

WEBTECHNOLOGIEN

02 – JAVASCRIPT –

EINFÜHRUNG, VARIABLEN UND KONTROLLSTRUKTUREN

PROF. DR. MARKUS HECKNER

WARUM JAVASCRIPT?

- Cross-platform
- Objektorientiert
- Hohe Nutzung und hohe Verbreitung Nutzung (99% der Clients haben JavaScript aktiviert, 90% aller Websites verwenden JavaScript)



WAS IST JAVASCRIPT?

Heute

Client

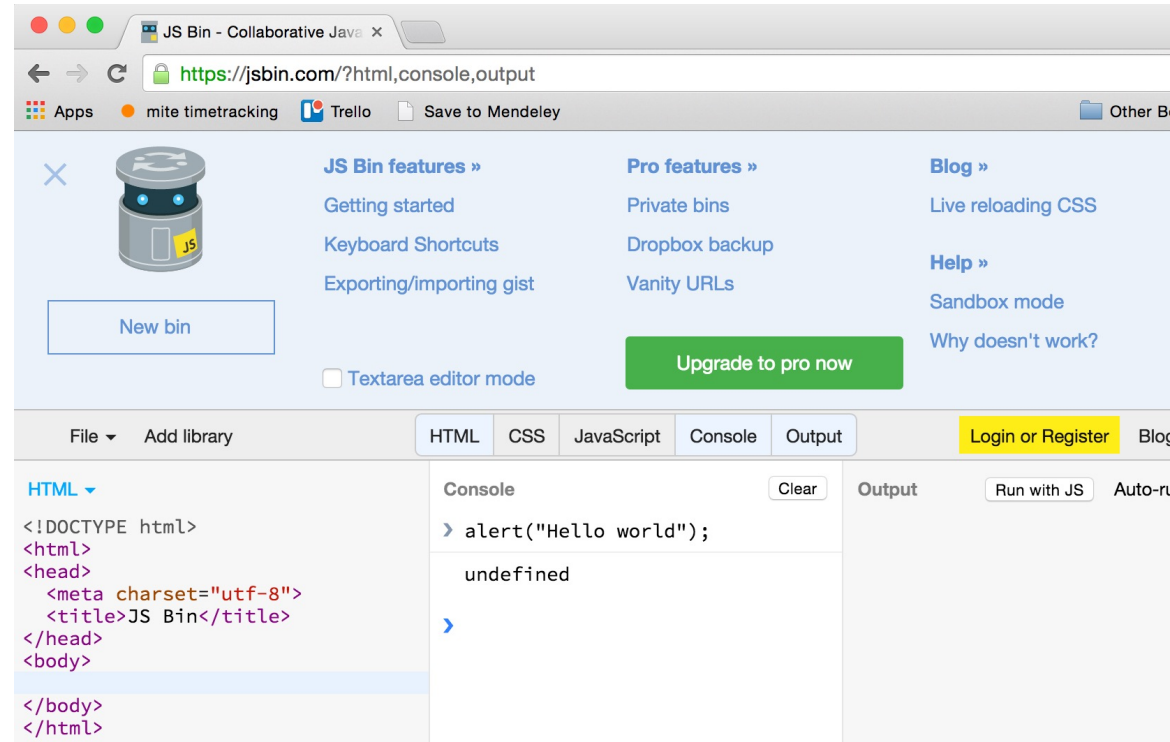
- Erweiterung des JavaScript Kerns, der API anbietet um mit dem Browser und den Elementen einer HTML Seite zu interagieren (DOM) – z.B. Hinzufügen von Elementen zu einer Seite, Reaktion auf Nutzereingaben
- **Business-Logik auf dem Client**

Später

Server

- Erweiterung des JavaScript Kerns, durch ein API, mit dem es möglich ist JavaScript auf einem Server auszuführen – z.B. Ablegen von Dateien im Dateisystem, Zugriff auf einen Datenbankserver
- **Business-Logik auf dem Server**

ERSTE VERSUCHE MIT JSBIN.COM



- Schnelles Testen von Code Snippets durch integrierte Konsole ohne Erstellung von größeren Programmen...

JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive Datentypen	number	string	boolean	undefined	null
Komplexe Typen	Array	Eigene Objekte	Function		

- JavaScript definiert primitive (= skalare) Datentypen und komplexe Typen / Objekte
- Skalare Datentypen speichern einen Wert, verfügen aber über keine Methoden (d.h. sind keine Objekte) – Aber mit String, Number und Boolean existieren Wrapperobjekte
- Zusammengesetzte / komplexe Typen (Objekte) können Arrays, eigene Objekte, Funktionen, etc. sein... Sozusagen alles andere außer primitive Datentypen.
- Eigene Objekte und Funktionen ➔ Beim nächsten Mal...

Vgl. z.B. <http://typedarray.org/JavaScript-refresh/#primitive-composite> oder
https://developer.mozilla.org/de/docs/Glossary/einfache_datenelemente

JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive
Datentypen

number

string

boolean

undefined

null

- Im Gegensatz zu Java, keine Unterscheidung der Datentypen (double, int, ...) bei der Definition / Initialisierung der Variablen
- Typ **number** wird für alle Ganzzahlen und Dezimalzahlen verwendet
- Da kein Konzept int und double, kein type-casting

```
var a = 4;  
var b = 6.9;  
var c = -5;
```

Operator, der Zeichenkette zurückgibt, die den Typ des ausgewerteten Operanden beschreibt...

```
// outputs: 1.3333333333333333  
console.log(a / 3);  
// outputs: number number number  
console.log ( typeof a, typeof b, typeof c );
```

?

