

Frontend <Lektion class="åtta">

Utbildare: Mahmud Al Hakim

NACKADEMIN

Lektionstillfällets mål

Mål med lektionen

- Funktioner
- Parametrar och Argument
- Selektioner
- Iterationer
- Arbeta med objekt
- Egenskaper och Metoder
- **Arbetsmetod**
- Teori och praktik varvas under lektionen

NACKADEMIN

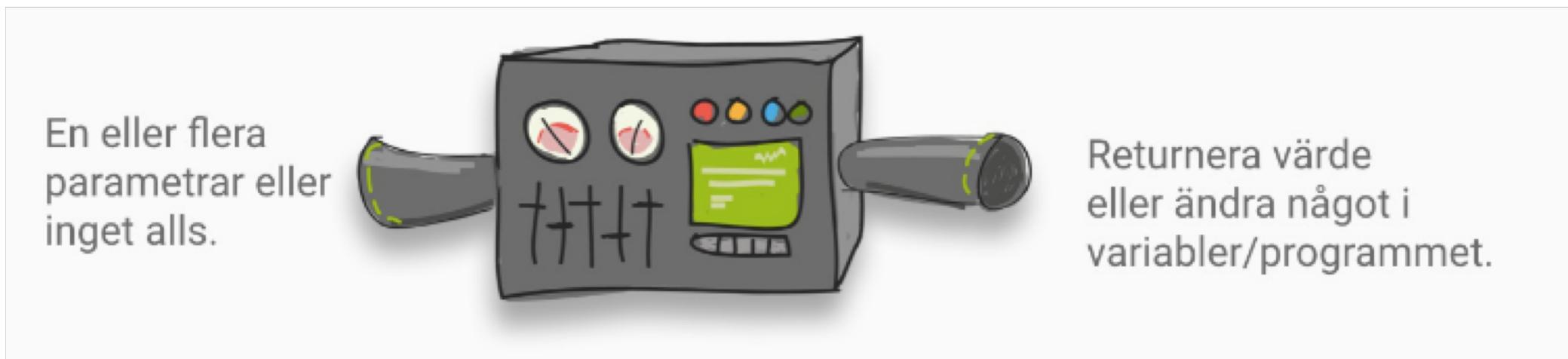
Kort summering av föregående lektion

Föregående lektion:

- Att förstå JavaScript
- Att länka JavaScript till HTML
- Introduktion till objekt och metoder
- Syntax
- Satser (statements)
- Kommentarer
- Variabler och värde
- Datatyper (typ)
- Arrayer

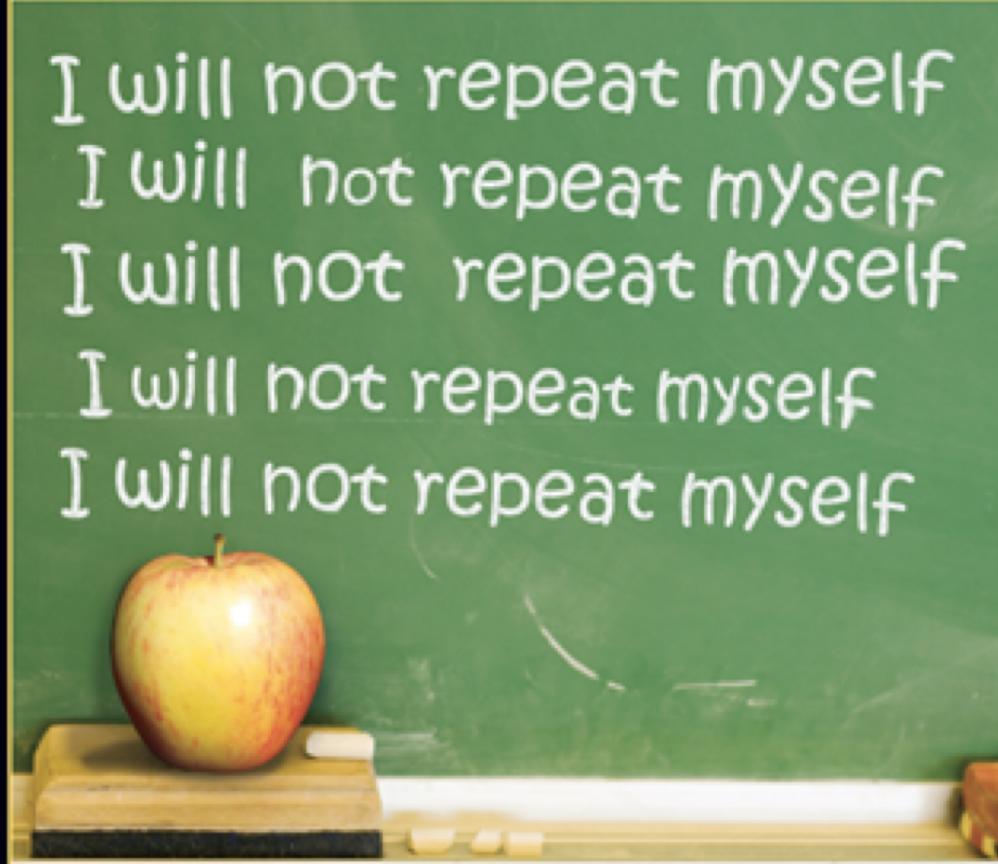
Vad är en funktion?

