**Modul 323 – Funktional Programmieren**

**Vorwort:**Grundlage für diese Aufgabe ist die Präsentation «Einführung JavaScript»

**Dauer:** 90 Minuten  
**Optional: 3**0 Minuten  
**Total:** 120 Minuten

Nachfolgend werden Sie sich mit den Grundlagen von JavaScript auseinandersetzen welches in diesem Modul als Programmiersprache dient an deren Beispiel die Prinzipien von funktionalem Programmieren erläutert wird.

JavaScript Kompendium

<https://www.w3schools.com/js/>

**Aufgabe 1)**

Installieren Sie NodeJS auf Ihrem System damit Sie für die späteren Aufgaben in der Lage sein werden, JavaScript direkt auf der Konsole auszuführen, ohne dass ein Browser benötigt wird.

1. **Installieren Sie NodeJS mithilfe des Tools «nvm». Schauen Sie, dass Sie mindestens Version 18 und im besten Fall eine LTS-Version verwenden.**

<https://github.com/coreybutler/nvm-windows>

ℹ️ Tipp: «nvm install lts && nvm use …»

|  |
| --- |
|  |

1. **Worum handelt es sich bei NodeJS und was kann damit erreicht werden?**

|  |
| --- |
| Es ist eine runtime environment um javascript ausserhalb vom Browser ausführen zu können |

1. **Welche Version von NodeJS haben Sie installiert und mit welchem Befehl kann man diese herausfinden?**

|  |
| --- |
| node -v |

**Starten Sie NodeJS im interaktiven Modus. Was können Sie damit nun anstellen?**

|  |
| --- |
| Mit .break kann man aus dem aktuellen Ausführungsprozess ausbrechen  Mit .editor kann man den Editor Modus beitreten  Mit .exit kann man die Nodejs REPL verlasssen  Mit .help kann man die verfügbare Befehle herausfinden  Mit .load kann man JavaScript-Code aus einer Datei in die REPL-Sitzung laden und ausführen. Zum Beispiel: .load myscript.js.  Mit .save kann man die aktuelle REPL Sitzung speichern z.B. .save session js |

**Aufgabe 2)**

Richten Sie sich eine Entwicklungsumgebung ein. Folgende Entwicklungsumgebung wird empfohlen:

Visual Studio Code: <https://code.visualstudio.com/download>

1. **Was glauben Sie, warum sich Visual Studio Code besonders für JavaScript eignet?**

|  |
| --- |
| Vs Code:  Startet schnell  Hat viele Extensions  Ist mit GIT integriert  Gute Codevervollständigung  Gute Debugging Funktion für JavaScript  Hat ein integriertes Terminal  Unterstützt Frameworks für js wie react, angular. Unterstützt auch nodejs  Vs Code hat eine aktive Community |

1. **Installieren Sie folgendes Plugin**

* Quokka

**Versuchen Sie mit diesem Plugin etwas herumzuspielen und beschreiben Sie die Funktionalität des Plugins kurz in eigenen Worten.**

|  |
| --- |
| Ermöglicht uns javascript während dem Schreiben auszuführen |

**Aufgabe 3)**

Beschäftigen Sie sich mit den verschiedenen primitiven Datentypen, welche in JavaScript existieren und wie diese Variablen zugeordnet werden können. Erstellen Sie für diese Aufgabe eine Datei mit dem Namen «**primitive\_datatypes.js**».

1. **Mit welchem Befehl kann in JavaScript etwas auf die Konsole ausgegeben werden? Und welchen Sinn und Zweck hat die Konsole im Normalfall?**

|  |
| --- |
| Console.log()  Man kann es für unit tests brauchen oder allgemein um was zu testen |

