Javscript Full Course Zusammenfassung

Term	Explanation	Video Reference
Operator		
Syntax		
String	Text	44.14
Concatenation	Combination of Strings	
Character	Letter/number/symbol in the text	59.14
Escape character	\character; only text	
Newlline character	\n; new line of text	
Interpolation	${}; insert values directly into the string	01.03.40
Variable	Container, saves content	01.54.00
Booleans	Either true or false	02.33.00
If Statements	**If (…** is true**)** then do **{…}** *Optional:* **else**, if the information between brackets isn't true do **{…}**	f
Truthy and falsy statements		03.23.30
Logical operators	And, or, not operator	03.03.46
Scope	Variable just exists inside {}	03:43:30
Return	Get a value out of a function, calling a function results in the return value (number, string), after return value function ends End a loop inside a function early with a return statement	03:46:30 09:27:00
Parameters	Value used in a function you can change each time you call a function, is written between ()	03:56:00
Object	Group multiple values, access values through goup.name and change	04:16:00
Dot notation	object.name	04:22:13
Property/field	Name of value in object	04:21:35
Property-value pair	Property and corresponding value in an object	04:21:45
Nested object	Object inside an object	04:43:30
Method	Function inside an object	04:46:30
Built-In Objects	By language provided Objects (e. g. console.log())	04:48:10
Json		04:48:30
Local Storage	Saving data after refreshing	04:54:00
Null vs. Undefined	Null: intentionally empty	05:08:50
Auto boxing	Wrap a value automatically into a special object	05:09:30

Reference	Objects, Object name that points to the location where the values are stored of the computer's memory	05:12:50
DOM Document Object Model	Whole Website/Document as an object	
Placeholder	Text written in an empty text box	06:16:05
Events	Clicks and keydowns	06:28:10
Events listeners	Onclick and onkeydown	06:28:10
Index	Position of a value in an array	07:49:20
Callback	Function used as a parameter in another function	09:40:17
Asynchronous vs. Synchronous code	Not wait for the previous line to finish vs. wait for the previous line to finish	09:43:35
Hoisting	Call a function before creating it	09:35:05
Class	Object generator	18:17:15
Private Porperty	Can only be used inside a class	18:25:55
Field	Property	18:27:10
HTML Attribute	Inside HTML Tag, specifies	
Recursion	Function calls itself	14:56:25

Javascript

Code	**Meaning**
let name = string or number	
name = string or number	
const name = string or number	
alert()	
console.log()	
if (… is true) {…} else if (… is true) {…} else {}	
eval()	
name.toString ()	
Math.round ()	
Math.random ()	
number +, -, *, /, ** number, %	%: division (Rest z.B. 10/3 = 1)
number ===, ==, !==, <, >, <=, >= number	
Name +=, -=, *=, /= number or ++, --	
// Js, /* */ multiline/CSS, <!-- --> HTML	
${}	
… ? '…is truthy' : '…is falsy' Truthy && do that Let name = truthy && string/number Let name = falsy && string/number	
True && true, true || false, !false	
If Truthy, do that || if not, do this	
name.length	

Const object = {name1: string/number, [‘name2’]: string/number…} Object.name = string/number Object.notExistingName = true Delete object.name Object.name or Object[‘name’]	Create object Change value Add value Delete value Access value
Const object = { Object1: {name1: string/number} } Const object = { Name1: function () {…} }	Nested object method
Typeof name	Check type of value
JSON.stringify() JSON.parse()	translate js code into json code translate json code into js code
localStorage.setItem(‘name’ , ‘value’) localStorage.getItem(‘name’) localStorage.removeItem()	

Const { name , … } = object;
Const object = { name }
Const object = {Name1 () {…} }
 	
Const object = {
  Name1: value1,
  Name2(this.name1){}
}
 	
Class name {
  Name1 = value1;
  …
  Constructor (parameter from new class) {do this whenever “new” object is created};
  …
  #Name2;
}
Const randomname = new name(parameter for constructor);
 	