- En funktion är en del av ett program som kan anropas för att utföra en viss uppgift oberoende av resten av koden.
- Funktioner hjälper oss att återanvända kod.



Bildkälla: <https://matematikvideo.se/lektioner/funktioner-i-programmering/>

DRY



I will not repeat myself
I will not repeat myself

DON'T REPEAT YOURSELF

Repetition is the root of all software evil.

Deklarera en funktion – Grundläggande syntax

```
function name() {  
    // En eller flera satser...  
}
```

En funktion deklareras med nyckelordet **function**. Efter namnet kommer alltid parenteser () och direkt efter ett **kodblock** som skrivs inom klammerparenteser { } (måsvingar).

En parameterlös funktion

```
function hej(){  
    alert("Hej från en funktion");  
}
```

Anropa en funktion (call/invoke)

- En funktion kan anropas med ()-operator (parenteser)

```
hej();
```

Övningar

- Skapa en funktion som heter **meddelande**
- Funktionen visar meddelandet "Jag har anropats!" i konsolfönstret.
- Anropa funktionen 3 gånger.
- Flera övningar

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_functions1

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_functions2

Parametrar

- Funktioner behöver ibland data för att utföra ett visst uppdrag.
- Data skickas till en funktion via ett godtyckligt antal parametrar.

```
function name(parameter1, parameter2, ...) {  
    // En eller flera satser...  
}
```

En funktion med en parameter

```
function hej(msg){  
    console.log(msg);  
}
```

Argument

- Data skickas till en funktion vid anropet som argument (inom parenteser)

```
hej("Argument – första anropet");
```

```
hej("Argument – andra anropet");
```

Förvald (default) parameter

- Vad får du om du anropar funktionen utan några argument?

```
hej(); // undefined
```

- Lösningen till detta problem är att ange en förvald parameter

```
function hej(msg = "Welcome") {  
    console.log(msg);  
}  
hej(); // Welcome
```

Funktioner med två parametrar

```
function hej(firstName, lastName){  
    console.log("Hej " + firstName + " " + lastName );  
}
```

```
hej("Mahmud", "Al Hakim");  
hej("Kalle", "Anka");
```

Funktioner med flera parametrar

```
function manyThings(...things) {  
    console.log(things);  
}  
manyThings(1);  
manyThings(1,2);  
manyThings(1,2,3);
```

... är en operator
(tre punkter)
Här skapas en array av alla
argument som skickas till
funktionen

Return

- Funktioner kan returnera ett värde.
- Detta kan vi göra med hjälp av nyckelordet **return**

```
function fullName(firstName, lastName){  
    return firstName + " " + lastName;  
}
```

```
let name = fullName("Mahmud", "Al Hakim");  
console.log(name);
```

Övningar

- a) Skapa en funktion som beräknar medelvärdet av två tal.
- b) Skapa en funktion som beräknar medelvärdet av tre tal.

Kontrollstrukturer

- En algoritm består av ett antal operationer och anvisningar om i vilken ordning operationerna skall utföras.
- En algoritms komponenter kallas för kontrollstrukturer.

1. Sekvens

En följd av satser som utförs en i taget i den ordning de skrivits.

2. Selektion (val)

Utför olika satser i olika situationer.

Innebär att ett alternativ bland två eller flera skall väljas.

3. Iteration (repetition, upprepning)

Satser som upprepas ett visst antal .

if-satsen

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

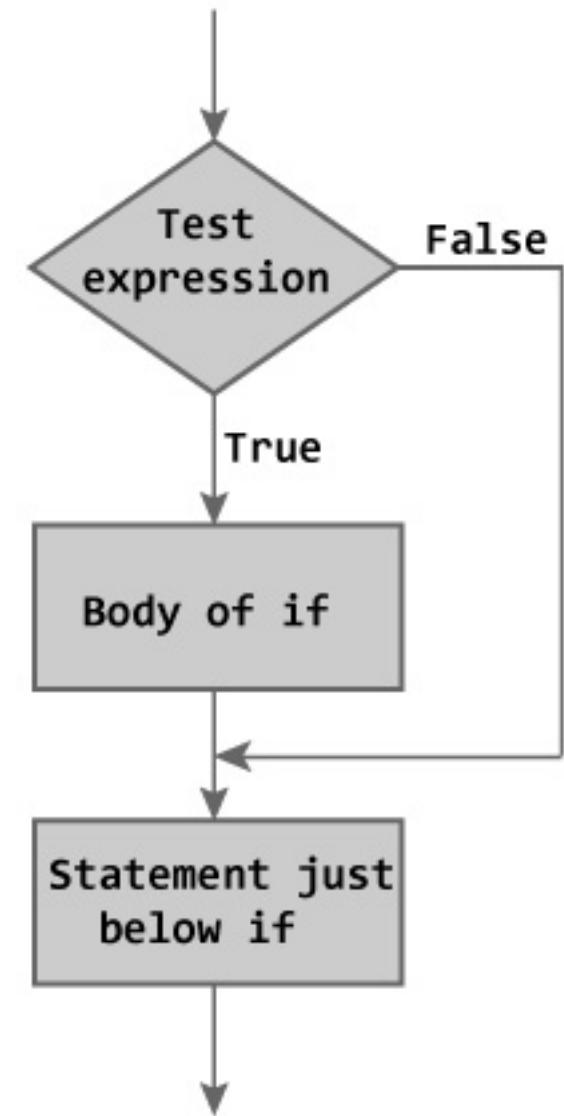


Figure: Flowchart of if Statement

if-satsen – Exempel

```
let score = 75;  
  
if (score >= 50) {  
    congratulate();  
}  
  
function congratulate(){  
    let msg ='Congratulations! '  
    msg += 'Proceed to the next round.';  
    console.log(msg);  
}
```

