



JavaScript 1  
alert("Lektion 7");

Utbildare: Mahmud Al Hakim

NACKADEMIN

# Lektionstillfällets mål

- **Mål med lektionen**
  - Scope (räckvidd)
  - Arbeta med objekt
  - Objektliteral (literal notation)
  - Punktnotation (dot notation)
  - Egenskaper och metoder
  - Primitiva typer och referenstyper
- **Arbetsmetod**
  - Teori och praktik varvas under lektionen

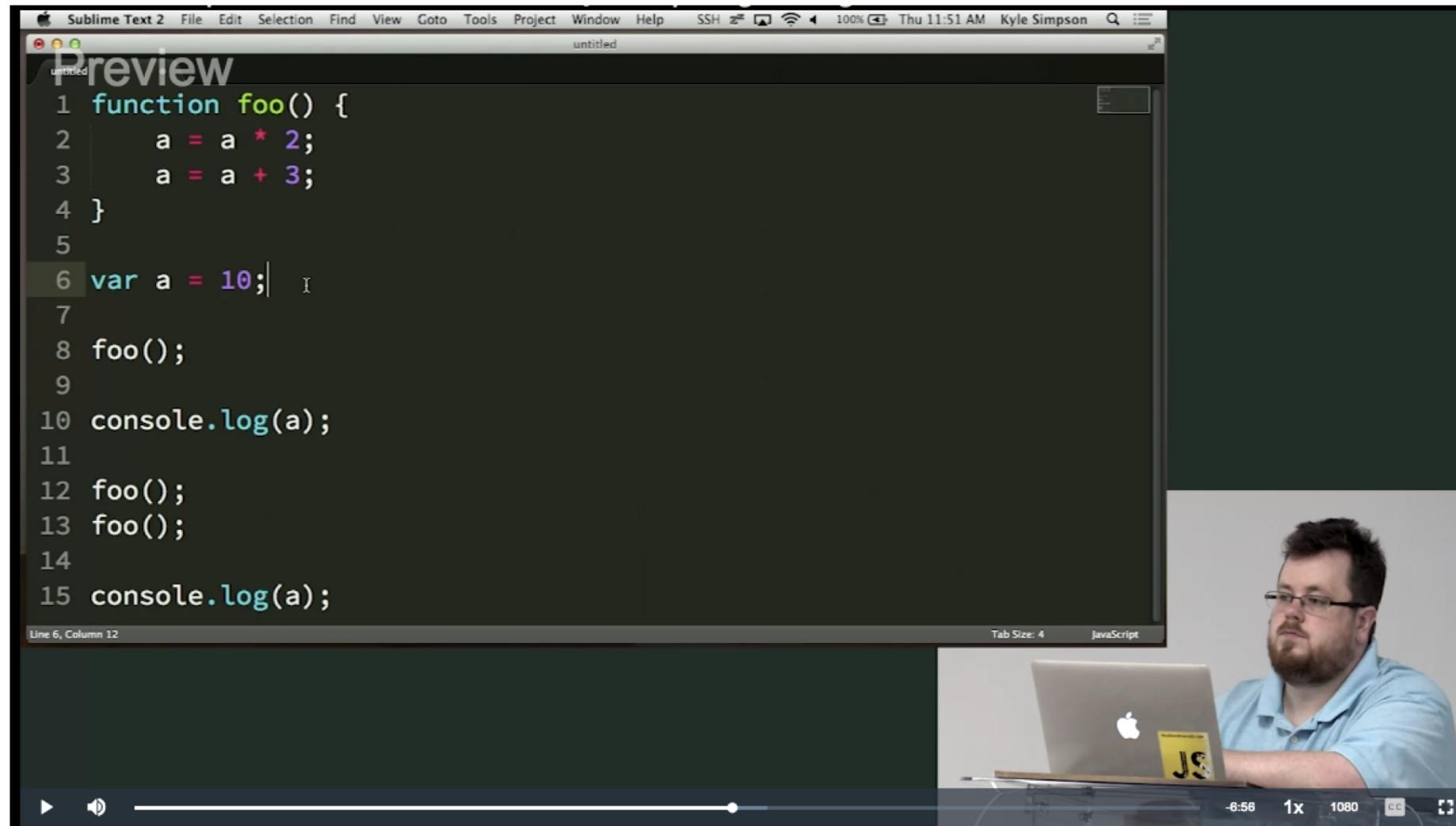
# Kort summering av föregående lektion

## Föregående lektion:

- Vi har jobbat med funktioner
- Parametrar och argument

# Repetition: Genomgång av funktioner med Kyle Simpson

<https://frontendmasters.com/courses/javascript-basics/values-types>



Tips: <https://github.com/getify/You-Dont-Know-JS>

# Scope (räckvidd)

- En variabel har något som kallas scope (räckvidd).
- Scope är de platser i koden där variabeln kan användas.
- I JavaScript finns det två typer av scope.
  1. Global scope
  2. Local scope.



# Local Scope – Lokala variabler

- Variabler som deklareras med `let` inne i kodblock, alltså inom `{ }` blir lokala variabler.

```
{  
    let x = 1;  
    console.log(x);  
}  
console.log(x); // ReferenceError: x is not defined