Class name extends name2 {
  Variable;
  Constructor () {
  Super()
  Super.function()
  …
  }
}
	Const name = object.name; const … = object…
	Const object = { name : name }
	Const object = {Name1: function function1 () {…} }
	Objects shortcuts (05:18:20)
	Const object = {
	  Name1: value1,
	  Name2(object.name1){}
	}
	! name2: this.name1 = undefined!
	Create object
	
	Variable with # cannot be changed
	
	Only inside the class available
	Create new object with same data as the “name”
	Object oriented programming
	
	Create childclass with additional values to parentclass name2
	
	copy constructor/method from parentclass when defining a new constructor/method

Document.title = ‘…’ Document.body = ‘…’ Document.body.innerHTML = ‘…’ Document.querySelector(‘html-element/class’) .querySelectorAll(‘html-element/class’)	Change head Change body Change all HTML inside body Access first html-element/class-element

Document.querySelector('html-element/class').innerHTML = '…' .innerText .value	Access all html-elements/class-elements Change text of first html-element/class-element Access text Access text inside text box
<input>	Text Box
Console.log(event)	Object with information (values) about an event, e.g. event.key (tells you which key was pressed)
Number() String()	Convert a string into a number Convert a number into a string
Const name = […, …, …] Name[#] Name [#] = … Array.length Array.push(…) Array.splice(#, number)	Create array Get # value Change value Number of values in an array Add value to array Delete value of an array (starting from # as many as number)

While(this is so) {keep doing this}
For(variable; being so; keeps changing by this) {do this in addition}
…{if (i=…) {break;}}
…{if (i=…) {continue;}}
	Loop
	Loop
	
	Stop for loop early
	Skip one number of for loop

setTimeout(function () {}, time ms) setInterval(function () {}, time ms) clearInterval() array.forEach(function (value, index) {…}) .addEventListener(event, function) .filter((value, index) => {return}) .map((value, index) => {return})	Run a function after a certain time Run a function every … ms/time Stop interval built-in function For(…; i < array.length; …) {function} Onclick="" goes through every index and saves each value in an array, returns whatever is true in an array goes through every index and saves every value, returns every value changed in an array
Export variable; Import {name (as name)} from 'filepath'; Import * as {name} from 'filepath'; Import 'filepaht'; Name.name1	Use variables from another js file !!!: specify html type = module Imports everything in an object Import/run all of the code Access as a property/method
let geolocation = navigator.geolocation.getCurrentPosition(successfulCallback, errorCallback); or .watchPosition	Get location Auto update location
New Promise ((resolve) => {function(() => {resolve(value)})}).then((value) => {}) Promise.all([new Promise (), new Promise (), …]).then() Async function name () { Return New Promise(…)	Wait for resolve and then move to the next step and share values between steps Array of Promises New Promise ((resolve) => {function(() => {resolve(value)})})

Await function2(); Return value }	

Node JS

Require('…')	Cache a file and its content once and never again, parses json code automatically
Module.exports = name;	Export a variable so that it can be required in another file
http	

http.createServer((req, res) => {
  //js code
})
	what's being done on requests

http.listen(port, ip-address, () => {
  //js code
})
	what's being done when the server is running

req.url	Path
express	Express routing (expressjs.com)

Wieso kann man die Funktionen von packages gleich benutzen z. B. fs.readFile?

CMD

cd	Change Directory, choose Folder
cls	Verlauf löschen
dir	Show all files and folders
cd ..	Go to superior folder
cd .	Access to current folder
Mkdir Name	Add new folder
Echo Text > file-name	Add new file
Copy file-name new-file	

Error

Error	Lösung/Problem
Failed to load module script: Expected a JavaScript module script but the server responded with a MIME type of "". Strict MIME type checking is enforced for module scripts per HTML spec.	Type=module ohne module – type=module löschen
new.js:12 POST http://localhost:8000/create_new_game 500 (Internal Server Error) (anonymous) @ new.js:12	App.use(express.json()) Module.exports.name = name;

ReferenceError: Cannot access 'checkAddPlayer' before initialization	
Access to XMLHTTPRequest has been blocked by cors policy.	