Jämförelseoperatorer (Comparison Operators)

Operator	Description
<code>==</code>	equal to
<code>====</code>	equal value and equal type
<code>!=</code>	not equal
<code>!==</code>	not equal value or not equal type
<code>></code>	greater than
<code><</code>	less than
<code>>=</code>	greater than or equal to
<code><=</code>	less than or equal to
<code>?</code>	ternary operator

```
if...else
```

```
let score = 40;
```

```
if (score >= 50) {  
    congratulate();  
}
```

```
else{  
    encourage();  
}
```

```
function encourage(){  
    let msg ='Have another go!';  
    console.log(msg);  
}
```

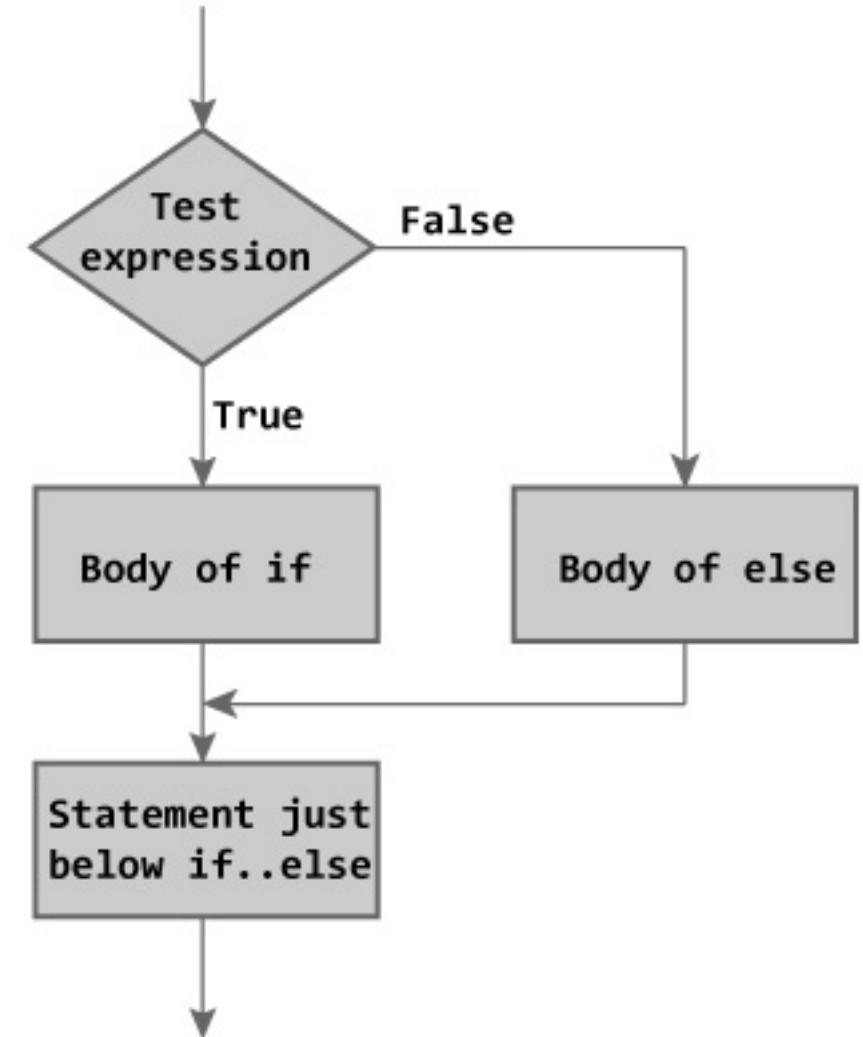


