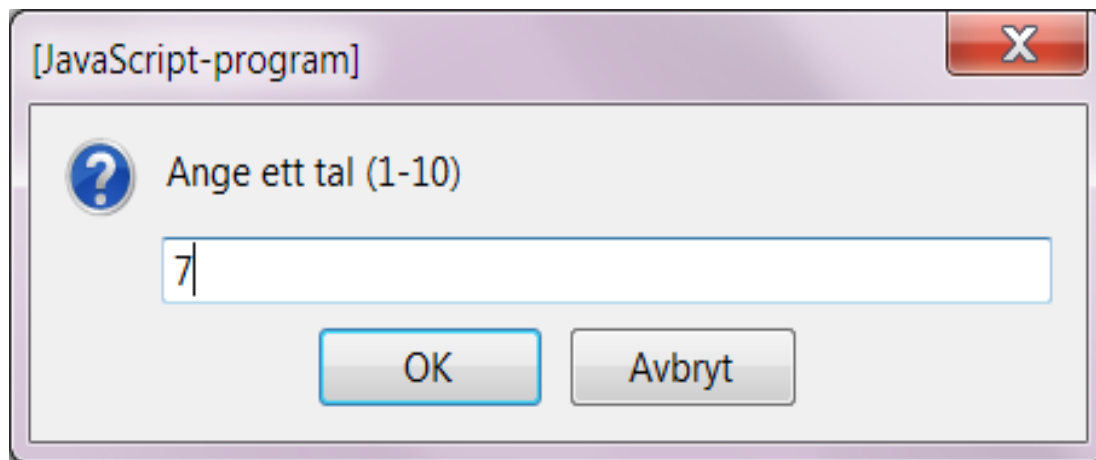
Skapa ett skript som visar femmans multiplikationstabell i en HTML-tabell

$1 \times 5 = 5$
$2 \times 5 = 10$
$3 \times 5 = 15$
$4 \times 5 = 20$
$5 \times 5 = 25$
$6 \times 5 = 30$
$7 \times 5 = 35$
$8 \times 5 = 40$
$9 \times 5 = 45$
$10 \times 5 = 50$

# ÖVNING 2

Skapa ett skript som ber användaren mata in ett tal mellan 1 och 10.

Visa därefter talets multiplikationstabell enligt bilden nedan



A JavaScript dialog box titled "[JavaScript-program]" with a close button (X) in the top right corner. The main area contains a question mark icon and the text "Ange ett tal (1-10)". Below this is a text input field containing the number "7". At the bottom are two buttons: "OK" and "Avbryt".



1	x	7	=	7
2	x	7	=	14
3	x	7	=	21
4	x	7	=	28
5	x	7	=	35
6	x	7	=	42
7	x	7	=	49
8	x	7	=	56
9	x	7	=	63
10	x	7	=	70

# ÖVNING 3

Skapa en funktion som visar multiplikationstabellen (1-10) enligt bilden.

Placera funktionen i en separat JS-fil.  
Testa funktionen i ett HTML-dokument.

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100