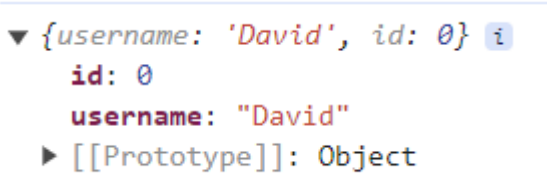
Bemerkungen:

- Portnumber! URL! StatusCode etc. like 200 = ok! Routing!
- Schriftlicher Kommentar, Quellen ?
- Wieso scheint es, als würde die Seite immer zweimal geöffnet? Oder weshalb wird zuerst der Post request gemacht und dann erst die Seite geöffnet?
- Kann es sein dass watchPosition einfach ein Intervall ist?
- Date funktioniert irgendwie noch nicht ganz (s. json file).
- Session
- Realtime live
- https ?
- classes
- libraries (like leaflet)
- ip, port etc.
- create an own method being used multiple times (like e. g. getcurrentposition)
- ssl, datenbank, server, html/CSS, Photoshop
- verschiedene Begriffe
- Webseite bei schwarzem Bildschirm weiterlaufen lassen
- Put, delete und patch neben get und post kennenlernen
- In der url steht ja der code und die id: da kann ja jeder einfach zu jedem Spieler wechseln – gibt es da eine Verschlüsselung?
- Verhindern, dass Website geschlossen wird, Warnung!

Fragen:

- 05:01:00
  LocalStorage: Das Object Score ist ja nirgendwo mehr definiert.
- Abgesehen von localStorage und Übersicht, wofür ist ein Object gut?
- Wie kann man document.body ändern/document.q….innerHTML löschen?
- Unterschied object und array?
- How to store advanced Todo?
- ToDo List: Über Zeile hinausschreiben?
- ToDo List: Beim Löschen eines Arrayelements verändern sich ja die Indexe aller nachfolgenden Elemente um eins. Dann sollte dies doch nicht mehr funktionieren?
- Wenn man eine Funktion nur definiert, wird sie dann trotzdem ausgeführt? (z. B. wenn sie einen eventListener beinhaltet)
- Weshalb gibt es immer gleich zwei neue Requests?
- Grund für CSS im Head und JS im Body?

- JSON Daten in HTML Seite einbauen?
- Durchsichtiger Platzhalter?
- Unterschied getElementById vs. querySelector?
- CSS und JS mit Server?
- Andere URL statt localhost/127.0.0.1?
- Variable.function?
- 12.00.00 Wieso Parameter button?
- Wofür sind classes? Kann ja dasselbe auch mit normalen Objects machen. Was bedeutet new vor einer Funktion?
- Was bedeutet ___ vor einem Namen/Variable etc.?
- Leerer Placeholder für dom inner HTML Text (bevor dieser angezeigt wird)?
- How to use https?
- Intervallfunktion beginnt erst nach einer Einheit? – der Code tut dies immer nach der angegebenen Zeit



- Was ist Prototype? Set_tasks.js Z. 15
- Weshalb muss man id und value bei select angeben und nicht einfach nur class und der value ist das HTML?
- Gibt es eine andere, bessere Möglichkeit um bei einem Objekt eine Property hinzuzufügen?
- Xhr.responseText statt JSON.parse(xhr.response)
- Bedeutet code etwas spezielles in JS?
- Wann passiert was, wenn man im Browser zurückgeht.
- Gibt es etwas bessers als pathname+search

To-Do:

11.10. 10.00	Node js in «JS Begriffe» ergänzen (mithilfe Express etc. alle Codes verstehen)
11.30	SSE verstehen (Youtube Video)
Pause	
12.30	Javascript Full Course fertig 2x v
Pause	
19.00	MongoDB Tutorial
20.00	Vercel zu Ähnlichkeiten mit localhost überprüfen
	Mindmap/Übersichtsplan

Screenshots: