

Systemutveckling

PHP

Föreläsning 16 - JavaScript

Mikael Olsson
mikael.olsson@emmio.se
076-174 90 43



Dagens ämnen

- Datatyper
- Jämförelseoperatorer
- Funktioner
- DOM
- Array-funktioner
- jQuery
- JSON
- API
- Node.JS
- npm
- webpack
- Continuous Integration

Vad är JavaScript?

- Scriptspråk som används för att skapa dynamiska hemsidor
- Exekveras i webbläsarens JavaScriptmotor, behöver inte kompileras
- Standarden av JavaScript kallas ECMAScript.

Variabler

- En del språk kräver att man berättar vilken typ av data variabeln innehåller, t ex sträng, tal osv. De språken kallas typade eller starkt typade.
- JavaScript är ett svagtypat språk, vi behöver inte berätta vilken typ av information vi vill spara i en variabel.

Typning

- Statiskt typade språk ex:

```
int x = 5;  
x = "hej"; //kompileringsfel!
```

- Dynamiskt (checkas i run-time) & löst (går att byta) typade språk ex:

```
var x = 5; //x är av typen int  
x = "hej"; //x är nu av typen string, typen  
ändras i run-time
```

Datatyper

- `number` för nummer, både heltal och decimaltal.
- `string` för strängar/texter/bokstäver.
- `boolean` för sant/falskt.
- `null` för okända värden.
- `undefined` för värden som inte har tilldelade värden.
- `object` för mer komplexa datastrukturer.
- `symbol` för unika identifierare.
- Array räknas som objekt i JavaScript.
- <https://javascript.info/types>

Variabler - deklaration

```
let firstName = "Mikael"; //firstName är en sträng "Mikael"  
let lastName = ""; //lastName är en tom sträng  
let age; // age är undefined  
lastName = "Olsson"; //lastName är "Olsson"
```

Tilldelning

- `var` - deklarerar en variabel, ev med ett värde. Variabeln finns globalt eller inom en hel funktion.
- `let` - deklarerar en variabel som alltid har ett lokalt scope, ev med ett värde. Variabeln existerar inte utanför blocket, statementet eller uttrycket som det skapades i.
- <http://jsfiddle.net/9ghwc5st/>
- `const` - deklarerar en konstant som inte kan ändras

Jämförelseoperatorer

- "truthy" vs "falsy" jämförelser:
- ==
- !=
- <
- <=
- >=
- >
- vanlig jämförelse mellan datatyper
- ===
- !==
- The identity operator returns true if the operands are strictly equal with no type conversion.
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Comparison_Operators

Aritmetiska operatorer

- Användbara aritmetiska operatorer

+

-

*

/

%

++

--

+=

Aritmetiska operationer - tal

```
let a = 1;
```

```
let b = 2;
```

```
let c = a + b; // c = 3
```

Aritmetiska operationer - sträng

+ används för att konkatenera (slå ihop) strängar

```
let firstName = "Mikael";
```

```
let lastName = "Olsson";
```

```
let name = firstName + lastName; // name = "MikaelOlsson"
```

```
let fullName = firstName + " " + lastName; // fullName = "Mikael Olsson"
```

console.log() och felhantering

console.log(), används för att skriva ut variabler eller element i konsolen i webbläsaren

Användbart under utvecklingen och vid felhantering

```
let firstName = "Mikael";  
  
console.log(firstName); //Skriver ut "Mikael"  
console.log("firstName = " + firstName); //Skriver ut  
"firstName = Mikael"
```

Interaktion med HTML/CSS

- Importeras ungefär som CSS-filer i HTML-koden
- Kan importeras i head eller nära `</body>`, för bättre performance.
- Kan "nå" HTML-elementen och ändra dem.
- JavaScript kan "triggas" av användaren genom HTML-sidan

Interaktion med HTML/CSS

Importerera JavaScript-filer i HTML

```
<!DOCTYPE html>
<html>
  <head>
    <script src="script1.js"></script>
  </head>
  <body>

    <script src="script2.js"></script>
  </body>
</html>
```

Interaktion med HTML/CSS

Nå HTML element från JavaScript-filen

```
document.addEventListener("DOMContentLoaded", function(event) {  
  let headline = document.getElementById("headline");  
  let text = headline.innerHTML;  
});
```


