

# JavaScript

Niklas Hjelm

# Vad är JavaScript?

- JavaScript är ett översatt objektorienterat programmeringsspråk.
- Uppfanns av Brendan Eich på Sun Microsystems för webbläsaren Netscape 2 1995.
- Fick sitt namn och sin syntax på grund av programmeringsspråket Java's ökande popularitet.
  - [Java-reklam](#)
- 1997 sattes standarden ECMA-262
  - Gemensam standard som skulle stödjas av alla webbläsare

# Vad är ECMAScript?

- ECMAScript är den del av JavaScript som kan tolkas av alla webbläsare.
- JavaScript är alltså inte 100% ECMAScript
  - En interpreter för JavaScript kan tolka ECMAScript
  - En interpreter för ECMAScript måste inte kunna tolka JavaScript
- Sedan 2018 har alla webbläsare stöd för standarden ES6
  - Ej Internet Explorer 11, eftersom Microsoft bytt webbläsare till Edge.

# Översatta språk

- JavaScript är ett s.k. översatt språk (Interpreted Language)
- Detta innebär att till skillnad från språk som C# och Java så körs koden utan att kompileras.
- JavaScript körs av en Interpreter.
  - V8 av Google som används av alla Chromium-webbläsare
    - Chrome, Edge mfl.
  - SpiderMonkey av Mozilla för Firefox
  - JavaScriptCore av Apple för Safari
  - Chakra av Microsoft för Internet Explorer och första versionerna av Edge

# Dynamisk typning

- JavaScript är ett Dynamiskt typat språk.
- Datatyper finns fortfarande
  - String
  - Number
  - Bigint
  - Boolean
  - Undefined
  - Null
  - Symbol
  - Object

## Objekttyper

- Array
- Date
- Object

# Variabler

- I JavaScript finns det tre olika sätt att deklarerera variabler.
  - **var** – Äldre variant av variabeldeklaration
  - **let** – Det nya sättet att deklarerera variabler
  - **const** – Används för konstanter
- Let vs. Var.
  - Var är inte block scoped, kan omdeklareraras.
  - Let beter sig som en variabel i andra språk som C#.
- Grundregel: Deklarera variabler som const, om variabeln kommer behöva byta värde använd let.
  - Endast om man vill försäkra sig om att alla äldre webbläsare ska kunna köra koden ska man använda var.

# Konsolen i webbläsaren

- Webbläsarens konsol kan köra JavaScript direkt.
- Öppnas genom att trycka på **f12** alternativt **ctrl+shift+i**
- Testa att öppna konsolen.
  - Deklarera en variabel `let letVar = 2;`
  - Skriv sedan ut värdet med `console.log(letVar);`
  - Gör sedan samma sak med const.
  - Prova att ändra värde på båda och se vad som händer.
- Om man är tveksam kan man alltså använda konsolen för att testa saker.

# Script-tag och file

- JavaScript kan inkluderas på en hemsida med hjälp av elementet `<script>`
- Ett script kan antingen skrivas direkt i html-filen inom elementet eller i en egen fil, länkat i ett script-element
  - `<script src="./script.js"></script>`
  - Dessa måste ligga längst ner i `<body>` eller i `<head>`.
  - Om man lägger det i head bör man använda attributet `defer`
  - Det gör så att scriptet körs efter att alla element är skapade.



