
FUNCTIONS AND OBJECTS

AGENDA

- ▶ Bygga ett program
- ▶ Fler array metoder
- ▶ Strängar och properties
- ▶ Restparametrar, spread och destructuring
- ▶ Math objekt
- ▶ JSON

BYGGA ETT PROGRAM

- ▶ En blog (tillsammans)

```
let blog = [];  
  
function addBlogEntry(tags, entry) {  
  blog.push({tags, entry});  
}
```

- ▶ Lägg till en blogEntry, printa ut blog efter
- ▶ Uppgift
 - ▶ Addera 7 blog entries (för en vecka)
 - ▶ Printa ut blog

BYGGA ETT PROGRAM

- ▶ En blog (tillsammans)
- ▶ Nu vill vi publicera en veckas händelser i en tidning, behöver veta hur många tecken som finns

```
function countWeekChars(blog) {  
    return blog[0].entry.length + // mån  
           blog[1].entry.length + // tis  
           blog[2].entry.length + // ons  
           blog[3].entry.length + // tor  
           blog[4].entry.length + // fre  
           blog[5].entry.length + // lör  
           blog[6].entry.length;  // sön  
}  
  
console.log(countWeekChars(blog));
```

- ▶ Uppgift
 - ▶ Printa ut din blogs antal tecken
 - ▶ Refaktorerera skapa "countEntryChars" och anropa från "countWeekChars"

BYGGA ETT PROGRAM

- ▶ En blog (tillsammans)
- ▶ Nu behöver vår kunna filtrera fram blogentries m.h.a olika taggar

```
function entriesFor(tag, blog) {  
  let entries = [];  
  for (let i = 0; i < blog.length; i++) {  
    let entry = blog[i];  
    if (entry.tags.includes(tag))  
      entries.push(entry);  
  }  
  return entries;  
}  
  
console.log(entriesFor("pizza", blog));
```

- ▶ Uppgift
 - ▶ Testa för en godtycklig tagg

BYGGA ETT PROGRAM

- ▶ En blog (tillsammans)
- ▶ Refaktorera

```
for (let i = 0; i < blog.length; i++) {  
  let entry = blog[i];  
  // Do something with entry  
}
```

```
for (let entry of blog) {  
  // Do something with entry  
}
```

BYGGA ETT PROGRAM

- ▶ En blog (tillsammans)
- ▶ Nu har vi
 - ▶ addBlogEntry
 - ▶ entriesFor (filter)
 - ▶ countWeekChars
 - ▶ countEntryChars
- ▶ Uppgift: Vi vill skapa en funktion "countCharsFor" som räknar alla tecken för en tagg
 - ▶ Vilka funktioner kan vi ha nytta av då?

FLER ARRAY METODER

- ▶ Säg att vi vill skapa en todolist applikation

```
let todoList = [];  
function remember(task) {  
  todoList.push(task);  
}  
function getTask() {  
  return todoList.shift();  
}  
function rememberUrgently(task) {  
  todoList.unshift(task);  
}
```

- ▶ **shift**: plockar ut något från slutet
- ▶ **unshift**: pressar in något i början

FLER ARRAY METODER

► Uppgift

- Skapa en variabel "todoList" och (utan att använda remember, getTask eller rememberUrgently)
- Lägg till 3 todos
- Tryck in 1 ny todo i början
- Ta bort och skriv ut den sista todon

```
let todoList = [];  
function remember(task) {  
  todoList.push(task);  
}  
function getTask() {  
  return todoList.shift();  
}  
function rememberUrgently(task) {  
  todoList.unshift(task);  
}
```

FLER ARRAY METODER

► Frågor

```
console.log([1, 2, 3, 2, 1].indexOf(2));  
// → 1  
console.log([1, 2, 3, 2, 1].lastIndexOf(2));  
// → 3
```

► Vad ger **indexOf(3)**?

► Vad gör **lastIndexOf(1)**?

FLER ARRAY METODER

► Frågor

```
console.log([0, 1, 2, 3, 4].slice(2, 4));  
// → [2, 3]  
console.log([0, 1, 2, 3, 4].slice(2));  
// → [2, 3, 4]
```

► Vad gör **slice(4)**?

► Vad gör **slice(1, 3)**?

FLER ARRAY METODER

► Uppgift

- Vi vill skriva en funktion "remove" som tar 2 argument (array och index)
- Tar bort elementet angivet av index

```
function remove(array, index) {  
  return array.slice(0, index)  
    .concat(array.slice(index + 1));  
}  
console.log(remove(["a", "b", "c", "d", "e"], 2));
```

- Finns några problem med denna funktion

STRÄNGAR OCH PROPERTIES

- ▶ Vi har properties som **length** och **toUpperCase** på strängar
- ▶ Men om vi försöker lägga till en property (behandlar sträng som objekt)

```
let kim = "Kim";  
kim.age = 88;  
console.log(kim.age);  
// → undefined
```

- ▶ Varför?

