

Javascript Grundlagen

Modul 133

Ziele dieser Lektion

- Sie wissen wie mit Javascript **Funktionen** erstellt werden und können selbst einfache Funktionen programmieren.
- Sie können die Begriffe **Argument** und **Parameter** mit eigene Worten beschreiben.
- Sie kennen den Unterschied zwischen **Function Declaration**, **Function Expression** und **IIFE**
- Sie kennen die **Integrierten Objekte** von Javascript und können Ihre Anwendung und Funktion unterscheiden.

Aufbau einer Funktion

```
function name(parameter) {  
    return xy;  
}  
var z = name(argumente);
```

function: Schlüsselwort zu erstellen einer Funktion.

name: Name der Funktion, Case Sensitive (CamelCase), mit diesem Namen kann die Funktion überall im Code aufgerufen werden.

parameter: stehen innerhalb der Funktion als lokale Variablen zur Verfügung.

return: Rückgabewerte der Funktion.

argumente: Werte, welcher der Funktion beim Aufruf übergeben werden können.

Beispiel einer einfachen Funktionen

```
function sayHello() {  
    document.write('Hello ');  
}  
sayHello();
```

Die Funktion kann mit dem Funktionsaufruf `sayHello();` immer wieder ausgeführt werden.

Funktion mit Parametern und Rückgabewerten

```
function calculateArea(width, height = 1) {  
    var area = width * height;  
    return area;  
};  
  
var wallOne = calculateArea(3,5);  
var wallTwo = calculateArea(5);
```

Lokale und Globale Variablen

```
var a = 20;  
var b = meineBerechnung(a);  
alert("a: "+a);  
alert("b: "+b);  
alert("x out: "+x);
```

```
function meineBerechnung(x) {  
    x+=10;  
    alert("x in: "+x);  
    return x;  
}
```

Funktionen mit mehreren Rückgabewerten

```
function getSize(width, height, depth)
{
    var area = width * height;
    var volume = width * height * depth;
    var sizes = [area, volume];
    return sizes;
};

var areaOne = getSize(3, 2, 3) [0];
var volumeOne = getSize(3, 2, 3) [1];
```

Arten von Funktionen:

Function Declaration:

Benannte Funktionen werden **vom Interpreter immer zuerst eingelesen**. Eine deklarierte Funktion deshalb auch **vor der Deklaration aufgerufen werden**.

```
var roomSize = area(7, 8);  
  
function area(width, height) {  
    return width * height;  
};
```


Arten von Funktionen

Function Expression:

Anonyme Funktionen, eine Funktion ohne Namen speichert in der Variable `area` die **Funktion**, diese kann deshalb **nur nach der Deklaration** aufgerufen werden

```
var area = function(width, height) {  
    return width * height;  
};  
var roomsize = area(7, 8);
```