```

# OBS! Problem med `var`

```
{  
  var x = 1;  
  console.log(x);  
}  
console.log(x); // OK! var har ingen block-scope
```

# Variabler inne i funktioner är alltid lokala

```
function demo(){
    var x = 1;
    let y = 2;
    console.log(x);
    console.log(y);
}

demo();
console.log(x); // ReferenceError
console.log(y); // ReferenceError
```

# Global Scope – Globala variabler

```
var x = 1;
let y = 2;

function demo(){
    console.log(x); // OK. x är en global variabel
    console.log(y); // OK. y är en global variabel
}

demo();
console.log(x);
console.log(y);
```

# Global och lokal

```
let x = 1; // En global variabel

function demo(){
    let x = 2; // OBS! En ny lokal variabel
    console.log(x);
}

demo();
console.log(x); // skriver ut 1
```

# Vad är 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. Objektliteral

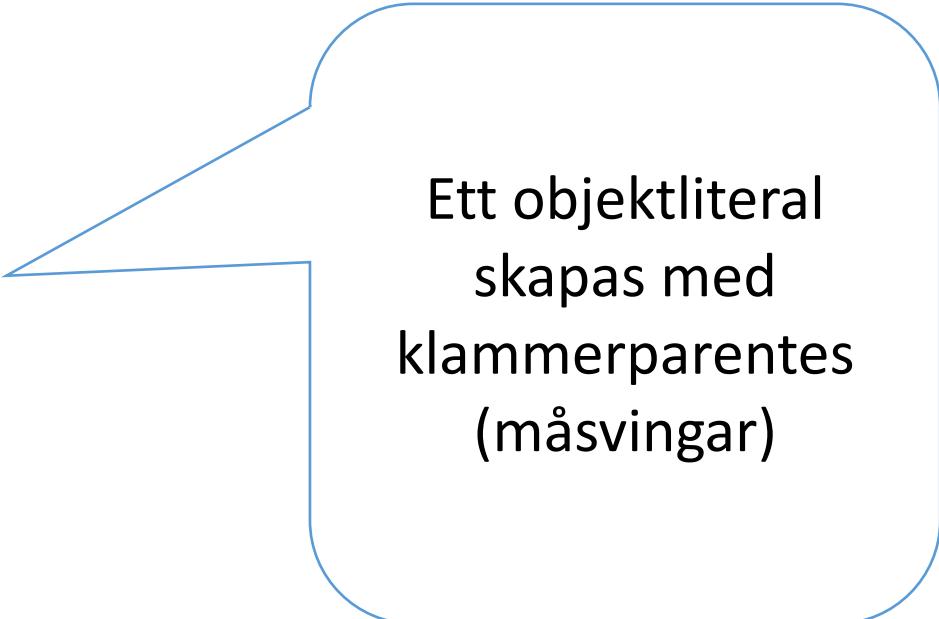
```
let car = {};
```

## 2. Objektkonstruktör (constructor)

```
let car = new Object();
```

# Att skapa ett objektliteral "Literal notation" eller "object initializer"

```
let mahmud = {};
```



Ett objektliteral  
skapas med  
klammerparentes  
(måsvingar)