1. **Definieren Sie in dieser Datei eine Variable vom Typ «String» und geben Sie diese auf der Konsole aus. Führen Sie danach Ihre Datei mit Quokka oder NodeJS aus. Wie kann eine JavaScript Datei mit Node ausgeführt werden?**

|  |
| --- |
|  |

1. **Erweitern Sie das Script «primitive\_datatypes.js». Definieren Sie weitere Variablen mit den Datentypen «Number», «Boolean» und «undefined». Geben Sie auch diese jeweils in der Konsole aus. Führen Sie danach das Script aus und schauen Sie, ob die erwartete Ausgabe erscheint.**

|  |
| --- |
|  |

1. **Deklarieren Sie eine Variable mit dem «var» Schlüsselwort und weisen Sie dieser einen Wert vom Typ «String» zu. Weisen Sie danach einen anderen Wert der gleichen Variable zu welcher den Datentype «boolean» hat. Was stellen Sie fest?**

|  |
| --- |
| Das Datentyp passt sich in der Variable an |

1. **Deklarieren Sie 2-mal eine Variable mit dem «var» Schlüsselwort und der gleichen Bezeichnung. Was stellen Sie fest?**

|  |
| --- |
| Es ist möglich zwei var mit dem gleichen Namen zu erstellen |

1. **Deklarieren Sie 2-mal eine Variable mit dem «let» Schlüsselwort und der gleichen Bezeichnung. Was stellen Sie fest?**

|  |
| --- |
| Es ist nicht möglich zwei let mit dem gleichen Namen zu erstellen |

1. **Welches Schlüsselwort zur Variablendeklaration würden Sie generell vorziehen «var» oder «let»? Und aus welchem Grund?**

|  |
| --- |
| Let, weil es eben kein Durcheinander gibt mit den Variablen mit gleichen Namen.  Ein weiterer Grund wäre der Scope, var ist in der ganzen Klasse zugreifbar und let nur in dem Block, in welchem sie deklariert wurde, was auch einheitlicher ist. |

1. **Deklarieren Sie eine Variable mit dem «const» Schlüsselwort und weisen Sie dieser einen Wert zu. Versuchen Sie danach, den Wert anzupassen. Was stellen Sie fest?**

|  |
| --- |
| Es ist nicht möglich den Wert einer const Variable zu verändern |

1. **Deklarieren Sie eine Variable mit dem «const» Schlüsselwort und weisen Sie dieser den Wert «undefined» zu. Deklarieren Sie eine weitere Variable mit dem «const» Schlüsselwort und weisen Sie dieser den Wert «null» zu. Vergleichen Sie danach diese 2 Variablen mit dem Vergleichsoperator «==» oder «===». Was stellen Sie fest und können Sie dieses Resultat erklären?**

|  |
| --- |
| Das Resultat ist, dass die Werte gleich sind. Weil == nur den Wert vergleicht, aber keinen Datentyp.  Würden wir es mit === kontrollieren, würde false rauskommen, da die Datentypen unterschiedlich sind |

1. **Welche Namenskonvention für das Benennen von Variablen haben Sie für die Übungen verwendet und welche ist in der Welt von JavaScript am beliebtesten?**

|  |
| --- |
| Variablen: Lower camelCase, Selbstbeschreibend  Konstanten: Grossbuchstaben mit Unterstrich als Abtrennung (NOT\_PI)  Klassen und Konstruktoren: Pascalcase (Person), (CarModel)  Dateinamen: Kebabcase (test-task.js)  Funktionen: aussagekräftige Verben, Lower camelCase |

**Aufgabe 4)**

Beschäftigen Sie sich mit dem Definieren von Strukturen und Objekten, um mehrere Variablen in grösseren Formationen zusammenfassen zu können. Erstellen Sie für diese Aufgabe eine Datei mit dem Namen «**objects.js**».

1. **Deklarieren Sie eine Variable welche als Wert ein Objekt hat, dass mindestens 3 Eigenschaften und verschiedene Datentypen hat. Z.B. Person, Auto usw. Geben Sie nun die einzelnen Eigenschaften separat auf die Konsole aus.**

|  |
| --- |
| Oder |

1. **Deklarieren Sie eine Variable welche als Wert ein Objekt hat, und dieses Objekt soll wiederum eine Eigenschaft besitzen, welche ein Objekt mit mehreren Eigenschaften hat. Ist das möglich und wie tief kann man Objekte verschachteln?**

|  |
| --- |
| Es ist möglich und man kann sie so tief verschachteln, bis man kein RAM mehr hat. |

**Aufgabe 5)**

Beschäftigen Sie sich mit Listen und den verschiedenen Operationen, welche auf diesen vorgenommen werden können. Erstellen Sie für diese Aufgabe eine Datei mit dem Namen «**arrays.js**».