Figure: Flowchart of if...else Statement

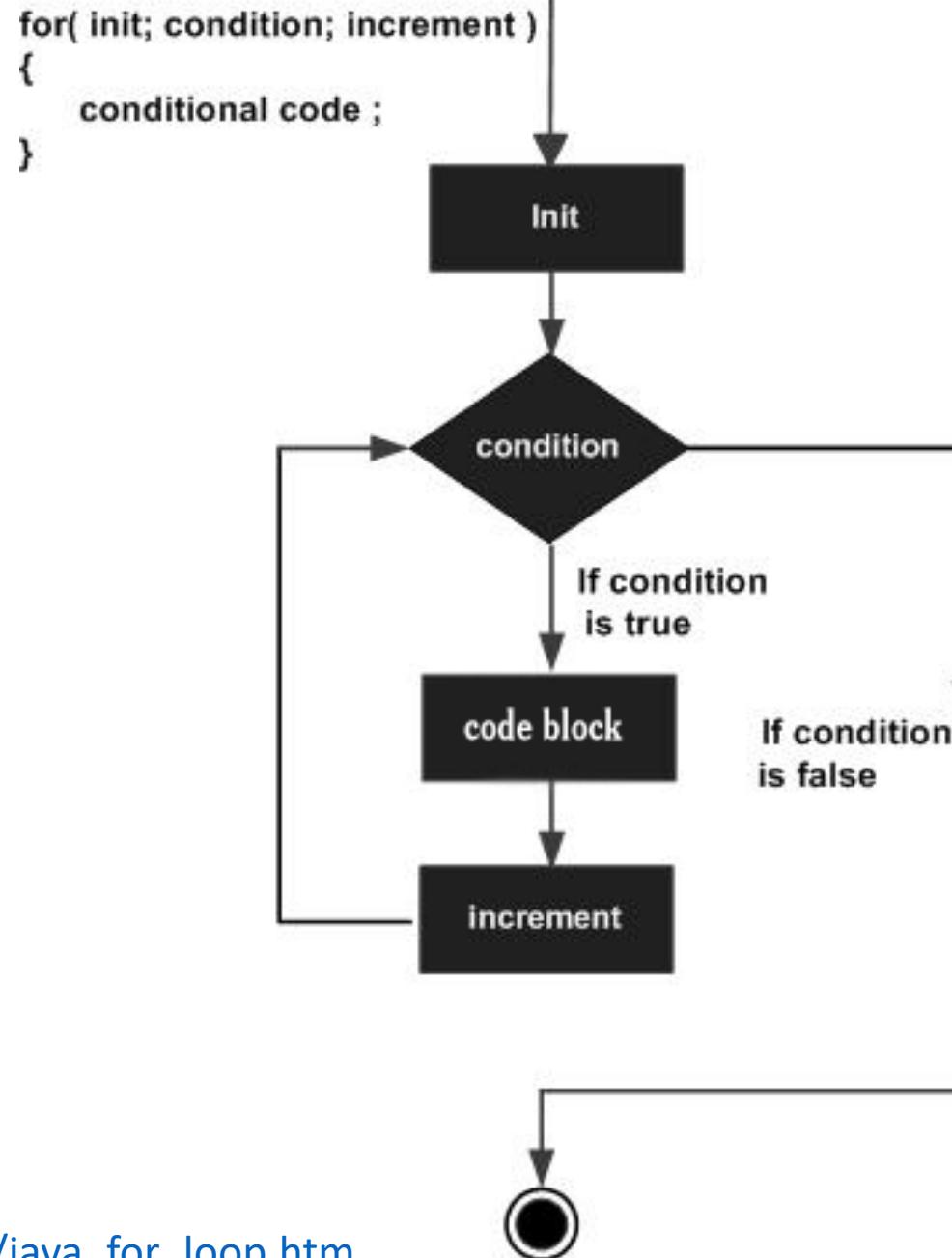
Logiska operatorer (Logical Operators)

Operator	Description
&&	and
	or
!	not

Logiska operatorer – Exempel

```
function promptAge() {  
  
    let age = prompt("Hur gammal är du?");  
    if (age == null || age == "")  
        return;  
  
    let voteable = (age < 18) ? "Du får inte rösta!"  
                            : "Ok. Du får rösta";  
    alert(voteable);  
}  
  
promptAge();
```

For-satsen



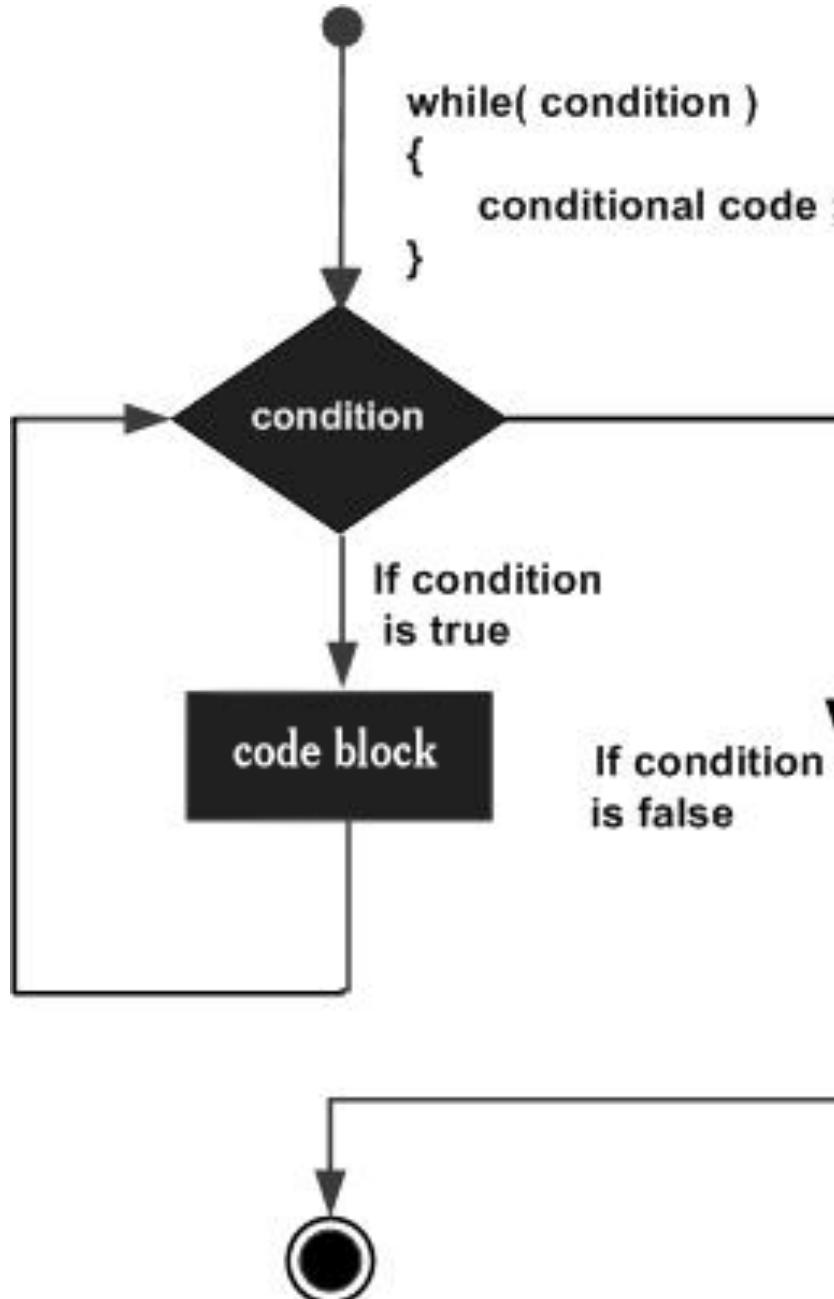
For-satsen – Exempel

```
for (let i = 0 ; i < 10 ; i++) {  
    console.log(i);  
}
```

