

JavaScript 1
alert("Lektion 9");

Utbildare: Mahmud Al Hakim

NACKADEMIN

Lektionstillfällets mål

- **Mål med lektionen**
 - Arbeta med globala JS-Objekt
 - 1. String
 - 2. Number
 - 3. Math
 - 4. Date
- **Arbetsmetod**
 - Teori och praktik varvas under lektionen

NACKADEMIN

Kort summering av föregående lektion

Föregående lektion:

- Vi har jobbat med konstruktorer.
- Att skapa arrayer av objekt.
- Lite med BOM och DOM.

Repetition: Object Constructor Function av [Rob Merrill](#)
https://youtu.be/oSs_25dmxOE

OBJECT CONSTRUCTOR FUNCTION

A constructor is useful when you want to create multiple similar objects with the same properties and methods. The code creates objects as an instance of it.



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 the String object. The tooltip is displayed over the variable 'firstName'. It shows the property name 'length' with a small icon, the type '(property) String.length: number', and a blue information icon. Below the tooltip, a list of other string methods is visible: link, localeCompare, match, normalize, repeat, replace, search, slice, small, split, and startsWith. Each method is preceded by a small purple cube icon.

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

Strängkonkatenering

```
// Du kan använda metoden concat() istället för +  
let firstName = 'Mahmud';  
let lastName = 'Al Hakim';  
let fullName = firstName.concat(' ' + lastName);  
console.log(fullName);
```

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 slice()

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

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

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

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

Substr()

```
let myName = 'Mahmud Al Hakim';  
  
// Returnera en sträng från index 7  
// men enbart 2 tecken efteråt  
let al = myName.substr(7,2);
```

Strängar indexeras på samma sätt som arrayer

```
let myName = 'Mahmud Al Hakim';
console.log(myName.length);
console.log(myName[0]);                                // M
console.log(myName[1]);                                // a
console.log(myName.charAt(1));                          // a
console.log(myName[myName.length-1]); // m
console.log(myName.indexOf('m'));                      // 3
console.log(myName.indexOf('x'));                      // -1
console.log(myName.lastIndexOf(' ')); // 9
```

Split

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

Trim

```
let fullname = ' Mahmud Al Hakim      '; // OBS! extra blanka tecken
let nameArray = fullname.split(' ');
console.log(nameArray);
// [ '', 'Mahmud', 'Al', 'Hakim', '', '', '' ]  
  
// Lägg till metoden trim() för att rensa blanka tecken
nameArray = fullname.trim().split(' ');
console.log(nameArray);  
  
// OBS! Trim rensar enbart start och slut, ej mitten!
```

Replace

```
let fullname = 'Mahmud Al Hakim';
fullname = fullname.replace('Al Hakim' , 'Al–Hakim')
```

```
// Tips. Rensa alla mellanslag i en sträng
fullname = "Mahmud      Al      Hakim      ";
fullname = fullname.replace(/ +/g, ' ');
```

Ovanstående teknik kallas ”RegExp”. OBS! Överkurs

https://www.w3schools.com/js/js_regexp.asp

Escape-tecken

Code	Result	Description
\'	'	Single quote
\\"	"	Double quote
\\"\\	\	Backslash

Escape-tecken – Exempel

```
let msg = '';
msg += "The character \\ is called backslash.";
msg += '\n'; // New Line
msg += "We are the so-called \"Vikings\" from the north.";
msg += '\n';
msg += 'It\'s alright.';
console.log(msg);
```

Övning

Skapa en funktion som har en parameter.

- Ange funktionsnamnet **capitalize**

- Funktionen ska ta en sträng och ändra den första bokstaven till versal.