# Demo

# Flödeskontroll

- If – samma som vi är vana.
- Else/else if – också som vi är vana

• **MEN**

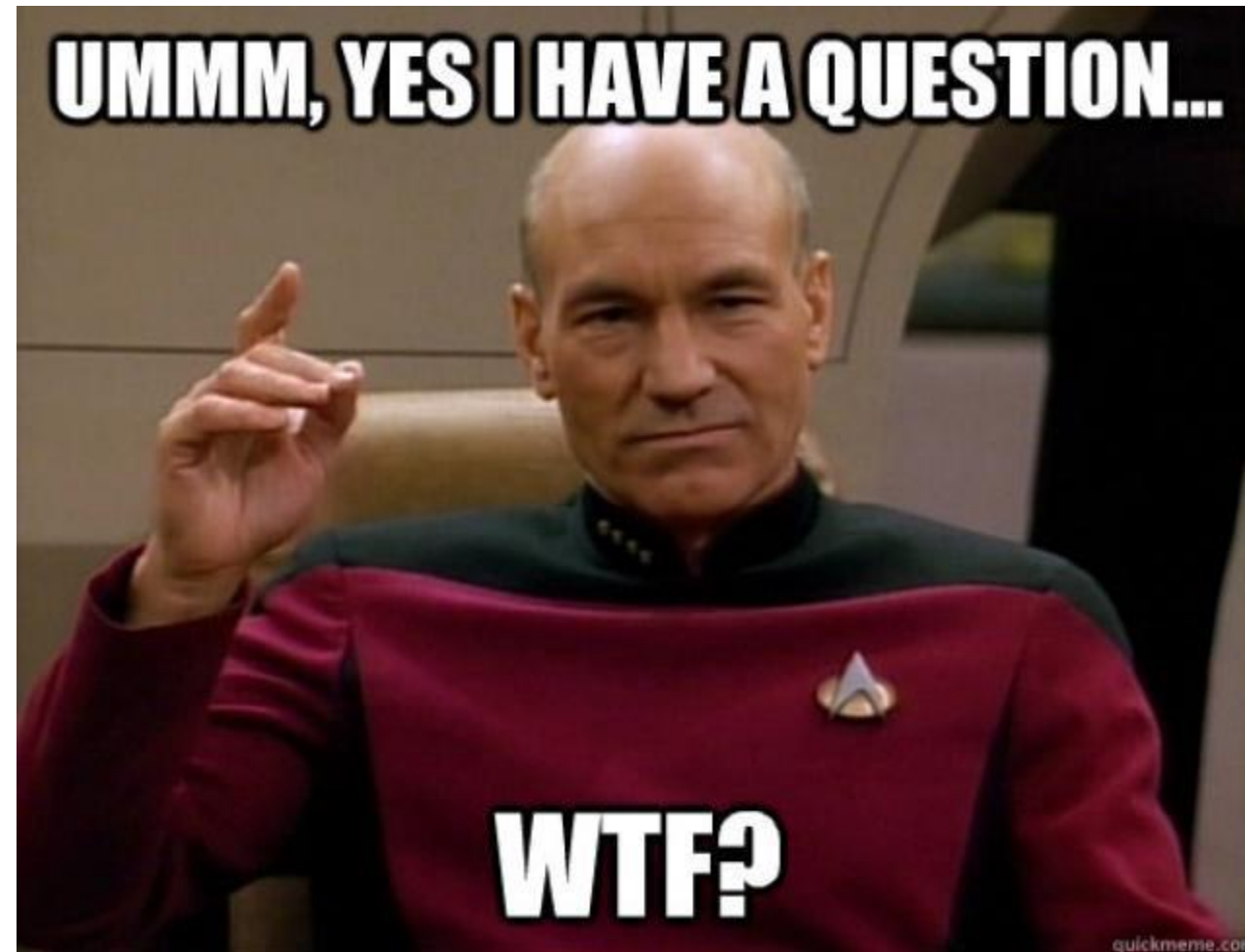
```
if (villkor) {  
    //gör något  
}  
else if (annat_villkor) {  
    //gör något annat  
}  
else {  
    //gör något om inget annat är sant  
}
```