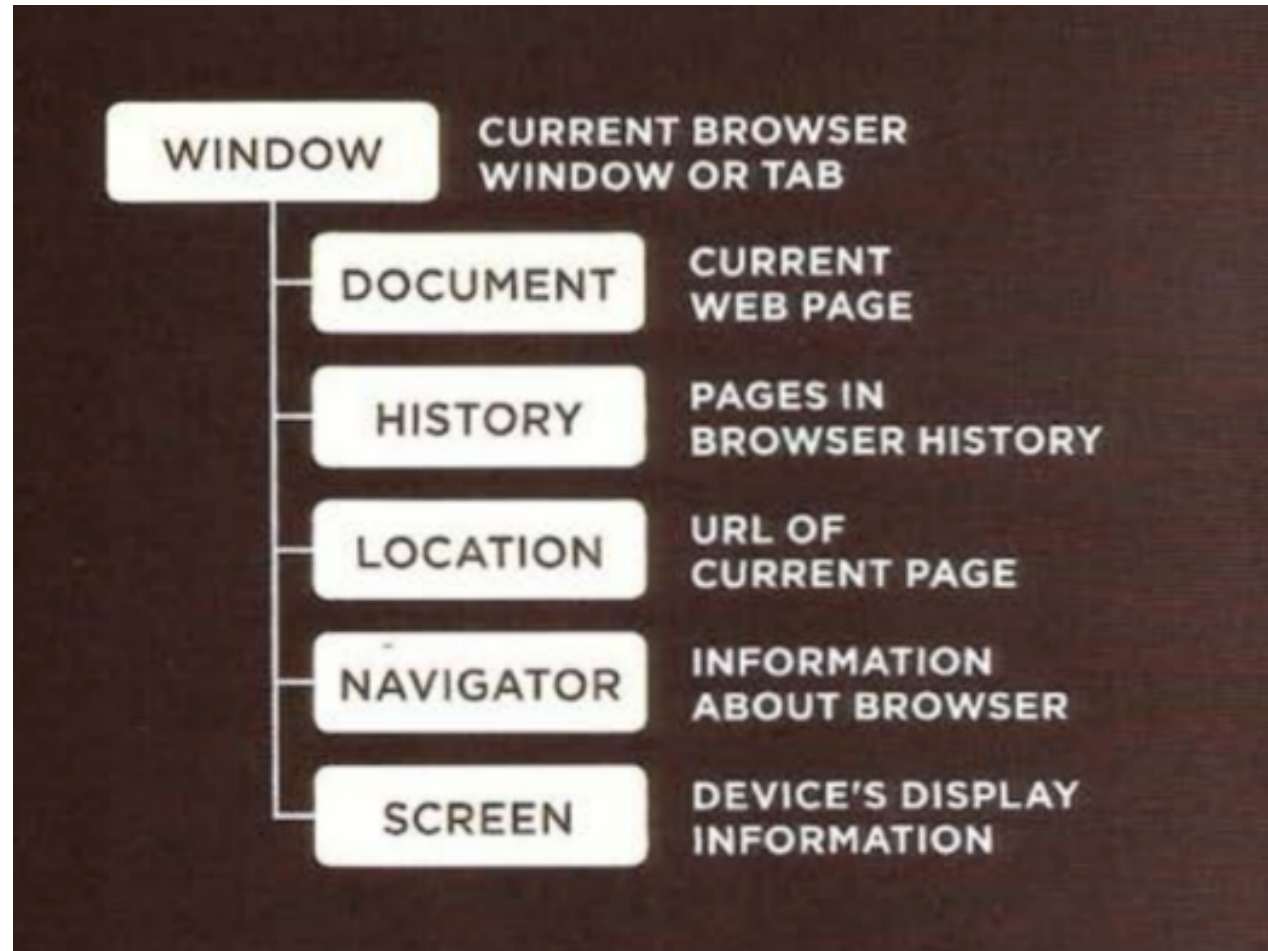
Arten von Funktionen

IIFE: Immediately-invoked Function Expression

Ist ein sofort ausgeführter Funktionsausdruck. Die Variable `area` speichert nicht die Funktion, sondern nur das Resultat der anonymen Funktion. Wird verwendet, um Konflikte zwischen unterschiedlichen Scripts zu verhindern (lokale und globale Variablen).

```
var area = (function() {  
    var width = 3;  
    var height = 2;  
    return width * height;  
}()) ;
```

Integrierte Objekte



Integrierte Objekte

- Werden vom Browser zur Verfügung gestellt.
- Bieten Werkzeuge für den Zugriff auf
 - die geladene Seite (Document / Document Object Model)
 - den Browserverlauf (History)
 - die aktuell geladene URL (Location)
 - Informationen über den verwendeten Browser (Navigator)
 - Informationen über das Display des anzeigenden Gerätes (Screen)

`window.document.URL` = aktuelle URL des Tabs

`window.screen.width` = Breite des Tabs

`window.history.length` = Anzahl Objekte in der Browser-History

Document Object Model (DOM)

- Das Document Object Model legt ein Modell der Webseite an.
- Das oberste Objekt ist `document` und steht für die Seite als Ganzes.
- Seine Kindobjekte stellen einzelne Elemente der Seite dar.

`document.getElementById(, ID `)` = Selektioniert ein Element anhand seiner ID.

`document.getElementsByTagName(, li `)` = Selektioniert alle li-Elemente in der Seite.

Globale Javascript Objekte

Globale Javascript-Objekte liefern Methoden um mit den Javascript-Datentypen zu arbeiten.

String:

```
var saying = "Hello World";
```

```
saying.length; > 11
```

```
saying.toUpperCase(); >  
HELLO WORLD
```

Number:

```
var zahl = 10.123456
```

```
zahl.toFixed(2); > 10.12
```

Globale Javascript Objekte

Weiter gibt es das `Math` und das `Date` Objekt.

Math:

`Math.random()` ; Zahl
zwischen 0 und 1

`Math.round(zahl)` ; Rundet
die Zahl auf die nächste
Ganzzahl

`Math.ceil(zahl)` ; Aufrunden

`Math.floor(zahl)` ; Abrunden

Date:

`var today = new Date()` ;
`today.getDate()` ; Tag des
Monats

`today.getTime()` ; > Unix
Timestamp

Weitere Informationen

Buch Javascript&jQuery:

Kapitel 3: Funktionen, Methoden & Objekte