

Übungsblatt 04 Webtech

Exercise 1 – Scope, Closures, Currying

a) What does the JavaScript Window Object represent and how can it be used?

<https://developer.mozilla.org/de/docs/Web/API/Window>

https://www.w3schools.com/js/js_window.asp

- Das Window-Objekt repräsentiert ein offenes Fenster in einem Browser.
- Jeder Tab hat sein eigenes Window-Objekt
- Falls ein Dokument einen Frame (<iframe>) enthält, wird dafür ein eigenes Window-Objekt erzeugt.
- Globale Variablen sind Attribute des Window-Objekts
- Globale Funktionen sind Methoden des Window-Objekts
- Das Dokument-Objekt (vom HTML DOM) ist ein Attribut des Window-Objekts
- Weitere Attribute:
 - window.innerHeight //Innere Höhe des Browserfensters in Pixel
 - window.innerWidth //Innere Breite des Browserfensters in Pixel
- Weitere Methoden:
 - window.open() //Öffnet neues Fenster
 - window.close() //Schließt aktuelles Fenster
 - window.moveTo() //Bewegt aktuelles Fenster
 - window.resizeTo() //Neue Größe des aktuellen Fensters

b) What is the notion of scope in JavaScript?

- Man unterscheidet
 - Lokalen Scope
 - Globalen Scope
- Da globale Variablen Attribute des Window-Objekts sind, können diese von allen Skripten in diesem Fenster verändert werden (Globaler Scope), außer bei Closures
- Globale Variablen können lokal gemacht werden mit Closures
- Jede Funktion bildet einen neuen Scope
- Variablen in einer Funktion sind nicht von außen sichtbar

c) What is the difference between declaring variables with var, let, const or no keyword at all? What happens when 'use strict' is invoked for a script?

- **var**
Ist im aktuellen Kontext gültig, entweder der Kontext der Funktion welche die Variable enthält oder der globale Kontext bei globalen Variablen.

- **Let**
deklariert eine Variable im Gültigkeitsbereich des lokalen Blocks.

<pre> 1 function varTest() { 2 var x = 31; 3 if (true) { 4 var x = 71; // gleiche Variable! 5 console.log(x); // 71 6 } 7 console.log(x); // 71 8 }</pre>	<pre> 10 function letTest() { 11 let x = 31; 12 if (true) { 13 let x = 71; // andere variable 14 console.log(x); // 71 15 } 16 console.log(x); // 31 17 }</pre>
---	---

- **Const**
Die const-Deklaration erstellt eine Konstante. Gleich wie bei der let-Deklaration ist der

Gültigkeitsbereich auf den lokalen Block begrenzt. Der Wert einer Konstanten kann nicht verändert werden durch Zuweisung oder Neudeklaration.

- **Ohne Keyword**

Undeklarierte Variablen sind immer globale Variablen

- **'use strict'**

- Indiziert dass der Code im "Strict Mode" ausgeführt werden soll
- Alle aktuellen Browser unterstützen "Strict Mode"
- Im Strict Mode dürfen **keine undeklarierten Variablen** verwendet werden → error
- Fördert "secure"-JavaScript, also schlechte Syntax wird Fehler gewandelt
- Durch Typos können ungewollt globale Variablen entstehen, im Script Mode nicht
- Gleiche Parameter Namen für eine Funktion sind nicht erlaubt
`function x(p1, p1) {};`
- Löschen einer Variable / Funktion sind nicht erlaubt
`var x = 3.14;`
`delete x;`
- Strings wie "argument" oder "eval" sind nicht erlaubt als Variablenname
`var arguments = 3.14;`