



Einführung in die Programmierung

- Google Apps Script & JavaScript -

Claudia Scheffler

16. September 2013



1 Objekte

- Was ist ein Objekt?
- Objekthinhalte
- Eigenschaften von Objekten
- Funktionen von Objekten
- Erzeugung von Objekten
- Zugriff auf Objekte
- Eigene Objekte
- Objekt-Referenz
- Objekthierarchie
- Objekte vs. Klassen



Fragen ?



Objekt

Ein Objekt ist eine zusammengesetzte Datenstruktur mit Eigenschaften zur Speicherung von Informationen über das Objekt und Funktionen, die bereitgestellt werden, um Objektwerte zu erhalten, zu setzen oder komplexere Operationen auszuführen.



- fassen Daten und Funktionen so zusammen, dass sie als etwas in sich geschlossenes, funktionell Ganzes erscheinen
- leicht in verschiedenen Situationen einzusetzen, ohne sich mit den Details ihrer genauen Funktionsweise zu beschäftigen



Fragen ?



Objekthinhalt

Bei den Inhalten eines Objektes, handelt es sich um eine Referenz (ein Verweis) auf das Objekt selbst.

- es können alle möglichen Daten zusammengefasst werden
- **Beispiel:**
 - ganze Zahlen
 - Arrays
 - Objekte



Fragen ?



Objekteigenschaften

Bei den Eigenschaften eines Objektes handelt es sich um Variablen von Objekten. Diese sind festgelegt, da sie einen vorgegebene Wert liefern.

- können aus einen beliebigen Datentyp, Array oder Objekt bestehen
- **Inhalte von Eigenschaften:**
 - konkrete Werte
 - Eigenschaften können mit neuen Werten versehen werden



Fragen ?



Objektfunktionen

Bei Funktionen von Objekten handelt es sich, um Funktionen, die von den Objekten bereitgestellt werden.

■ zwei Arten:

- Boolesche Funktionen → liefert true oder false zurück
- Referenz Funktionen → liefert einen konkreten Wert zurück



Fragen ?



Ein Auto ist ein Objekt mit

- Eigenschaften:

- Farbe
- Anschaffungspreis
- Baujahr
- aktuelles Gewicht
-

- Methoden:

- starten
- beschleunigen
- bremsen
- schalten
-



Fragen ?



- eine Variable, über die das Objekt zugegriffen werden kann
- zwei Arte:
 - vom Interpreter automatisch unter einem vorgegebene Namen erzeugt
 - die nach Bedarf eingesetzt werden
- zu Erzeugung muss der Name des Objektes bekannt sein



window	das aktuelle Browserfenster
document	das aktuelle HTML-Dokument
navigator	der gerade benutzte Browser
location	die Adresse des aktuellen Dokuments
history	die History-Liste des Browsers
screen	der Bildschirm des Benutzers

- Objekt-Variablen sind automatisch vorhanden, sobald eine Seite im Browser betrachtet wird → damit kann auf das Objekt zugegriffen werden



Konstruktor

Unter einem Konstruktor versteht man eine spezielle Funktion, die die nötigen Anweisungen enthält, um ein neues Objekt gemäß einer Vorlage zu konstruieren.

- Aufruf erfolgt mit dem Schlüsselwort **new** und liefert ein neues Objekt zurück
- bei manchen Konstruktoren können auch Argumente übergeben werden
- verschiedene Arten von Objekten:
 - Array-Objekte



Instanz

Unter einer Instanz versteht man, wenn ein konkretes Objekt einer Familie gleichartiger Objekte, durch einen Konstruktoraufruf erzeugt wird.

Beispiel

```
var a = new Array();
```



- um auf Objekte zuzugreifen, existieren zwei Arten:
 - Zugriff auf einzigartige Elemente, die immer vorhanden sind
 - Zugriff über einen Index



- zuerst wird auf das Hauptobjekt zugegriffen → dies geschieht über den Namen des Objektes

Beispiel

- document
- window

Beispiel

```
<html>
<head>
<title>Hallo Welt</title>
<script type="text/javascript">
var titel = document.title;
alert(titel);
</script>
</head>
<body>
</body>
</html>
```



- bei dieser Variante wird das Unterobjekt, die Eigenschaft oder auch die Methode, durch einen Punkt getrennt, angesprochen

Syntax

- `hauptobjekt.unterobjekt/methode/eigenschaft`
- `hauptobjekt.unterobjekt.unterunterobjekt.methode`