**==** **!=** **==**

**==** **==** **===**

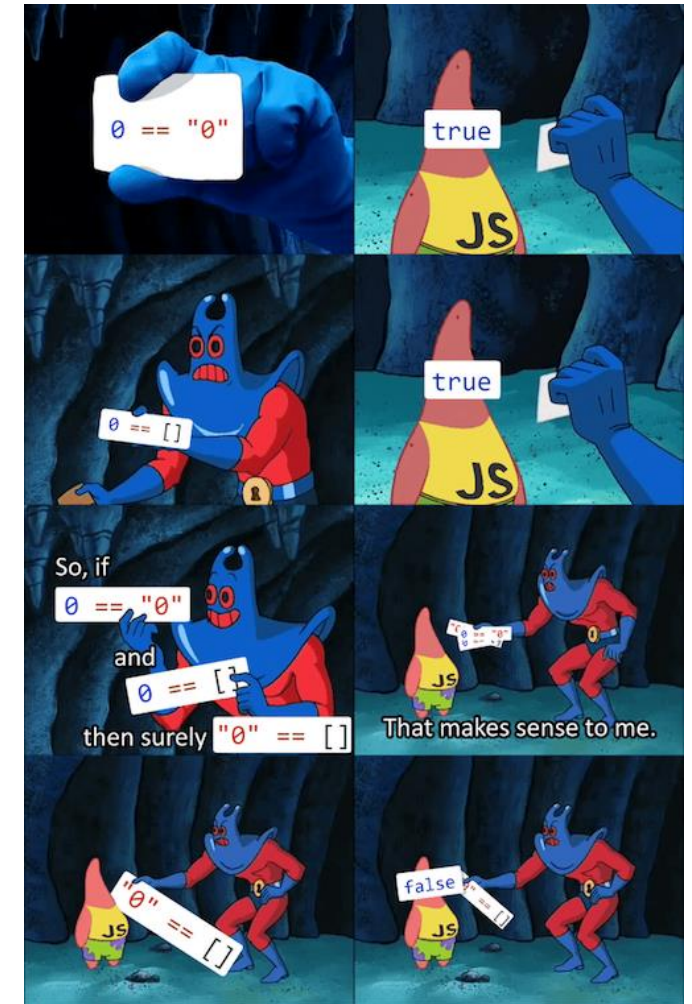
**11** **==** **"11"** : **true**

- **==** jämför värden men inte datatyp
- **===** jämför värde och datatyp
- Det samma gäller inte lika med **!=**