Interaktion med HTML/CSS

Ändra HTML element från JavaScript-filen

```
document.addEventListener("DOMContentLoaded", function(event)
{
    let headline = document.getElementById("headline");
    let oldText = headline.innerHTML;
    headline.innerHTML = "New Headline";
    headline.style.color = "#0000ff";
});
```

Interaktion med HTML/CSS

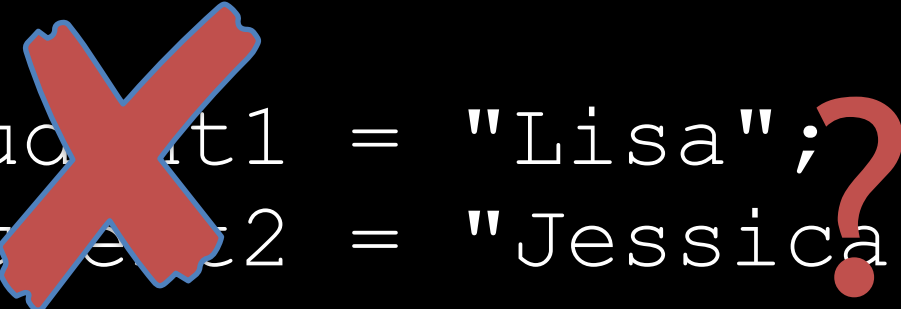
Hämta information från inputfält

```
document.addEventListener("DOMContentLoaded", function(event) {  
  let name = document.getElementById("name").value;  
  let acceptRules = document.getElementById("rules").checked;  
});
```

Värden som hör ihop

- Hur ska man göra om man har flera studenter?
- Begränsande!

```
let student = "Kalle";  
let student1 = "Lisa";  
let student2 = "Jessica";  
let student3 = "Lisa";
```

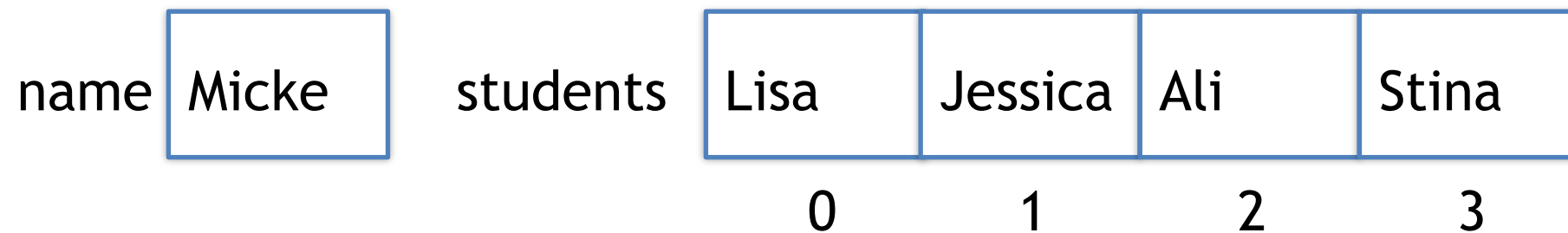


Array

En array är en speciell datatyp som kan innehålla flera värden.

```
let students = ["Lisa", "Jessica"];  
  
// Kan stå på flera rader  
  
let students = [  
    "Lisa",  
    "Jessica"  
];
```

Array



- En array är en speciell datatyp som kan innehålla flera värden.

```
let name = "Micke";  
let students = [  
  "Lisa",  
  "Jessica",  
  "Ali",  
  "Stina"  
];
```

Array

- *Uppgift:* Skapa en array och skriv ut varje värde i den.
- <http://jsfiddle.net/>

```
let students = [  
  "Lisa",  
  "Jessica",  
  "Ali",  
  "Stina"  
];  
  
console.log(students[0]);  
// osv...
```

Lisa



Avancerade variabler

- Saker i riktiga världen består ofta av olika variabler.
- Hänger de ihop på något sätt?

```
let name = "Micke";  
let age = 42;  
let shoe_size = 43;
```

Objekt

- *Objekt* är variabler med "undervariabler".

```
▶ {name: "Micke", age: 42, shoe_size: 43}
```

```
>
```

```
let person = {};  
  
person.name = "Micke";  
person.age = 42;  
person.shoe_size = 43;  
  
console.log(person);
```

```
let person = {  
  name: "Micke",  
  age: 42,  
  shoe_size: 43  
};  
  
console.log(person);
```


Avancerade variabler

- *Uppgift:* Skapa ett objekt som håller reda på en *kurs* med några egenskaper.

```
let person = {  
  name: "Micke",  
  age: 42,  
  shoe_size: 43  
};
```

Funktions beståndsdelar

- Funktioner är en kod-block som utförs när vi vill att den ska utföras.
- Funktioner kan ta emot data i så kallade parametrar.
- Funktioner kan kallas av JavaScript-kod eller när användaren gör något, t.ex. klickar på en knapp.

