

**JavaScript 1**  
**alert("Lektion 11");**

Utbildare: Mahmud Al Hakim

**NACKADEMIN**

# Lektionstillfällets mål

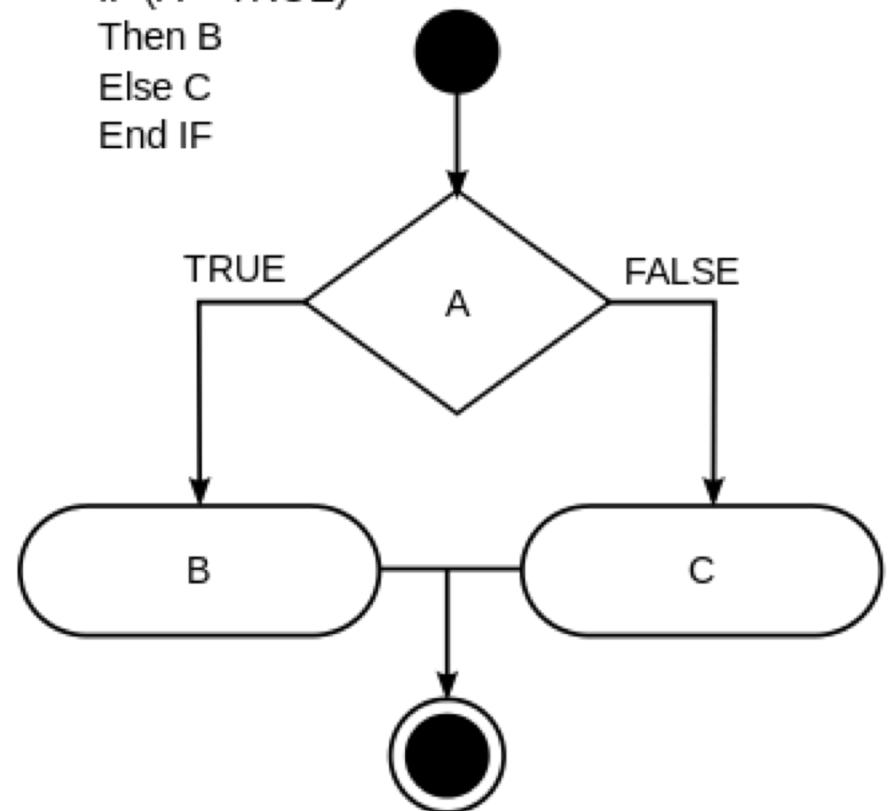
- **Mål med lektionen**
  - Arbeta med iterationer
  - For–satsen
    - For-in-satsen
    - For-of-satsen
  - While–satsen
  - Do–while–satsen
- **Arbetsmetod**
  - Teori och praktik varvas under lektionen

# Kort summering av föregående lektion

## Föregående lektion:

- Vi har jobbat med selektioner.

```
IF (A = TRUE)
Then B
Else C
End IF
```



# Repetition: if...else med Mosh

<https://youtu.be/lsg4Xd6LlsM>

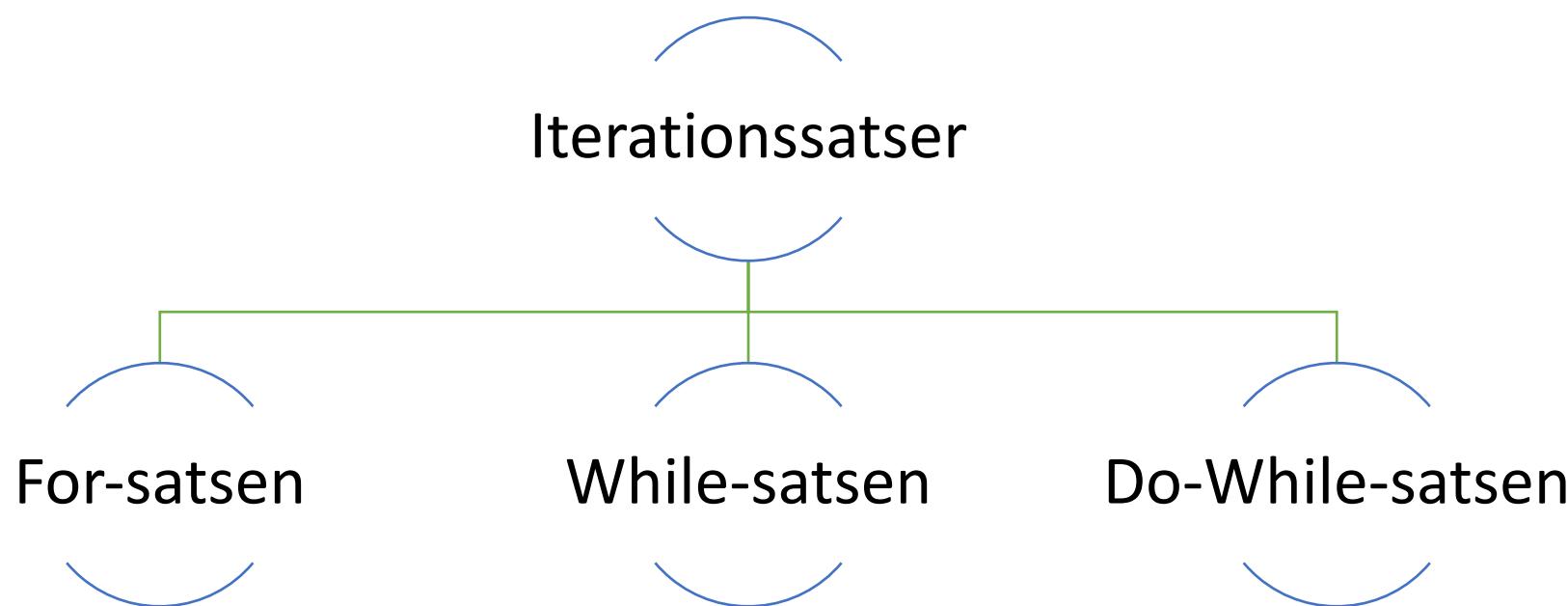
If...else



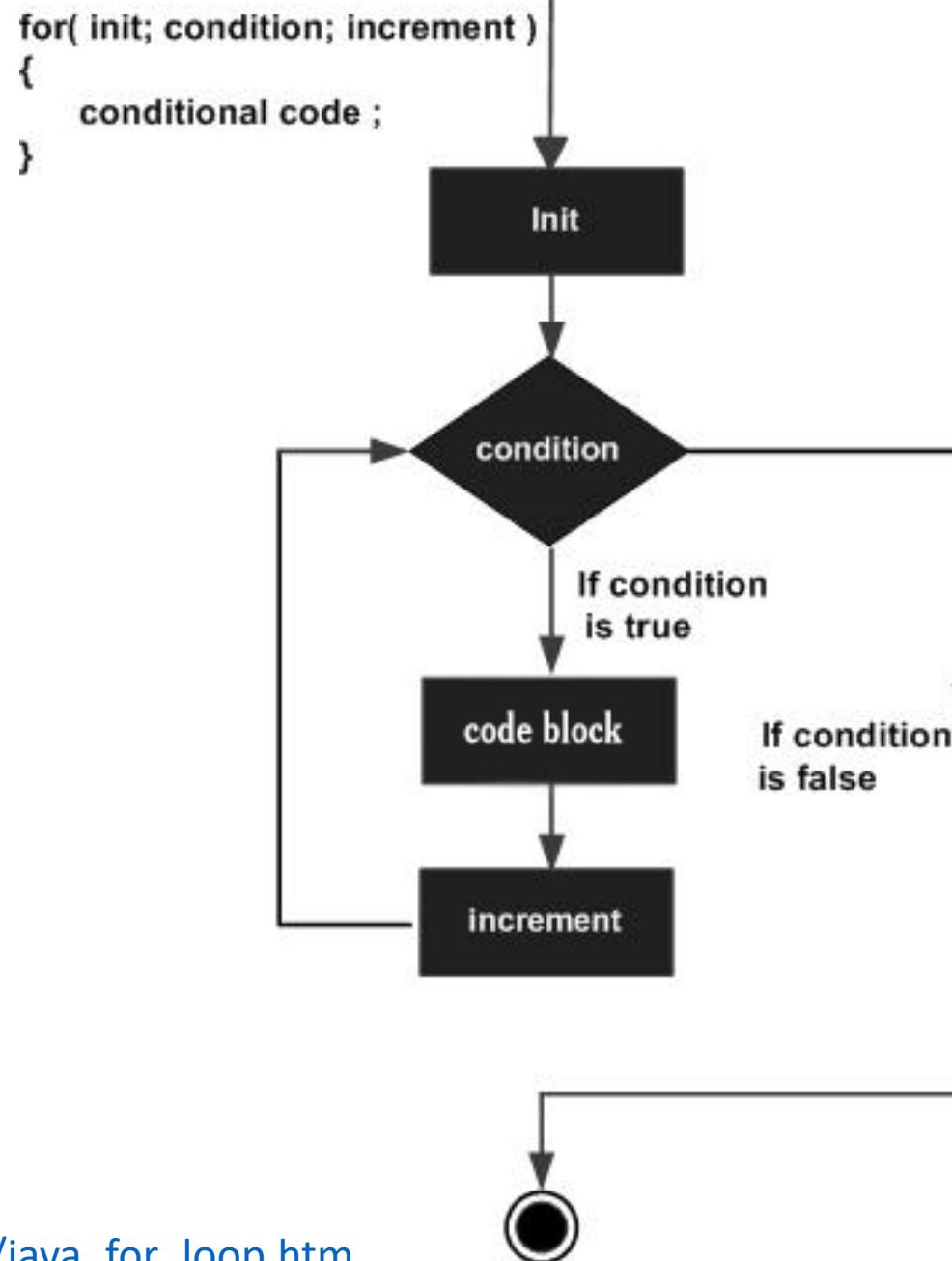
**Mosh Hamedani**  
[programmingwithmosh.com](http://programmingwithmosh.com)

# Vad menas med en iteration?

- Iteration är ett annat ord för upprepning.
- Inom JavaScript åstadkoms iteration genom så kallade slingor (loops).



# For-satsen



## For-satsen – Exempel 1

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

## For-satsen – Exempel 2

```
for (let i = 1 ; i <= 10 ; i++) {  
    console.log(i);  
}
```

## For-satsen – Exempel 3

```
for (let i = 1 ; i <= 10 ; i=i+2) {  
    console.log(i);  
}
```

## For-satsen – Exempel 4

```
for (let i = 1 ; i <= 10 ; i++) {  
  
    if(i%3 === 0)  
        console.log(i + ' är delbart med 3');  
  
}
```

# Lite om debugging i Chrome DevTools

The screenshot shows the Chrome DevTools debugger interface. On the left, there is a code editor window titled "test.js" containing the following JavaScript code:

```
1 for (let i = 1; i <= 10; i++) {  
2     if(i%3 == 0)  
3         console.log(i + ' är delbart med 3');  
4 }  
5  
6
```

The line "if(i%3 == 0)" is highlighted with a blue selection bar. The status bar at the top right indicates "Debugger paused".

On the right side, there is a sidebar with the following sections:

- i Debugger paused
- ▶ Watch
- ▶ Call Stack
- ▼ Scope
- ▼ Block
  - i: 5
- ▶ Global

# Övning 1

Skriv ut alla jämna tal  
mellan 1 och 20

# Övning 1

Skriv ut alla tal  
mellan 1 och 100,  
men enbart tal som är  
jämnt delbart med 5.