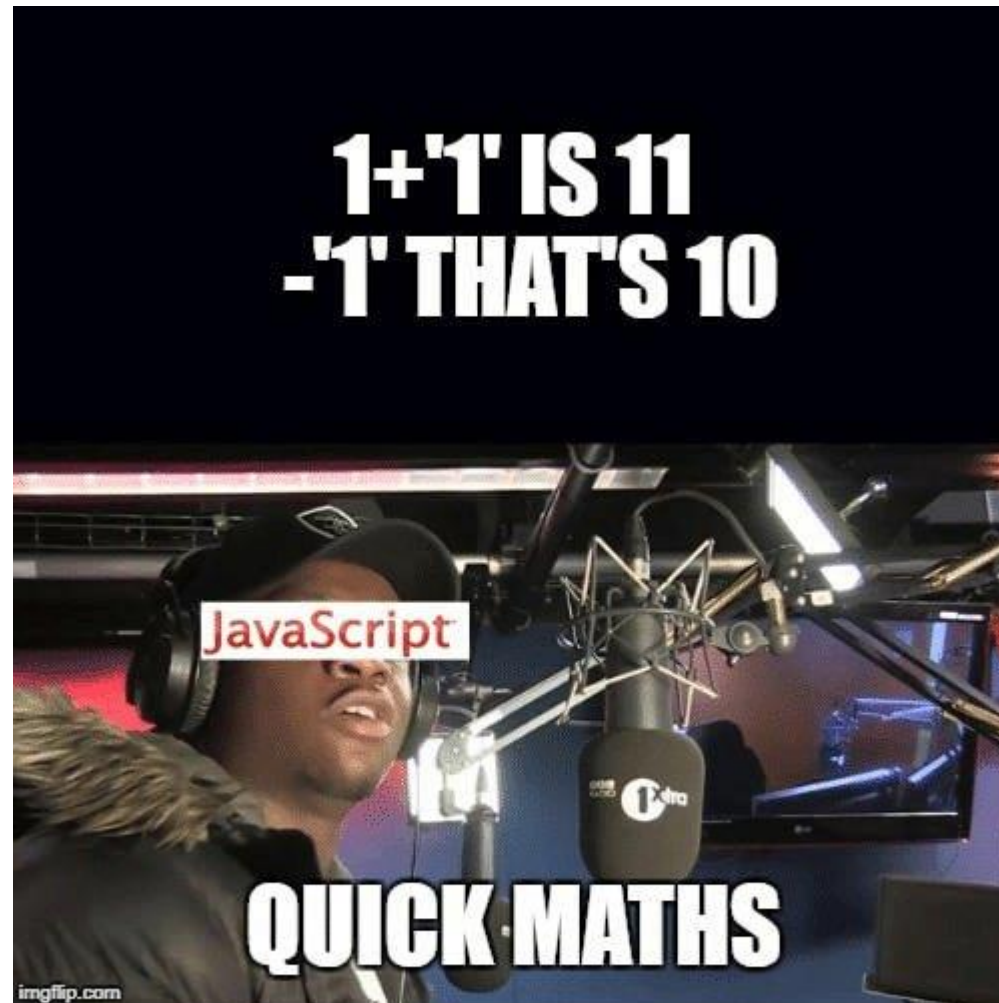


# Type coercion (typförvrängning)

- Eftersom typen på en variabel valideras när koden körs så "hjälper" JavaScript till och försöker matcha datatyp i ett villkor eller i en operation.
- Detta gör att om vi vill kolla om något är samma värde OCH typ så använder vi ===



<code>" " == 0;</code>	→	<code>true</code>
<code>1 == [1];</code>	→	<code>true</code>
<code>11 + "1";</code>	→	<code>111</code>
<code>"11" - "1";</code>	→	<code>10</code>
<code>[] - "1";</code>	→	<code>-1</code>
<code>{ } + [ ];</code>	→	<code>0</code>
<code>{ } + { };</code>	→	<code>NaN</code>
<code>{ } - " ";</code>	→	<code>-0</code>



"Logic"



IT-HÖGSKOLAN  
GÖTEBORG

# Switch

```
switch (letVar) {  
  case 1:  
    console.log("apa");  
    break;  
  case 2:  
    console.log("bepa");  
    break;  
  case 3:  
    console.log("cepa");  
    break;  
  default:  
    console.log("wooo");  
    break;  
}
```



# Loopar

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

```
while (true) {  
  console.log("troll");  
}
```

# Arrayer

- Definiera alltid arrayer som const
  - `const arr = ["a", "b", "c"];`
- Dynamisk typning gör att en array kan innehålla olika typer på olika platser i samma array.
  - `const arr = ["a", 1, true];`
  - `const arr = ["a", 1, [true, 0.1]];`

```
> null > -1  
⏪ true
```



**JavaScript is a pathway to many abilities some consider to be unnatural**

# For in

- For in itererar över element i en array
- Men den itererar över index

```
const arr = ["a", "b", "c"];  
for (const key in arr) {  
  console.log(key);  
}
```

0  
1  
2

# For of

- For of itererar över element i en array
- Fungerar som ni är vana att foreach gör i C#