Funktioner

- Funktionens beståndsdelar

```
function addNumbers(a, b) {  
  console.log(a + b);  
}
```

```
addNumbers(2, 2);
```

- Namn
- Parametrar
- Innehåll
- Anrop

4



Funktioner

- En funktion kan även returnera ett värde.

```
function addNumbers(a, b) {  
  return(a + b);  
}  
  
let result = addNumbers(2, 2);  
  
console.log(result);  
console.log(addNumbers(3, 3));
```

4
6
>

Funktioner

- Ser vi något mönster?
- Vad händer om vi måste ändra i formeln?

```
let temp_f = [80, 75, 88];  
let temp_c;  
let tmp;
```

```
tmp = temp_f[0];  
temp_c = (5/9) * (tmp-32)  
console.log(temp_c);
```

```
tmp = temp_f[1];  
temp_c = (5/9) * (tmp-32)  
console.log(temp_c);
```

```
tmp = temp_f[2];  
temp_c = (5/9) * (tmp-32)  
console.log(temp_c);
```

Funktioner

```
function toCelcius(fahrenheit) {  
  let temp_c = (5/9) * (fahrenheit-32);  
  console.log(temp_c);  
}
```

```
let temp_f = [80, 75, 88];  
let tmp;
```

```
tmp = temp_f[0];  
toCelcius(tmp);
```

```
toCelcius(temp_f[1]);  
toCelcius(temp_f[2]);
```

26.666666666666668

23.88888888888889

31.111111111111114

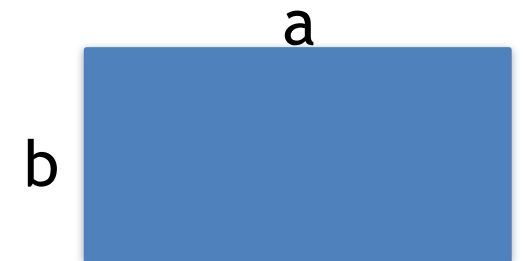
>

Funktioner

- *Uppgift:* Gör en funktion som räknar ut arean av en rektangel och returnerar resultatet.
- <http://jsfiddle.net/>

```
function toCelcius(fahrenheit) {  
  let temp_c = (5/9) * (fahrenheit-32);  
  console.log(temp_c);  
}
```

```
toCelcius(80);
```



Area = a * b

Mer om array

```
let students = [  
  "Lisa",  
  "Jessica",  
  "Ali",  
  "Stina"  
];  
  
let l = students.length;  
for (let i = 0; i < l; i++) {  
  console.log(students[i]);  
}
```


Mer om array

```
let students = [  
  "Lisa",  
  "Jessica",  
  "Ali",  
  "Stina"  
];  
  
students.forEach(function(element) {  
  console.log(element);  
});
```

Lägga till värde i array

```
let students = [  
  "Lisa",  
  "Jessica",  
  "Ali"  
];  
  
students.push("Stina");
```

Ta bort värde från array

- pop - tar bort från slutet av en array och returnerar värdet.
- shift - tar bort från början av en array och returnerar värdet.
- splice - tar bort från en array för ett specifikt index och returnerar värdet / värdena.
- filter - returnerar en ny array med filtrerade element från en array.
- <https://love2dev.com/blog/javascript-remove-from-array/>

Objekt

Man kan ha objekt i arrayer.

<http://jsfiddle.net/2r1ph08o/>

```
let students = [];  
  
let person = {  
  name: "Micke",  
  age: 42,  
  shoe_size: 43  
};  
students.push(person);
```

```
person = {  
  name: "Jessica",  
  age: 25,  
  shoe_size: 36  
};  
students.push(person);  
  
console.log(students);
```

Grupppuppgift

- I grupper om 3-4 personer ska ni bygga en site där användare ska spela yatzy.
 - Det ska finnas ett Yatzy-formulär för upp till fyra spelare.
- Extrauppgifter:
 - Användare ska kunna logga in/registrera.
 - Användaren ska ha en profil-sida
 - Det ska finnas en lista med senaste matcherna (hitta på resultat t v).
 - Lägg upp ett gemensamt repo på GitHub.