# Iteration över en array

```
let os = ['Windows', 'Macintosh', 'Linux'];

for (let i = 0 ; i < os.length ; i++) {

    console.log(os[i]);

}
```

# Iterera över en array av objekt

```
let data = [
    {name: 'Mahmud', email: 'mahmud@email.com'},
    {name: 'Kalle' , email: 'kalle@email.com'},
    {name: 'Erik' , email: 'erik@email.com'}
];
for (let i = 0; i < data.length; i++) {
    console.log(data[i].name);
}
```

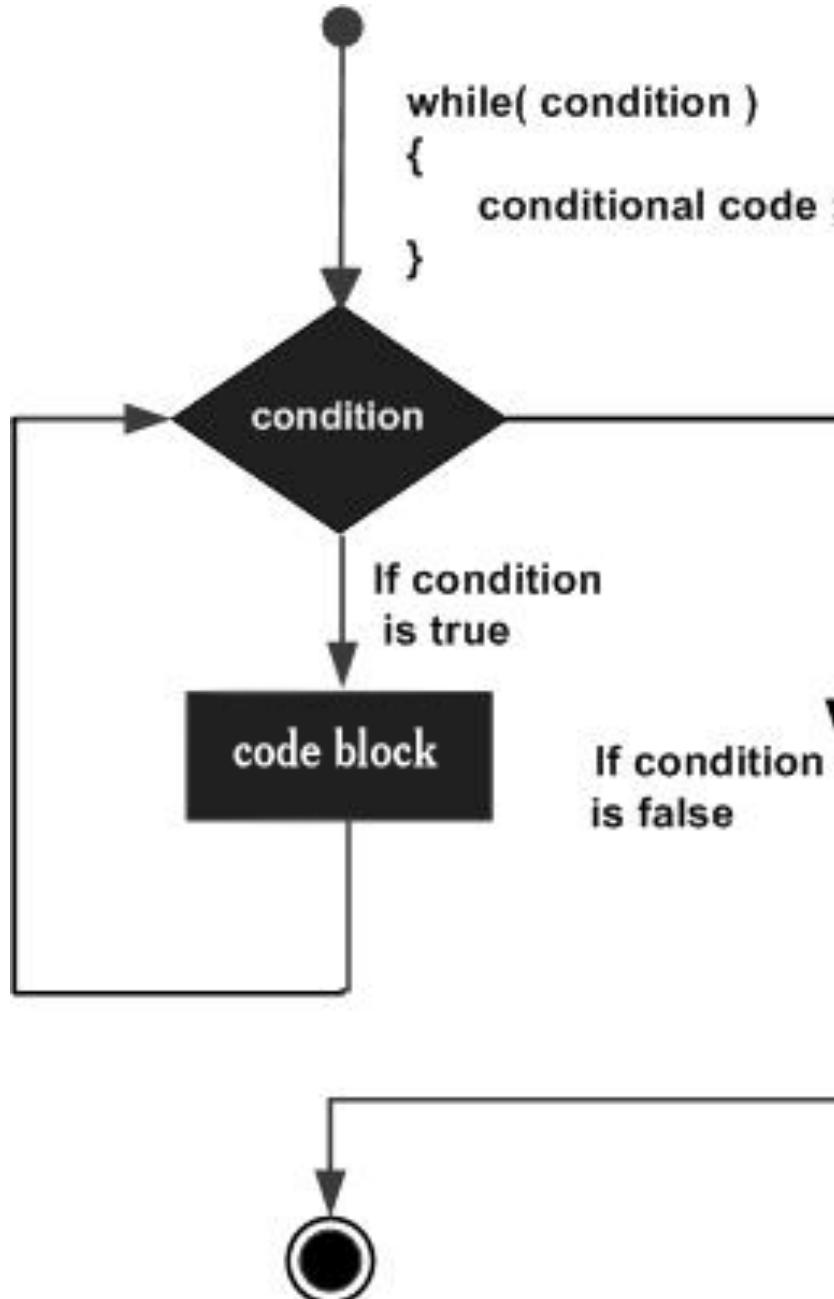
# For-in-satsen

```
let data = [
    {name:'Mahmud', email:'mahmud@email.com'},
    {name:'Kalle' , email:'kalle@email.com'},
    {name:'Erik'   , email:'erik@email.com'}
];
for (let i in data) {
    console.log(data[i].name);
}
```