Welche Ausgaben erzeugt der Code?



JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive
Datentypen

number

string

boolean

undefined

null

- Umwandlung einer Fließkommazahl in eine Ganzzahl durch Nutzung der Methode floor (= abrunden) des Math Objekts
- Math ist ein globales Objekt, das überall in JavaScript genutzt werden kann

```
var posNum = 45.95;  
var negNum = -45.95;
```

```
console.log(Math.floor(posNum));  
console.log(Math.floor(negNum));
```

?

Liste global verfügbarer Objekte:

https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects



JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive
Datentypen

number

string

boolean

undefined

null

- Vergleich von Strings erfolgt auf **lexikalischer** Basis mit ==
- Konkatenierung erfolgt mittels +

```
var firstNameOne = "Bill";  
var firstNameTwo = "Bill";  
var lastName = "Gates";  
  
var completeName = firstNameOne + " " +  
    lastName;  
//outputs: true  
console.log(firstNameOne == firstNameTwo);  
//outputs: Bill Gates  
console.log(completeName);
```

?

Welche Ausgaben erzeugt der Code?



JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive
Datentypen

number

string

boolean

undefined

null

- Bei Zugriff auf Eigenschaft oder Methode eines Strings mit `.`-Operator kann der String wie ein Objekt verwendet werden...
- Strings sind dann Objekte (wie in Java...) und bieten Methoden zum Zugriff auf die Daten des Objekts an...
- Iterieren eines Strings durch Abfrage der Länge und Zugriff über Indizes

```
var lastName = "Gates";  
for (var i = 0; i < lastName.length; i++) {  
    console.log(lastName.charAt(i));  
}
```

"G""a""t""e""s"

?

Welche Ausgaben erzeugt der Code?



JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive
Datentypen

number

string

boolean

undefined

null

- Strings lassen sich auch mit „Number“ konkatenieren
- Erzeugen eines Strings aus Zahlen durch Voranstellen eines leeren Strings möglich

```
var firstString = 1 + 2 + 3 + "hello";  
var secondString = "hello" + 1 + 2 + 3;  
var thisAlsoWorks = "" + 1 + 2 + 3;
```

```
console.log(firstString);    // outputs: 6hello  
console.log(secondString);  // outputs: hello123  
console.log(thisAlsoWorks); // outputs: 123
```

?

Welche Ausgaben erzeugt der Code?



JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive
Datentypen

number

string

boolean

undefined

null

- Achtung: Vergleich mit `==` führt zu automatischer Typkonvertierung
- `===` unterbindet Typkonvertierung

```
var a = true;  
var b = "true";  
var c = 1;  
var d = false;
```

```
console.log ( a == true ); // outputs: true  
console.log ( b == true ); // outputs: false  
console.log ( c == true ); // outputs: true  
console.log ( d == true ); // outputs: false  
console.log ( c === true ); // outputs: false
```

?

Welche Ausgaben erzeugt der Code?



JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive
Datentypen

number

string

boolean

undefined

null

- Variablen, denen noch kein Wert zugewiesen haben, haben den Wert **undefined**
- Was passiert in C? Was passiert in Java?

```
var firstNameOne = "Bill";  
var lastName;
```

```
console.log(firstNameOne); //outputs: Bill  
console.log(lastName);     //outputs: undefined
```

?

Welche Ausgaben erzeugt der Code?



JAVASCRIPTBASICS

ES GIBT 5 PRIMITIVES (NUMBER, STRING, BOOLEAN, UNDEFINED UND NULL) UND KOMPLEXE TYPEN

Primitive
Datentypen

number

string

boolean

undefined

null

- null repräsentiert das absichtliche Fehlen eines Werts
- null ist Schlüsselwort in JavaScript und wird ohne Anführungszeichen geschrieben

```
var userInput = null;  
  
if (userInput === null) {  
    console.log("ARGHL .. //outputs: ARGHL...")  
}
```

?

Welche Ausgaben erzeugt der Code?



Figuring out a variable's type

- In JavaScript, we have a keyword called `typeof` that returns the type of the variable.

```
typeof "";           // - "string"  
typeof 5;            // - "number"  
typeof false;        // - "boolean"  
typeof undefined;    // - "undefined"  
typeof null;         // this is not what we expect,  
                     // it returns "object"!
```

PROMPT UND MODULO

```
5 % 3 === 2 // true (the remainder when five is divided by 3 is 2)

var num = prompt('Please enter a whole number');
if ( num % 2 === 0 ) {
    console.log('the num variable is even!')
} else if ( num % 2 === 1 ) {
    console.log('the num variable is odd!')
} else {
    console.log('Hey! I asked for a whole number!');
}
```

QUELLEN

- Imbert, T. (2013). A JavaScript Refresh. Online verfügbar: <http://typedarray.org/JavaScript-refresh/>. Letzter Zugriff: 11.08.2015.
- Mozilla Developer Network. (2015b). A re-introduction to JavaScript (JS tutorial). Online verfügbar: https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript
- Mozilla Developer Network. (2015b). Introduction to Object Oriented JavaScript. Online verfügbar: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript#JavaScript_object_oriented_programming. Letzter Zugriff: 13.08.2015