Gruppuppgift

YATZY						
SPELARE:						Högsta poäng
Ettor						1 x 5 = 5p
Tvåor						2 x 5 = 10p
Treor						3 x 5 = 15p
Fyror						4 x 5 = 20p
Femmor						5 x 5 = 25p
Sexor						6 x 5 = 30p
SUMMA:						105poäng är max
BONUS (50)						Över 63p =50 bonus
Par						6+6 = 12p
Två Par						6+6+5+5 = 22p
Triss						6+6+6 = 18p
Fyrtal						6+6+6+6 = 24p
Kåk						6+6+6+5+5 = 28p
Liten stege						1,2,3,4,5 = 15p
Stor stege						2,3,4,5,6 = 20P
Chans						6+6+6+6+6 = 30p
Yatzy (50)						x+x+x+x+x = 50p
SUMMA:						374p är max
www.spelayatzy.se						

Yatzy

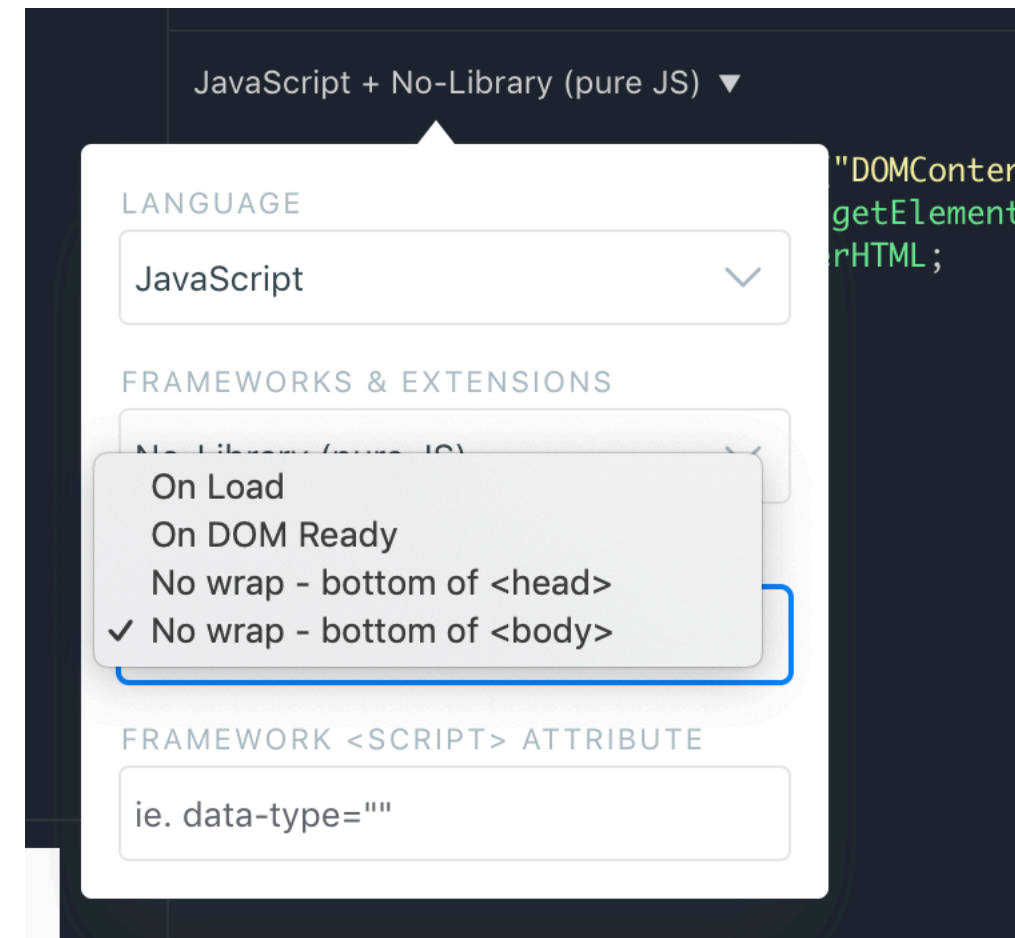
- Diskutera i era grupper hur en lämplig datastruktur för spelare och resultat skulle kunna se ut.
- Återvänd och diskutera i klassen.
- Implementera.