Övning

Skapa en for-loop
som skriver ut alla jämna tal
mellan 1 och 20

While-satsen



While-satsen – Exempel

```
let i = 1;  
  
while (i<=10) {  
    console.log(i + ' x 5 = ' + i*5) ;  
    i++;  
}
```

Break-satsen

```
while(true){  
    let name = prompt("Vad heter du?");  
    if(name === null){  
        alert('Hejdå');  
        break;  
    }  
    alert('Välkommen ' + name);  
} // Slut på while-satsen  
// Hit kommer vi när break exekveras
```

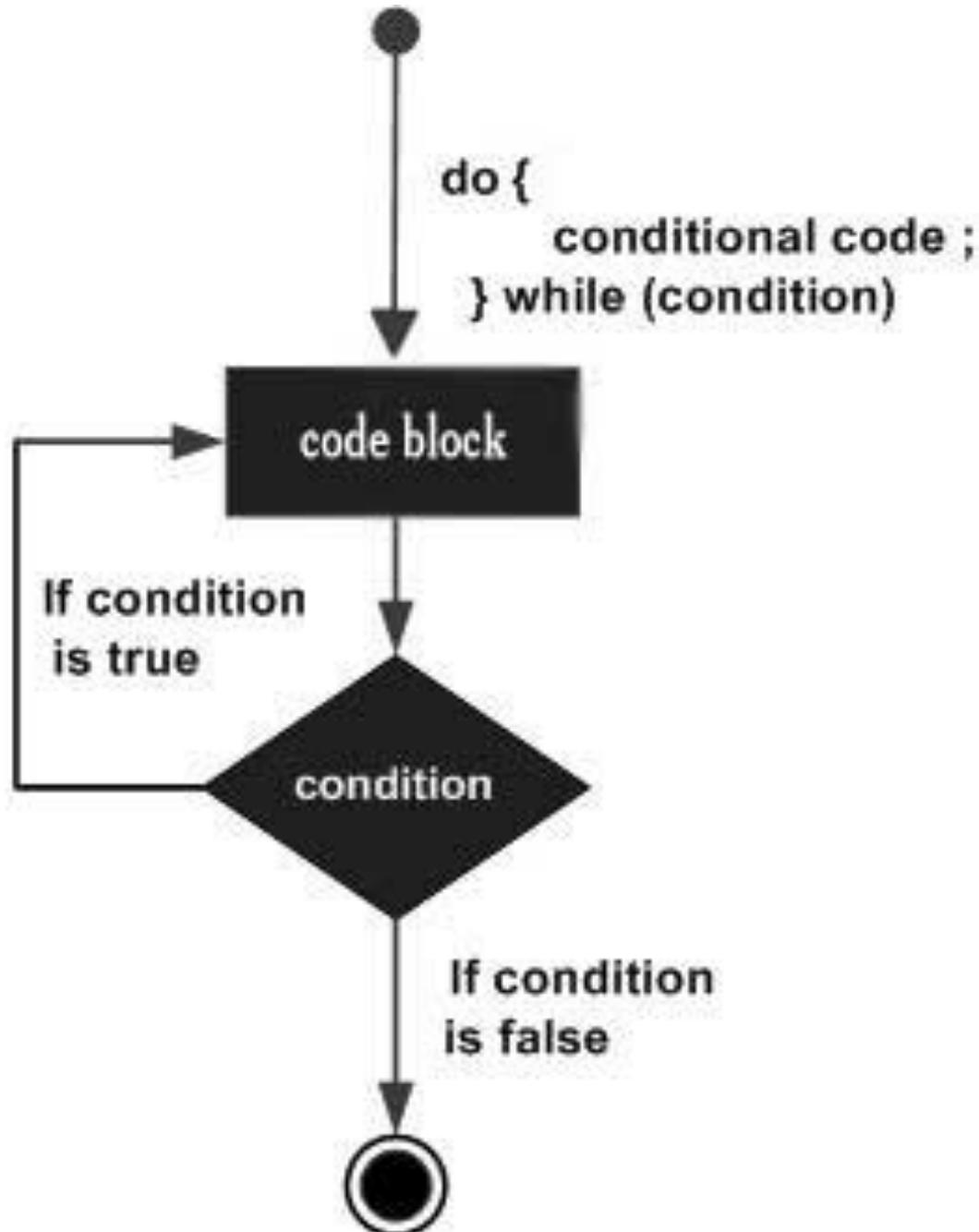
Continue-satsen

```
while (true) { // Hit kommer vi när continue exekveras
    let name = prompt("Vad heter du?");
    if (name === '')
        continue;
    if (name === null)
        break;

    alert('Välkommen ' + name);
}

// Hit kommer vi när break exekveras
```