# Nyckel-värde-par (name:value pairs)

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

Nyckelordet **this**  
refererar till det  
aktuella objektet



```
var hotel = {
```

```
  name: 'Quay',  
  rooms: 40,  
  booked: 25,  
  gym: true,  
  roomTypes: ['twin', 'double', 'suite'],
```

PROPERTIES  
These are variables

```
  checkAvailability: function() {  
    return this.rooms - this.booked;  
  }
```

METHOD  
This is a function

```
};
```

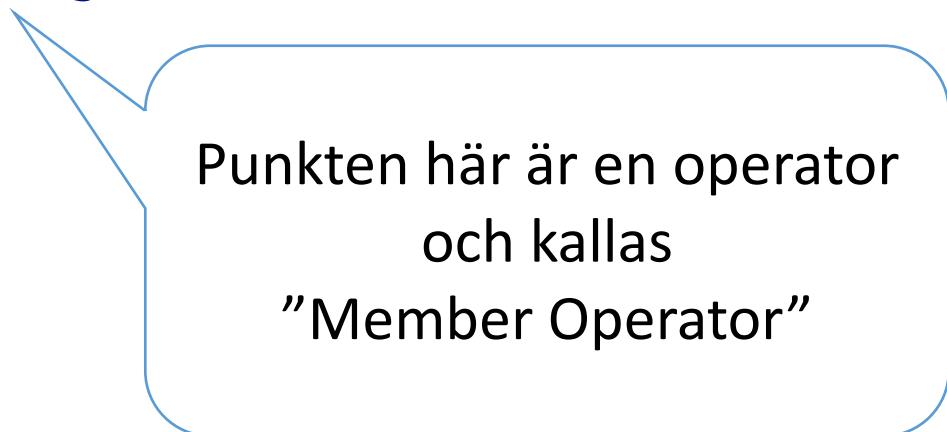
# Punktnotation (dot notation)

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

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

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

```
mahmud.info();
```



Punkten här är en operator  
och kallas  
"Member Operator"

The diagram illustrates the structure of JavaScript dot notation. It shows two lines of code: `var hotelName = hotel.name;` and `var roomsFree = hotel.checkAvailability();`. Above the first line, a bracket labeled "OBJECT" spans the word "hotel". Above the second line, another bracket labeled "PROPERTY/METHOD NAME" spans ".name;". A vertical line labeled "MEMBER OPERATOR" points down to the dot character in both examples.

```
var hotelName = hotel.name;
var roomsFree = hotel.checkAvailability();
```

OBJECT

PROPERTY/METHOD NAME

MEMBER OPERATOR

# Ändra objektets egenskaper/metoder

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

```
mahmud.name = "Mahmud Al Hakim"  
mahmud.age++;
```

```
console.log(mahmud);
```

# Ta bort en egenskap/metod

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

# Lägg till en egenskap/metod

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

```
mahmud.singel = false;
```

# Ta bort ett objekt

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

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

# Övning – Skapa följande objekt

Object	Properties	Methods
	car.name = Fiat	car.start()
	car.model = 500	car.drive()
	car.weight = 850kg	car.brake()
	car.color = white	car.stop()

