

JavaScript: Functions & Parameters



Functions (2. 15)

Det grundlæggende som du kender:

Erklære en funktion (lave en funktion):

```
function myFunc() {  
    // Code goes here  
}
```

Kalde en funktion (bruge en funktion):

```
myFunc( );
```

En funktionsreference (som i event listenere):

```
myFunc
```

[2.15]: <https://javascript.info/function-basics>

Funktions-parametre

Funktions - par a m e t r e

Et parameter er ligesom en variabel man sender til en funktion.

Parametre kan hedde hvad det skal være *ligesom variabler*

Når man kalder en funktion, behøver man ikke at vide hvad parametrene hedder, men man skal *kende rækkefølgen af parametrene*

```
function myFunc(parm1, parm2) {  
    // code uses parm1 and parm2  
}
```

Hvordan man bruger parametre #1

Du har prøvet at bruge parametre før

Dette er et string-parameter til alert

```
alert("Do not use alerts for user info!");
```

Det gør ingen forskel for funktionen om den modtager parameteret direkte eller som en variabel:

Dette er et string-parameter til console.log

```
console.log("JavaScript is running");
```

Dette er også et string-parameter til console.log

```
const message = "JavaScript is running";  
console.log(message);
```

Hvordan man bruger parametre #2

Vi har også brugt flere parametre før

Her, har vi to parametre til console.log

```
console.log("The value of my myVar", myVar);
```

Using Parameters #3

Vi har endda prøvet at bruge funktioner som parametre:

```
window.addEventListener("DOMContentLoaded", start);
```

1. Det første parameter er en string, og
2. Det andet parameter er en functions-reference og *ikke et kald til en function!*

Function Øvelser

Function Øvelse 1a

Så lad os starte med Øvelsen: sayHello

1. Lav en function, der hedder sayHello
2. og som modtager et parameter der hedder firstName, og
3. skriver `Hello \${firstName}` til konsollen

Function Øvelse 1b

Test af sayHello funktionen

Hvad sker der hvis:

1. Hvis funktionen bliver kaldt på en linje tidligere end den er erklæret på
2. Hvis funktionen bliver kaldt med en variabel der hedder myName?
3. Hvis der er en anden variabel der hedder firstName uden for funktionen
4. Hvis funktionen bliver kaldt uden noget parameter?
5. Hvis funktionen bliver kaldt med firstName og lastName
6. Og hvis funktionen bliver kaldt med lastName før firstName

Function Øvelse 2a

Flere parametre: Øvelesen presentPet

1. Lav en ny function der hedder presentPet
2. der modtager tre parametre:
 - a. firstName,
 - b. animalType,
 - c. animalName
3. Functionen skal logge `My name is \${firstName}, I have a \${animalType} called \${animalName}` til konsollen

Function Exercise 2b

Test funktionen: presentPet

Hvad sker der hvis:

Functionen bliver kaldt med variabler der hedder det samme som parametrene

Functionen bliver kaldt med variabler der hedder noget andet?

Functionen bliver kaldt med strings?

Functionen bliver kaldt med en blanding af strings og variabler?

Function bliver kaldt med variabler, men i en anden rækkefølge?

(Din) Konklusion

Nogle konklusioner om parametre

Skriv dine egne konklusioner om disse emner:

Parametre som variabler eller strings, eller andre faste værdier

Hvordan opfører de sig, hvordan bliver de brugt?

Navne på variabler der bliver brugt når man kalder en function?

Rækkefølgen af parametre når man kalder en function?

Antallet af parametre når man kalder en function?

Parametre der påvirker andre variabler udenfor functionen?

Flere eksperimenter...

Nogle interessante eksperimenter

Prøv dette:

Kan man erklære en variabel i en function med det samme navn som et parameter?

Hvilken af dem bliver brugt?

Kan man have et parameter med det samme navn som en global variabel?

Men alligevel få en anden værdi?

Kan man ændre værdien af en variabel inde i functionen som firstName = "Tom"?

Kan man ændre værdien af en global variabel inde fra en funktion?

default parameter

Vi kan sætte et function-parameter til en værdi, og så kan vi kalde functionen uden parameteret og parameteret vil så have den værdi vi satte det til *default*.

Hvis vi så alligevel vælger at kalde funtionen med parameteret, bliver default parameteret overskrevet og det parameter vi sender med bliver anvendt i stedet



```
script.js — default_param
JS script.js x # style.css <> index.html
JS script.js > ...
1  "use strict";
2
3  setColor();
4
5  function setColor(color = "pink") {
6    document.querySelector("div").style.backgroundColor = color;
7  }
```



```
script.js — default_param
JS script.js x # style.css <> index.html x
JS script.js > ...
1  "use strict";
2
3  setColor("lightblue");
4
5  function setColor(color = "pink") {
6    document.querySelector("div").style.backgroundColor = color;
7  }
```

~/Documents/KEA_dev/dev_efteraar_2023/the_new_jj_dev/html_css_js/default_param/index.html