Do-While-satsen



Do...While – Exempel

```
let again = true;  
do{  
    alert('Välkommen till spelet!');  
  
    again = confirm("Vill du spela igen?");  
  
}while(again);  
  
alert('Hejdå!');
```



Övning

På en arbetsplats erbjuds
man följande lön.

Första dagen tjänar man en
krona och sedan fördubblas
lönen varje dag.

Skriv ett program som räknar
ut hur länge man behöver
arbeta för att tjäna ihop
minst 10 miljoner kronor.

Arbeta med objekt i JavaScript

- Objektorienterad programmering är ett speciellt sätt att tänka och att bygga upp program.
- Ett program byggs upp som ett antal objekt som samverkar med varandra.
- Det innebär att man får en bättre och mer lätt hanterad struktur i ett program.
- **Nästan allt i JavaScript är objekt.**
- *Alla objekt i JavaScript har en speciell prototyp som heter Object (överkurs)*

Olika sätt att skapa objekt i JavaScript

1. Object Literal

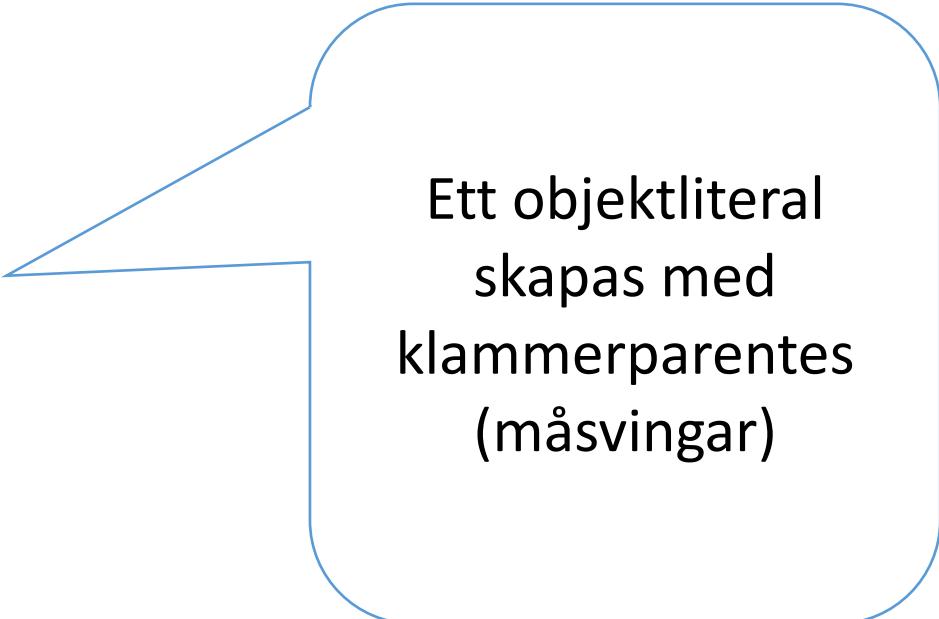
```
let person = {};
```

2. Object constructor

```
let today = new Date();
```

Att skapa ett objektliteral (Literal notation)

```
let person = {};
```



Ett objektliteral
skapas med
klammerparentes
(måsvingar)

Nyckel-värde-par (name:value pairs)

```
let person = {  
    name: "Mahmud Al Hakim",  
    age : 45  
}
```

Ett objekt kan innehålla flera nyckel-värde-par vilket är en kombination av en nyckel och ett värde t.ex.
`age` är en nyckel
`45` är ett värde

Egenskaper och metoder

- **Egenskaper**
 - Ett objekt har egenskaper (properties på engelska).
 - Egenskaper är information/data som skall hanteras.
- **Metoder**
 - Objekt har också funktionaliteter som skapas med hjälp av olika funktioner.
 - Funktioner i ett objekt kallas metoder.

Egenskaper och metoder – Exempel

```
let person = {  
    name: "Mahmud Al Hakim",  
    age : 45,  
    info: function(){  
        console.log(this.name + " är " +  
                    this.age + " år gammal!");  
    }  
}
```

Nyckelordet **this**
refererar till det
aktuella objektet

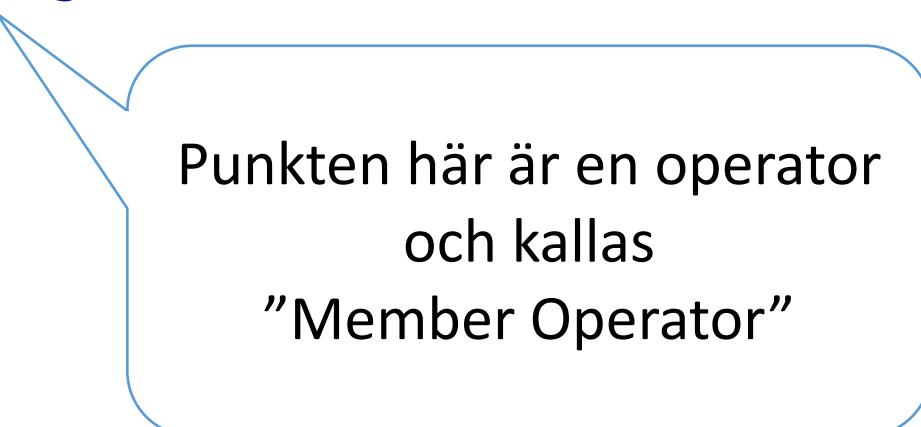
Punktnotation (dot notation)