STRÄNGAR OCH PROPERTIES

- ▶ Strängar är konstiga primitiva värden
 - ▶ Primitiva värden: Number, String, Boolean, Null, Undefined
- ▶ Strängar beter sig överlag som en array

```
console.log("coconuts".slice(4, 7));  
// → nut  
console.log("coconut".indexOf("u"));  
// → 5
```

```
console.log("one two three".indexOf("ee"));  
// → 11
```

- ▶ Men shift och unshift finns inte, varför?

STRÄNGAR OCH PROPERTIES

- Andra metoder på strängar än (**toUpperCase**, **toLowerCase**)

```
console.log("  okay \n ".trim());  
// → okay
```

```
console.log("h".padStart(3, "0"));  
// → 00h
```

```
let sentence = "Secretarybirds specialize in stomping";  
let words = sentence.split(" ");  
console.log(words);  
// → ["Secretarybirds", "specialize", "in", "stomping"]  
console.log(words.join(". "));  
// → Secretarybirds. specialize. in. stomping
```

```
console.log("LA".repeat(3));  
// → LALALA
```

STRÄNGAR OCH PROPERTIES

▶ Uppgift

- ▶ Skapa en variabel `const rawFullName` med ditt namn och efternamn med white space framför och efter
- ▶ Tilldela en ny variabel `cleanFullName` utan whitespace (`trim`)
- ▶ Printa ut ditt efternamn (`slice`)
- ▶ Skapa en ny array `names` av ditt förnamn och efternamn (`split`)
- ▶ Printa ut `names` i en sträng med `","` mellan (`join`)
- ▶ Printa ut ditt förnamn repeterat 5 gången (`repeat`)

STRÄNGAR OCH PROPERTIES

▶ Uppgift

- ▶ Skapa en funktion "printInitials" som tar en sträng med förnamn och efternamn
 - ▶ Ex: `console.log(printInitials("pelle svanslös")) => "PN"`
 - ▶ Ex: `console.log(printInitials(" Pelle Svanslös. ")) => "PN"`
- ▶ Använd de funktioner vi nyss tittat på
 - ▶ `split`, `join`, `trim`, `slice`, `toUpperCase`

RESTOPPARAMETRAR SPREAD OCH DESTRUCTURING

► Uppgift

► Restparametrar `func(...args)` och spreadoperator `(...array)`

```
function max(...numbers) {  
  let result = -Infinity;  
  for (let number of numbers) {  
    if (number > result) result = number;  
  }  
  return result;  
}  
console.log(max(4, 1, 9, -2));  
// → 9
```

```
let numbers = [5, 1, 7];  
console.log(max(...numbers));  
// → 7
```

```
let words = ["never", "fully"];  
console.log(["will", ...words, "understand"]);  
// → ["will", "never", "fully", "understand"]
```

► Gör en funktion "countCharacter" som tar flera strängar och räknar ut hur många tecken

► Ex: `console.log(countCharacter("hej", "då", "hur", "mår", "du?")) => 14`

RESTPARAMETRAR SPREAD OCH DESTRUCTURING

► Uppgift

► Destructuring

```
function phi(table) {  
  return (table[0] * table[1] - table[2] * table[3]);  
}
```

```
function phi([n0, n1, n2, n3]) {  
  return n0 * n1 * n2 * n3;  
}
```

```
let {name} = {name: "Faraji", age: 23};  
console.log(name);  
// → Faraji
```

► Uppgift: gör en funktion som tar en array av 4 siffror och adderar alla talen

RESTOPPARAMETRAR SPREAD OCH DESTRUCTURING

- ▶ Frågor

- ▶ Vad är rest parametrar?
- ▶ Vad är spread operatör?
- ▶ Vad är destructuring?