Exempel på anropet av funktionen

```
let helloMsg = 'hej och välkommen.'
```

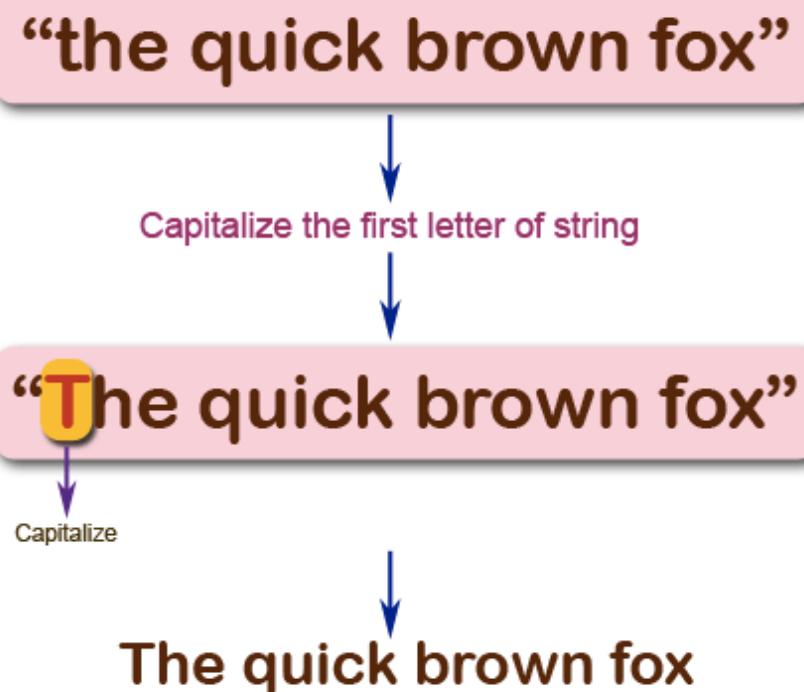
```
console.log(capitalize(helloMsg));
```

- // Skriver ut

- // Hej och välkommen.

Lösningsförslag

```
capitalize = function(str){  
    return str.charAt(0).toUpperCase() + str.substring(1);  
}
```



Övningar

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings1

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings2

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings3

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings4

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings5

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings6

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings7

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings8

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_strings9

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
```

Objektet Number – Några egenskaper

```
// MAX_VALUE ger största möjliga positiva talet som kan lagras i en variabel  
console.log(Number.MAX_VALUE); // 1.7976931348623157e+308
```

```
// OBS! Om du försöker lagra större tal  
console.log(Number.MAX_VALUE * 2); // Infinity
```

```
// För att få minsta möjliga negativa talet lägg till minusstecken  
console.log(-Number.MAX_VALUE); // -1.7976931348623157e+308
```

```
console.log(Number.MIN_VALUE); // 5e-324 = the value closest to 0
```

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
```

isNaN()

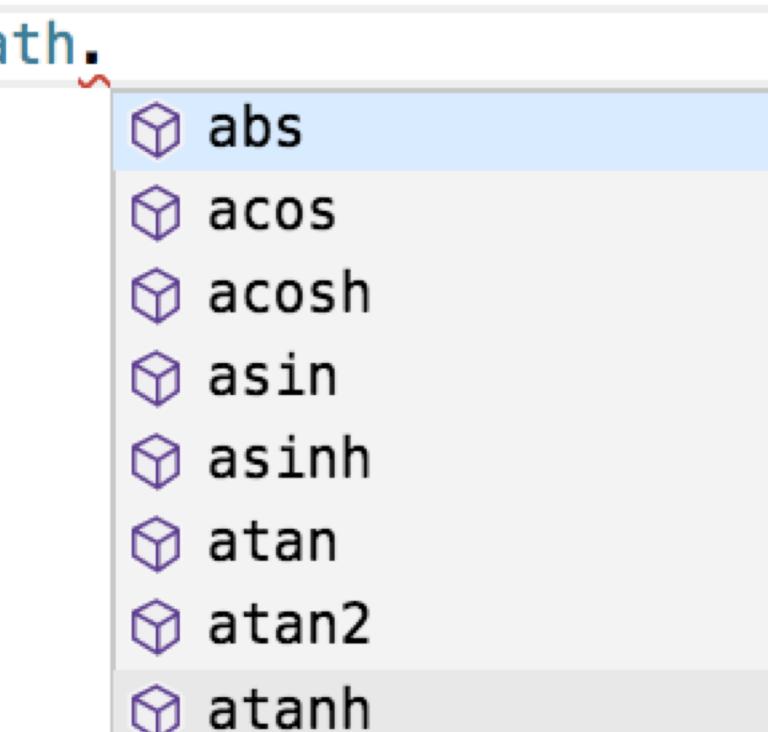
```
let x;  
x = x + 2;  
console.log(x);           // NaN  
console.log(Number.isNaN(x)); // true
```

Objektet Math

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

Math

Math.