Beispiel

```
<html>
<head>
<title>Hallo Welt</title>
<script type="text/javascript">
var titel = document.title;
alert(titel);
</script>
</head>
<body>
</body>
</html>
```



- verfügt ein Objekt über Unterobjekte, so kann man sich dessen Speicherung als eine Art Array vorstellen
- Zugriff über:
 - numerischen Index

Beispiel

```
hauptobjekt.unterobjekte[0];
```

- alphanumerischen Index

Beispiel

```
hauptobjekt['objektname']  
hauptobjekt.unterobjekte['objektname'];
```



- ist es notwendig auf ein Unterobjekt mittels einer Variable zuzugreifen, so muss selbige in eckigen Klammern angegeben werden

Beispiel

```
var blubb = 'irgendwas';  
hauptobjekt.unterobjekt[blubb]
```

Achtung!

- Wird die Variable nicht in eckigen Klammern angegeben, so produziert man eine Fehlermeldung
- **Beispiel:** hauptobjekt.unterobjekt.blubb → **Fehler!**



Fragen ?



- es existiert ein primitives Objekt als Vorlage für eigene Objekte
- von diesem Objekt sind alle anderen Objekte abgeleitet

Neues Objekt

Ein neues Objekt wird durch den Operator **new** und dem `Object()`-Konstruktor erzeugt.



- Ein Objekt wird in einem zweistufigen Prozess erstellt:
 - **1.Schritt:** Deklaration eines Objekts → anlegen einer Vorlage für das Objekt
 - **2.Schritt:** Instanziierung des Objektes → erzeugt die Datenstruktur durch den Operator `new`

Beispiel

- `var spiel = new Object;`
- `spiel.spieler = "Hans";`
- `spiel.punkte = 2;`



- ein Objekt "formular" könnte z.B. die Eingabefelder eines Formulars enthalten

Beispiel

```
var formular = new Object;  
formular.firstName = document.getElementById('firstName');  
formular.lastName = document.getElementById("lastName");  
formular.plz = document.getElementById("plz");  
formular.ort = document.getElementById("ort");
```

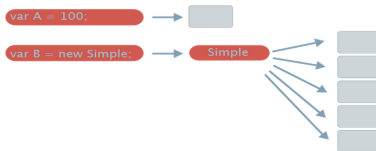
- Erzeugung eines eigenen Objektes als Instanz des primitiven Objekts gruppiert logisch zusammengehörige Variablen und Funktionen
- Zugriff auf die Elemente des Objekts mit dem Namen gestaltet das Script lesbarer



Fragen ?



- eine Variable verweist auf einen Speicherplatz, die sich zu Laufzeit nicht ändert
- Objekte hingegen speichern keine Daten wie Variablen, sonder verweisen nur auf die Werte



- Referenztypen wie Objekte verweisen nicht direkt auf den Speicherplatz, sondern auf eine Sammlung von Referenzen - erst diese Referenz verweist auf einen bestimmten Speicher



Fragen ?



- viele Objekte in sind in anderen Objekten enthalten
- sie stehen in einem "ist enthalten in" $\leftrightarrow$ "ist Behälter von"-Verhältnis zueinander und haben kein Klassen $\leftrightarrow$ Unterklassen-Verhältnis
- es existiert keine Vererbung



Fragen ?



- JavaScript kennt keine Klassendefinition und klassenorientierte Vererbung, dafür aber eine **Prototyp-basierte Vererbung**
- JavaScript Objekte haben keine Klasse-Unterklasse-Verhältnisse, sondern sind ineinander enthalten.
- wichtigsten Methode, mit denen Objekt-Systeme aufgebaut werden
 - die Vererbung (is-a)
 - die Aggregation (has-a)



- eine Prototyp-basierte Sprache, die vorgibt, eine Klassen-basierte Sprache zu sein
 - ein Objekt ist eine gekapselte mit Dateneigenschaften und Methoden
 - Dateneigenschaften sind Variablen oder andere Objekte, Methoden bearbeiten die Eigenschaften
 - eigene Objekte sind möglich
 - Objekte in der OOP sind Instanzen bestimmter Klassen
 - es existiert hier kein ausgereiftes Klassenkonzept mit Vererbung, nur ein Objektkonzept!
- hat Objekte, die Daten und Methoden und andere Objekte enthalten können
- keine Klassen, hat aber Konstruktoren, die dasselbe können wie Klassen und als Container für Eigenschaften und Methoden fungieren



Fragen ?

Claudia Scheffler
siclsche@stud.hszg.de