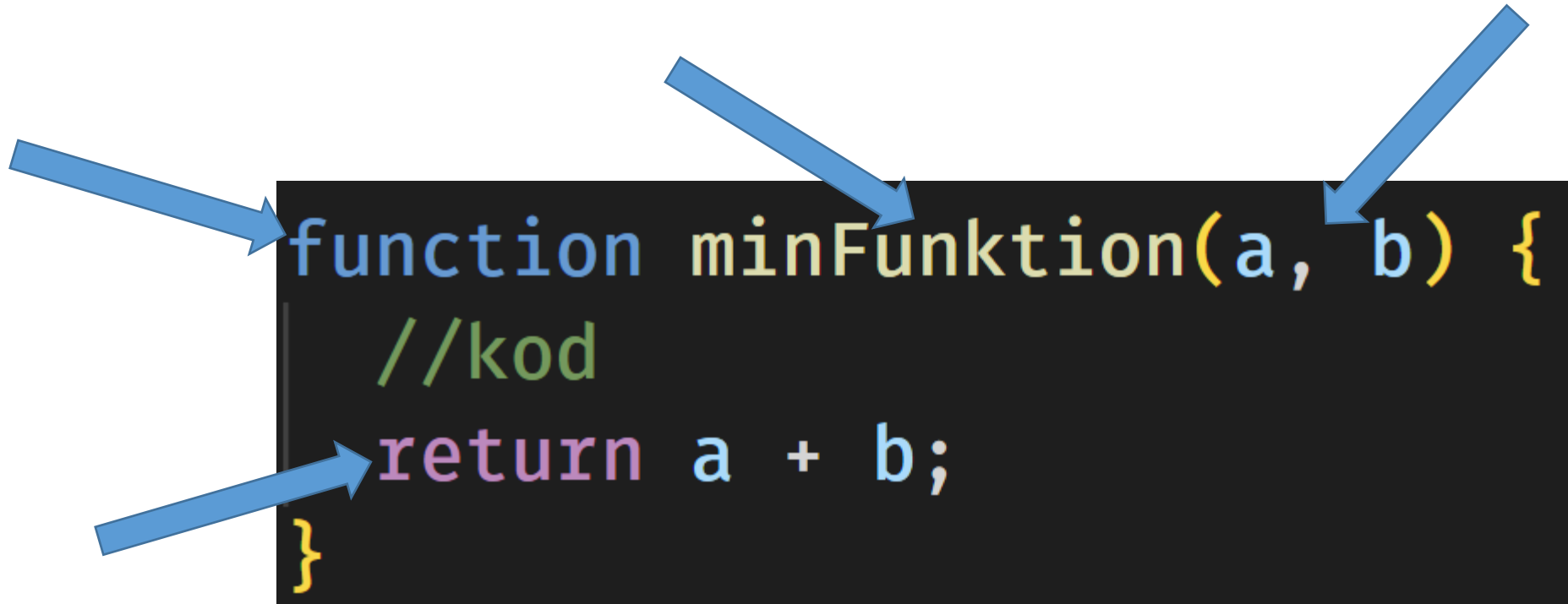
```
const arr = ["a", "b", "c"];  
for (const key of arr) {  
  console.log(key);  
}
```

"a"  
"b"  
"c"

# Demo

# Funktioner

- Funktioner i JavaScript är ungefär som metoder är i C#



```
function minFunktion(a, b) {  
  //kod  
  return a + b;  
}
```

# Arrow functions

- Precis som vi har lambda-funktioner i C# så finns det en motsvarighet i JavaScript. De kallas Arrow-Functions

```
const c = (a, b) => {  
  return a + b;  
};  
console.log(c(1, 2));
```



A meme featuring Morpheus from the movie The Matrix. He is sitting in a large, tufted red leather chair, wearing his signature black sunglasses and a dark, high-collared coat. The background is dark and out of focus.

**WHAT IF I TOLD YOU**

**JAVASCRIPT HAS LINQ**

# Array-metoder

- En array har många färdiga funktioner.
- Några av de vanligaste är:
  - toString() och join()
  - indexOf(), lastIndexOf() och find()
  - pop() och push()
  - shift() och unshift()
  - concat()
  - slice() och splice()
- [https://www.w3schools.com/jsref/jsref\\_obj\\_array.asp](https://www.w3schools.com/jsref/jsref_obj_array.asp)

# Demo