A screenshot of a code editor showing a dropdown menu of the properties and methods available on the Math object. The menu is triggered by the code 'Math.' followed by a red underline. The items listed are: abs, acos, acosh, asin, asinh, atan, atan2, and atanh. Each item has a small purple icon to its left.

- abs
- acos
- acosh
- asin
- asinh
- atan
- atan2
- atanh

Math – Några egenskaper

```
console.log(Math.PI);  
console.log(Math.E);  
console.log(Math.SQRT2);
```

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 potenser (x^3)

```
console.log(Math.pow(1, 3)); // 1
console.log(Math.pow(2, 3)); // 8
console.log(Math.pow(3, 3)); // 27
console.log(Math.pow(4, 3)); // 64
console.log(Math.pow(5, 3)); // 125
```

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));
```

Ceil och floor – närmaste heltal

```
console.log(Math.ceil(1.1));  
console.log(Math.ceil(1.5));  
console.log(Math.ceil(1.9));  
console.log(Math.floor(1.1));  
console.log(Math.floor(1.5));  
console.log(Math.floor(1.9));
```

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
```

Övning

Skapa en funktion som har en parameter.
Ange funktionsnamnet **maxArray**.
Funktionen ska ta en array av ett
godtyckligt antal tal.
Funktionen ska returnera det största
talet i arrayen.

Exempel vid anropet

```
console.log(maxArray( [10,4,5] ));  
// 10  
console.log(maxArray( [4,5,15] ));  
// 15
```

Lösningsförslag

```
function maxArray(arr){  
    return Math.max(...arr);  
}
```

Tips:

<https://codeburst.io/javascript-the-spread-operator-a867a71668ca>

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

Övningar

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_math1

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_math2

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_math3

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_math4

Objektet Date

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

Date



Arbeta med datum

```
let datum = new Date();
console.log(datum.getFullYear()); // YYYY t.ex. 2018
console.log(datum.getMonth());    // 0-11
console.log(datum.getDate());     // 1-31
console.log(datum.getDay());      // 1-6
```

Arbeta med tid

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

toDateString och toTimeString

```
console.log(datum.toDateString());  
// Visar t.ex. Thu Oct 11 2018
```

```
console.log(datum.toTimeString());  
// Visar t.ex. 10:14:19 GMT+0200
```

Övningar

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dates1

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dates2

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dates3

https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_dates4

Övning 1

Visa aktuellt datum
i detta format
YYYY-MM-DD
t.ex.
2018-10-12

OBS! Använd **inte** metoden
`toString()`

Övning 2

Visa klockan
i detta format
HH:MM:SS
t.ex.
14:05:10

Tips! Använd metoden
`toTimeString()`

Övning 3

Visa datum i detta format

Dag den xx månad år

t.ex.

Onsdag den 17 oktober 2018

Tips: Använd arrayer för att
lagra veckodagar och
månader.

Övning 4

Skapa en live digital klocka
och visa den i webbläsaren!

Tips!

<https://stackoverflow.com/questions/39418405/making-a-live-clock-in-javascript>

Övning 5

- Avståndsformeln kan användas för att beräkna avståndet mellan två punkter i ett koordinatsystem.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Skriv ett program som läser in två punkter och beräknar avståndet mellan punkterna.

Summering av dagens lektion

- Vi har idag jobbat med några viktiga globala JS-Objekt.
- String, Number, Math och Date.
- 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. 128-144
- Under nästa lektion kommer vi att jobba med **selektioner**.
- Lycka till med inlämningsuppgift 3



NACKADEMIN