- För att komma åt objektets egenskaper och metoder använder vi punktnotationen (dot notation).

```
console.log(person.name);
```

```
console.log(person.age);
```

```
person.info();
```



Punkten här är en operator
och kallas
"Member Operator"

Ändra objektets egenskaper

```
let person = {  
    name: "Mahmud Al Hakim",  
    age : 45  
}
```

```
person.name = "Mahmud Al-Hakim"
```

```
person.age++;
```

```
console.log(person);
```

Ta bort en egenskap

```
let person = {  
    name: "Mahmud Al Hakim",  
    age : 45  
}  
  
delete person.age;  
  
console.log(person.age); // undefined
```

Lägg till en egenskap

```
let person = {  
    name: "Mahmud Al Hakim",  
    age: 45  
}
```

```
person.weight = 90;
```

Ta bort ett objekt

```
let person = {  
    name: "Mahmud Al Hakim",  
    age : 45  
}
```

```
person = null;  
  
console.log(person); // null
```

Övning 1

- Skapa ett objekt som innehåller info om denna kurs.
Exempel på egenskaper:
 - Kursnamn
 - Lärare
 - Skola
 - Antal veckor
 - Period

Övning 2

- Skapa tre objekt som innehåller info om olika bilar.
Ange följande egenskaper:
 - Registreringsnummer
 - Fabrikat
 - Modell
 - Modellår
 - Växellåda

Globala JS-Objekt

String

Number

Math

Date

String

Objektet String

- Om du skapar en sträng i JS så kommer strängen att få tillgång till objektet String.
- Strängen kommer alltså att få ett antal egenskaper och metoder.

```
let firstName = 'Mahmud';
firstName.|
```

A screenshot of a code editor showing a tooltip for the 'length' property of a string variable. The tooltip is titled '(property) String.length: number' and includes a small info icon. The tooltip lists several other properties: link, localeCompare, match, normalize, repeat, replace, search, slice, small, split, and startsWith. The 'length' property is highlighted with a blue background.

- length (property) String.length: number
- link
- localeCompare
- match
- normalize
- repeat
- replace
- search
- slice
- small
- split
- startsWith

Ändra till versaler/gemener

```
let myName = 'Mahmud';
myName.toUpperCase();
console.log(myName); // Mahmud
myName = myName.toUpperCase();
console.log(myName); // MAHMUD
myName = myName.toLowerCase();
console.log(myName); // mahmud
```

“JavaScript
Strings are
immutable”

Substring() och indexOf()

```
let myName = 'Mahmud Al Hakim';

// Returnera en sträng från index 0 till index 6
let firstName = myName.substring(0,6);

// Hitta första mellanslaget
let space = myName.indexOf(' ');

// Returnera en sträng från första tecknet
// efter mellanslaget och ända till slutet.
let lastName = myName.substring(space+1);
```

Metoden Split()

```
// Metoden split() skapar en array  
let fullname = 'Mahmud Al Hakim';  
let nameArray = fullname.split(' ');\n  
console.log(nameArray);  
// [ 'Mahmud', 'Al', 'Hakim' ]
```

Metoden trim()

```
let fullname = '    Mahmud Al Hakim    ';
// OBS! extra blanka tecken i början och slutet av strängen

// Metoden trim() rensar blanka tecken
// OBS! trim() rensar enbart start och slut, ej mitten!

fullname = fullname.trim();
```

Objektet Number

Number

- Om du skapar en variabel som innehåller ett värde av typen Number så kommer värdet att få tillgång till objektet Number.

```
let PI = 3.14159;  
PI.  
  ↴  
  toExponential  
 toFixed  
  toLocaleString  
  toPrecision  
  toString  
  valueOf  
  ↵ PI
```