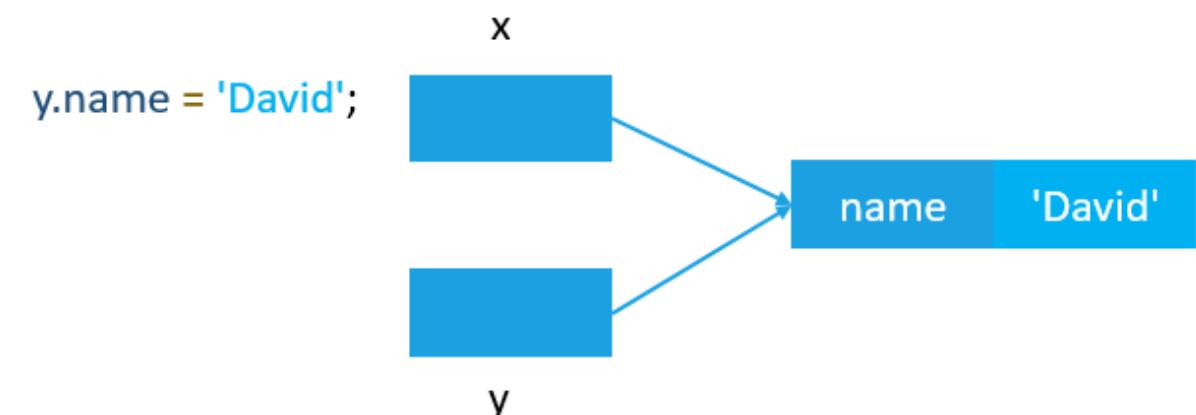
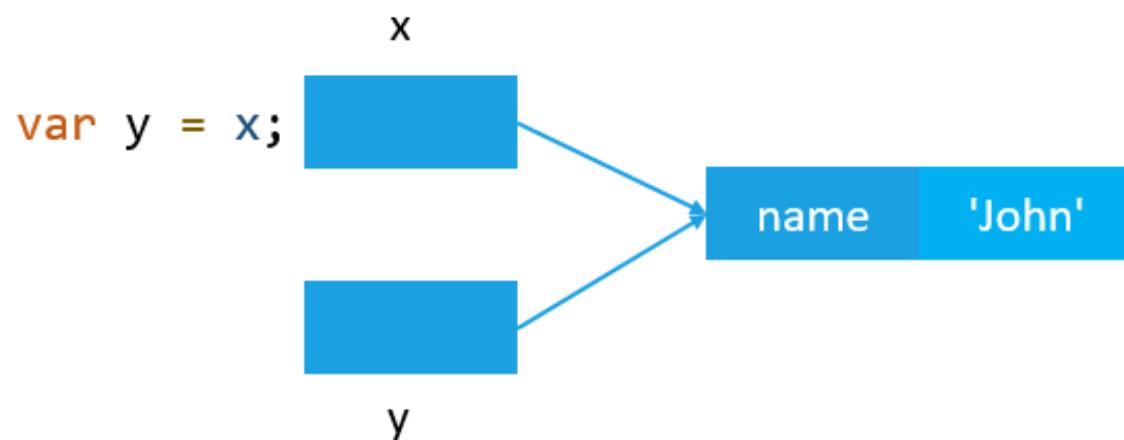
# Primitiva typer vs Referenstyper

- Med primitiva datatyper menas de grundläggande typer som tillhandahålls direkt av JavaScript, såsom string och number.
- Allt annat i JavaScript är referenstyper.
- Typen kallas "Object".
- Objekt är alltså en referenstyp.
- OBS! Arrayer är objekt i JavaScript!

# Referensstyper

```
let person = {};
let cars   = [];
console.log(typeof(person)); // object
console.log(typeof(cars));  // object
```

# Arbeta med referenser



Bildkälla: <http://www.javascripttutorial.net/javascript-primitive-vs-reference-values/>

# Kopiera referenser

```
let x = {name: 'John'};
```

```
let y = x;
```

```
console.log(x);
```

```
console.log(y);
```

```
x.name = "David";
```

```
console.log(x);
```

```
console.log(y);
```

“JavaScript  
Objects are  
Mutable”

# Kopiera objekt

- Metoden **assign()** används för att kopiera objekt \*1

```
let obj1 = { a: 1 };
let copy1 = Object.assign({}, obj1);
```

- Enklare syntax fr.o.m. ES2018 \*2

```
let copy2 = {...obj1};
```

1. OBS! Ej Deep Clone (Överkurs). Läs mer på MDN

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Object/assign](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/assign)

2. Spread syntax

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread\\_syntax](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax)



## KISS-principen

- KISS-principen (akronym för engelska Keep It Simple, Stupid!)
- KISS-principen går ut på att enkel design ska vara ett mål att försöka uppnå, och att man ska undvika komplex design.

# 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
  - Miltal
  - Tillverkningsår
  - Bränsle

# Övning 3

- Skapa en funktion som byter plats mellan två olika variabler.
- Kalla funktionen swap.

Tips:

<https://www.abeautifulsite.net/swapping-variables-with-javascript>

# Extra övningar

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_objects1](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_objects1)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_objects2](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_objects2)

[https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_objects3](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_objects3)

# Summering av dagens lektion

- **Vi har idag jobbat med objekt i JavaScript.**
- 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. 98–107
- Under nästa lektion kommer vi att fördjupa oss lite mer i arrayer och objekt.
- Lycka till med inlämningsuppgift 2.



NACKADEMIN