MATH OBJECT

- ▶ Math objektet är en hållare för en massa matterlaterade funktioner

- ▶ Math.max, Math.min

```
console.log(Math.random());
```

```
// → 0.36993729369714856
```

- ▶ Math.random

```
console.log(Math.random());
```

```
// → 0.727367032552138
```

- ▶ Vissa konstanter också

```
console.log(Math.random());
```

```
// → 0.40180766698904335
```

- ▶ Math.PI

```
function circleCircumference(radius) {  
    return 2 * radius * Math.PI;  
}
```

MATH OBJECT

► Uppgift

- Gör en funktion som ger ett godtyckligt tal mellan 1-100

```
console.log(Math.random());  
// → 0.36993729369714856  
console.log(Math.random());  
// → 0.727367032552138  
console.log(Math.random());  
// → 0.40180766698904335
```

JSON

- ▶ Ibland vill vi skicka data
 - ▶ Till andra applikationer (på samma dator)
 - ▶ Till hårddisk (för att spara)
 - ▶ Till en annan dator (över internet)
- ▶ För att skicka data behöver vi "serialisera" data
 - ▶ Konvertera datan till 1:or och 0:or

JSON

- ▶ Frågor och diskussion
 - ▶ Hur ska vi skicka data om en användare till bank t.ex för registrering
 - ▶ **const user = { firstName: "Pelle", lastName: "Svanslös", age: 15 }**
 - ▶ Vi kan skicka text, alla datorer förstår ju Unicode (Internet bygger på det)
 - ▶ Så vi antar att texten mottas på andra sidan utan problem
 - ▶ Men hur ska mottagaren tolka texten? Diskutera...
 - ▶ Exempel i XML: "<user><firstName>Pelle</firstName>...</user>"

JSON

- ▶ Ett populärt serialiseringsformat är JSON (JavaScript Object Notation)
- ▶ Så populärt och används av många programmeringsspråk
- ▶ Men grundar sig på JavaScripts notation för object och arrayer

```
{  
  "user": {  
    "firstName": "Pelle",  
    "lastName": "Svanslös",  
    "age": 15,  
    "confirmed": true,  
    "accountNrs": ["123-21234-123", "124-29276-998"]  
  }  
}
```

JSON

- ▶ Det finns ett annat inbyggt objekt JSON som har funktioner för JSON hantering

```
let string = JSON.stringify({squirrel: false,  
                             events: ["weekend"]});  
  
console.log(string);  
// → {"squirrel":false,"events":["weekend"]}  
console.log(JSON.parse(string).events);  
// → ["weekend"]
```

JSON

▶ Uppgift

▶ Skapa följande objekt

```
{  
  "user": {  
    "firstName": "Pelle",  
    "lastName": "Svanlös",  
    "age": 15,  
    "confirmed": true,  
    "accountNrs": ["123-21234-123", "124-29276-998"]  
  }  
}
```

▶ Spara objektet serialiserat som JSON-sträng (JSON.stringify)

▶ Printa ut strängen

▶ Konvertera den serialiserade strängen tillbaka till JavaScript objekt (JSON.parse)

JSON

▶ Uppgift

- ▶ Skapa en funktion "append" som tar en JSON-sträng, en key och en value

- ▶ Ex: `append("{ \"user\": { \"name\": \"Pelle\" } }", "age", 15)`

- ▶ `=> "{ \"user\": { \"name\": \"Pelle\" , \"age\": 15 } }"`

- ▶ Skapa en funktion "remove" som tar en JSON-sträng, en key och tar bort en key och value om de finns

- ▶ Ex: `remove('{ "user": { "name": "Pelle" , "age": 15 } }', "age")`

- ▶ `=> '{ "user": { "name": "Pelle" } }'`

JSON

▶ Frågor

- ▶ Vad är serialisering?
- ▶ Vad är det bra för?
- ▶ Vad är deserialisering?
- ▶ Hur är det möjligt att andra programmeringsspråk kan använda JSON?

SUMMERING (VAD HAR VI LÄRT OSS?)

- ▶ Vad gör: [].push, [].pop, [].shift, [].unshift, [].indexOf, [].lastIndexOf, [].slice
- ▶ Vad gör: "".split, "".trim, "".join, "".repeat
- ▶ Vad är rest parametrar, spread operatorn, destructuring?
- ▶ Vad gör Math.random?
- ▶ Vad används JSON till?

ÖVNINGAR

1. There are elections in your country! In a small town, there are only 50 voters. Use a for loop to simulate the 50 people voting, by logging a string like this to the console (for numbers 1 to 50): *'Voter number 1 is currently voting'*

ÖVNINGAR

1. Let's bring back the `'populations'` array from a previous assignment
2. Use a for loop to compute an array called `'percentages2'` containing the percentages of the world population for the 4 population values. Use the function `'percentageOfWorld1'` that you created earlier
3. Confirm that `'percentages2'` contains exactly the same values as the `'percentages'` array that we created manually in the previous assignment, and reflect on how much better this solution is

ÖVNINGAR

- ▶ Eloquent JavaScript
 - ▶ https://eloquentjavascript.net/04_data.html
 - ▶ Övningar: The sum of a range, Reversing an array, A list, Deep Comparison

LÄXA

- ▶ Eloquent JavaScript

- ▶ https://eloquentjavascript.net/05_higher_order.html