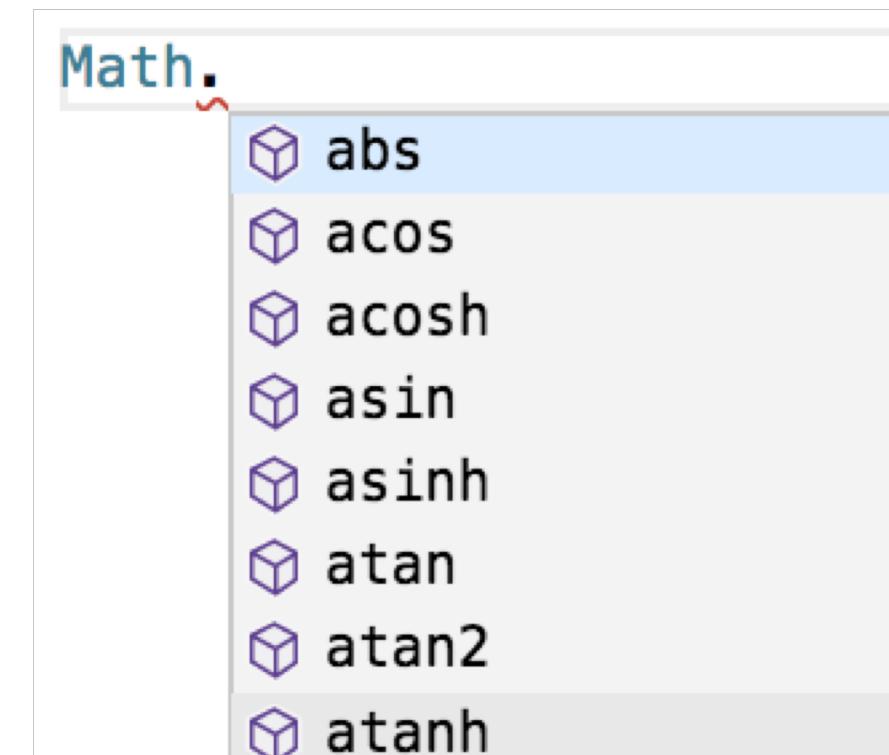
Avrundning med decimaler

```
let PI = 3.14159;  
// Avrunda till två decimaler (visa två decimaler)  
console.log(PI.toFixed(2)); // 3.14  
  
// Avrunda och visa sammanlagt 5 siffror  
console.log(PI.toPrecision(5)); // 3.1416
```

Objektet Math

- Det finns ett globalt objekt i JavaScript som heter Math.
- Math innehåller ett antal matematiska konstanter och metoder.

Math



Math – Några egenskaper

```
console.log(Math.PI);  
console.log(Math.E);
```

Avrunda till närmaste heltal

```
console.log(Math.round(2.234));  
console.log(Math.round(-2.234));  
console.log(Math.round(2.500));  
console.log(Math.round(-2.500));  
console.log(Math.round(2.999));  
console.log(Math.round(-2.999));
```

Beräkna potenser (x^2)

```
console.log(Math.pow(1, 2)); // 1
console.log(Math.pow(2, 2)); // 4
console.log(Math.pow(3, 2)); // 9
console.log(Math.pow(4, 2)); // 16
console.log(Math.pow(5, 2)); // 25
```

Beräkna kvadratrotten

```
console.log(Math.sqrt(1));  
console.log(Math.sqrt(4));  
console.log(Math.sqrt(9));  
console.log(Math.sqrt(16));  
console.log(Math.sqrt(25));
```

Max och min

```
console.log(Math.max(5, 10)); // 10  
console.log(Math.min(5, 10)); // 5  
console.log(Math.max(20, 5, 10, 15)); // 20  
console.log(Math.min(20, -5, 10)); // -5
```

Generera slumptal

- Metoden Random visar ett slumpmässigt tal mellan 0 och 1

```
console.log(Math.random());
console.log(Math.random());
console.log(Math.random());
```

```
/* Exempel
 0.6358858487626491
 0.7456010880615456
 0.8347153409686046
 */
```

Generera slumptal mellan 1 och 10

```
console.log(Math.random() * 10 + 1);  
console.log(Math.random() * 10 + 1);  
console.log(Math.random() * 10 + 1);
```

/* Exempel

7.022430794889102

6.892513947690885

10.537925677647532

*/

Övning
Ta bort alla
decimaler

Objektet Date

Date

- För att arbeta med datum och tid i JavaScript behöver du skapa en instans av objektet Date.

```
let today = new Date();
```

Arbeta med datum

```
console.log(today.getFullYear()); // YYYY t.ex. 2019  
console.log(today.getMonth());   // 0-11  
console.log(today.getDate());    // 1-31  
console.log(today.getDay());    // 1-6
```

Arbeta med tid

```
console.log(today.getHours());    // 0-23  
console.log(today.getMinutes()); // 0-59  
console.log(today.getSeconds()); // 0-59
```

toLocaleDateString och toLocaleTimeString

```
console.log(today.toLocaleDateString());  
// Visar t.ex. 2019-01-30
```

```
console.log(today.toLocaleTimeString());  
// Visar t.ex. 10:14:19
```

Övning 1

Visa datum i detta format

Dag den xx månad år

t.ex.

Onsdag den 30 januari 2019

Tips: Använd arrayer för att
lagra veckodagar och månader.

Övning 2

- Skapa ett program som läser in ett **namn** via en prompt.
- Skicka namnet till en funktion som returnerar en hälsningsfras, t.ex. "Hej **namn**. Varmt välkommen!".
- Visa meddelandet på webbsidan. (Alltså i body, ej alert eller console).



Övning 3

- Skapa ett program som läser in antalet minuter man i genomsnitt ringer per månad samt kostnaden per minut (via 2 prompt).
- Som resultat skall programmet visa en dialogruta (alert) där det anges hur stor den beräknade kostnaden per månad blir.

Summering av dagens lektion

- Vi har gått igenom
 - Funktioner
 - Parametrar och Argument
 - Selektioner
 - Iterationer
 - Arbeta med objekt
 - Egenskaper och Metoder
- Reflektioner kring dagens lektion?

Framåtblick inför nästa lektion

Under nästa lektion kommer vi att jobba med

- DOM – Document Object Model
- Händelsehantering (Events)