# For-of-satsen

```
let data = [
    {name:'Mahmud', email:'mahmud@email.com'},
    {name:'Kalle' , email:'kalle@email.com'},
    {name:'Erik'   , email:'erik@email.com'}
];
for (let obj of data) {
    console.log(obj.name);
}
```

# 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);
}

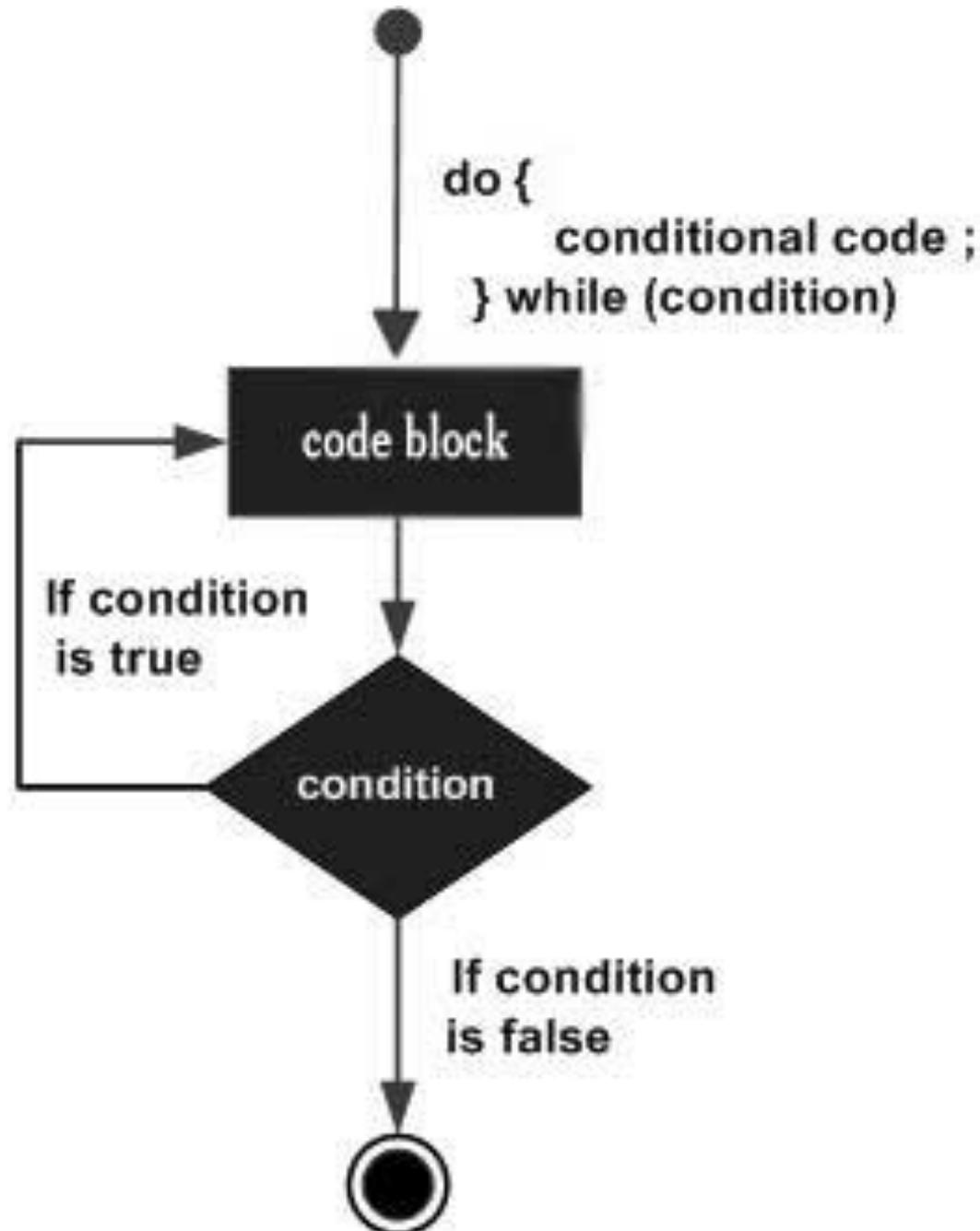
// Hit kommer vi när break exekveras
```

# Nästlade loopar