DOM

- Document Object Model - detta är JavaScriptets objektrepresentation av html-koden.
- Plattforms- och språkoberoende gränssnitt som ger programspråk möjligheten att dynamiskt läsa och uppdatera ett dokument innehåll, struktur och formatering.

jsfiddle.net

- JsFiddle ger möjlighet att bestämma när scriptet ska köras.
- När dokumentet har laddats.
- När DOM:en är klar.
- Lägg skriptet i `<head>`.
- Lägg skriptet precis innan `</body>`.



Interaktion med HTML/CSS

Om du har valt att inte wrappa skriptet kan du själv behöva se till att elementen är skapade.

```
document.addEventListener("DOMContentLoaded", function(event) {  
    // Do stuff  
});
```

Event

- Vad är det för typ av event?

```
<button id="calc">Beräkna</button>
```

```
// - - -
```

```
// Spara beräkna-knappen i en variabel.  
let calc_button = document.getElementById("calc");
```

```
// Ha koll på om någon klickar på beräkna-knappen.  
calc_button.addEventListener("click", function(event) {  
  // Do something.  
});
```

```
// - - -
```

Testa

- Lägg till en knapp som visar en meddelanderuta när ni klickar på knappen.
- Använd den inbyggda funktionen `alert("Meddelande");`
- *Extra uppgift:* Lägg in en input av typ `color`. När användaren väljer en färg ska ni uppdatera bakgrundsfärgen på ert dokument (eller något annat) med den färgen med hjälp av JS.

Yatzy

- Ge era inputs id:n, typ "player1_ones".
- När man klickar på Beräkna-knappen ska summan av alla ettor, tvåor, treor, fyror, femmor och sexor beräknas och summan ska uppdateras. Det räcker att göra det för en spelare.
- Skriv ut värde på olika sätt, t ex `innerHTML` om ni har ett element typ `p`, `span`, `td` eller liknande. `value` om ni har en `input`.
- Om spelarens summa är minst 63 poäng ska användaren få 50 poäng i bonus.
- *Extrauppgift:* Gör en funktion som tar en array med fem tal som parameter och returnerar sant om talen innehåller en kåk. (3 av samma + 2 av samma)

Funktioner

- Vilken typ är parametrarna?

Sträng

Funktion

```
document.addEventListener ("DOMContentLoaded", function(event) {  
    let headline = document.getElementById("headline");  
  
    headline.innerHTML = "Lorem ipsum";  
});
```

Funktioner

- Det måste betyda att vi kan ange en "extern" funktion som parameter!

```
function something(event) {  
  let headline = document.getElementById("headline");  
  
  headline.innerHTML = "Lorem ipsum";  
}  
  
document.addEventListener ("DOMContentLoaded", something);
```

Inbyggda funktioner

- JS har en massa inbyggda funktioner, t ex ett matte-bibliotek med matte-funktioner.
- Här är ett exempel på hur man kan slumpa fram ett tal.
- Returnerar ett tal större än eller lika med 0, mindre än 1.

```
let slump = Math.random();  
console.log(slump);
```

0.7126295249855692

0.5267871618023883

0.3054544574128275



Math.random

- Räkna fram ett heltal i ett visst intervall.

```
function getRandomInt(min, max) {  
  // The maximum is exclusive and the minimum is inclusive  
  return Math.floor(Math.random() * (max - min)) + min;  
}
```

Funktioner

- *Uppgift:* Gör ett formulär med ett text-fält och tre knappar.
- När man klickar på någon av knapparna ska en funktion anropas som slumpar fram ett tal och uppdaterar värdet i textrutan med det.
- Det ska vara samma funktion som anropas vilken knapp man än trycker på.



Button 1 Button 2 Button 3

```
let slump = Math.random();  
console.log(slump);
```

Sammanfattning

- Datatyper
- Jämförelseoperatorer
- Funktioner
- DOM
- Array-funktioner
- jQuery
- JSON
- API
- Node.JS
- npm
- webpack
- Continuous Integration

Utvärdering

- Prata i grupper om 2-3 personer i två minuter.
- Vad har varit bra idag?
- Vad skulle kunna förbättras?

Tack för idag!

Mikael Olsson
mikael.olsson@emmio.se
076-174 90 43