```
let table = '';
for (let row = 1; row <= 10; row++) {
  for (let col = 1; col <= 10; col++) {
    table += col * row + '\t';
  }
  table += '\n';
}
console.log(table);
```

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

# Do-While-satsen



# Do...While – Exempel

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



# Rekursion

- **Rekursion** uppstår när någonting definieras i termer av sig själv.
- Rekursion i JavaScript är förmågan att anropa en funktion från sig själv!  
*“self-calling-function”*

# En rekursiv funktion

```
function sum(n) {  
    if (n === 0) return 0;  
    else return n + sum(--n);  
}  
console.log(sum(5));
```



## Övning 1

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 10 miljoner kronor.

## Övning 2

Skriv ett program som skriver ut en tabell för talen 1 till 12.

På varje rad i tabellen skall talet visas liksom talet i kvadrat och talet i kubik.

x	$x^2$	$x^3$
1	1	1
2	4	8
3	9	81
...	...	...

# Övning 3

Skriv ett program som läser in ett godtyckligt antal tal större än 0 via prompt.

När användaren klickar på knappen Avbryt så skall programmet skriva ut det största och det minsta av de inlästa talen.

# Övning 4

- Skriv ett program som läser in ett heltal (n), via en prompt.  
Kontrollera att du får ett heltal och visa lämpliga felmeddelanden vid fel.
- Beräkna summan av n  
(dvs.  $1+2+3+4+\dots+n$ )  
Exempel: om talet n är 5 så skall programmet räkna ut  
 $1+2+3+4+5 = 15$
- Tips! Använd **Number.isInteger()**

[https://www.w3schools.com/jsref/jsref\\_isinteger.asp](https://www.w3schools.com/jsref/jsref_isinteger.asp)

# Övning 5

Skriv ett program som skriver ut heltalen från 1 till 100.

Men programmet ska visa ordet "Fizz" när talet är jämnt delbart med 3.

Och ordet "Buzz" när talet är jämnt delbart med 5.

Och ordet "FizzBuzz" när talet är jämnt delbart med 3 och 5.

Tips

<http://rosettacode.org/wiki/FizzBuzz>

1  
2  
Fizz  
4  
Buzz  
Fizz  
7  
8  
Fizz  
Buzz  
11  
Fizz  
13  
14  
FizzBuzz  
16

# Extra övningar

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_loops1](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_loops1)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_loops2](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_loops2)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_loops3](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_loops3)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_loops4](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_loops4)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_loops5](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_loops5)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_loops6](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_loops6)

# Summering av dagens lektion

- **Vi har idag jobbat med iterationer.**
- Reflektioner kring dagens lektion?
  - Vad tar du med dig från dagens lektion?
  - Finns det något som var extra svårt att förstå?
  - Finns det något som vi behöver repetera?
  - Hur upplevde du dagens arbetsmetoder?

# Framåtblick inför nästa lektion

- Läs: Bok 2, sid. 170–182
- Under nästa lektion kommer vi att DOM.