1. **Erstellen Sie eine Liste welche 3 Elemente vom Typ String enthält. Geben Sie diese in der Konsole aus.**

|  |
| --- |
|  |

1. **Geben Sie nun das zweite Element aus. Wie kann direkt auf ein spezifisches Element zugegriffen werden?**

|  |
| --- |
|  |

1. **Fügen Sie der zuvor erstellten Liste ein weiteres Element des Typs «Objekt» hinzu. Ist das möglich?**

|  |
| --- |
|  |

1. **Entfernen Sie das Element am Anfang der Liste und am Ende der Liste.**

|  |
| --- |
|  |

**Aufgabe 6)**

Beschäftigen Sie sich mit Kontrollstrukturen und wie mit diesen der Ablauf von Code gesteuert werden kann. Erstellen Sie für diese Aufgabe eine Datei mit dem Namen «**control\_structures.js**».

1. **Erstellen Sie eine Variable welche mithilfe von Math.random() eine zufällige Zahl zwischen 0 und 1 als Wert erhält. Erstellen Sie danach eine Kontrollstruktur, die folgendermassen funktionieren soll.**

* **Wenn > 0.5 = Konsolenausgabe «Grösser als 0.5»**
* **Wenn erste nicht zutrifft aber > 0.25 = Konsolenausgabe «Grösser als 0.25»**
* **Wenn keines zutrifft = Konsolenausgabe «Grösser kleiner oder gleich 0.25»**

**Welche Kontrollstruktur verwenden Sie?**

|  |
| --- |
|  |

1. **Erstellen Sie eine Variable welche mithilfe von «Math.floor(Math.random() \* 5)» eine zufällige ganzzahlige Zahl zwischen 0 und 5 als Wert erhält. Erstellen Sie danach eine Kontrollstruktur, die folgendermassen funktionieren soll.**

* **Wenn Zahl = 1 -> Konsolenausgabe «Der Wert der Zahl ist exakt 1»**
* **Wenn Zahl = 3 -> Konsolenausgabe «Der Wert der Zahl ist exakt 3»**
* **Wenn eine andere Zahl kommt -> Konsolenausgabe «Der Wert der Zahl ist weder 1 noch 3»**

**Welche Kontrollstruktur verwenden Sie?**

|  |
| --- |
|  |

**Aufgabe 7)**

Beschäftigen Sie sich mit Schleifen und wie mit diesen Listen oder auch Objekte verarbeitet werden können. Erstellen Sie für diese Aufgabe eine Datei mit dem Namen «**loops.js**».

1. **Erstellen Sie eine Schleife, welche auf der Konsole von 0-10 hochzählt.   
   Welchen Schleifentyp verwenden Sie?**

|  |
| --- |
|  |

1. **Erstellen Sie eine Schleife. Diese soll in jedem Durchlauf eine neue Zufallszahl zwischen 1-10 generieren diese ausgeben und wenn die Zahl grösser als 8 ist, soll die Schleife abgebrochen werden.  
   Welchen Schleifentyp verwenden Sie?**

|  |
| --- |
|  |

1. **Erstellen Sie einen Array mit 5 beliebigen Elementen. Erstellen Sie danach eine Schleife, welche alle Elemente des Arrays durchläuft und jedes Element einzeln auf der Konsole ausgibt.**

**Welchen Schleifentyp verwenden Sie?**

|  |
| --- |
|  |

**Aufgabe 8)**

Beschäftigen Sie sich mit Funktionen und wie diese dazu verwendet werden können JavaScript Befehle zu bündeln und wiederverwendbar zu machen. Erstellen Sie für diese Aufgabe eine Datei mit dem Namen «**functions.js**».

1. **Definieren Sie eine Funktion, welche 2 Parameter entgegennimmt und aus diesen 2 Zahlen das Produkt bildet und auf der Konsole ausgibt. Rufen Sie dann die Funktion mit verschiedenen Parametern auf.**

|  |
| --- |
|  |

1. **Definieren Sie eine Funktion, welche 1 Parameter (Grad Celsius) entgegennimmt, diesen Wert in Fahrenheit umrechnet und auf der Konsole ausgibt. Rufen Sie dann die Funktion mit verschiedenen Parametern auf.**ℹ️ Tipp: °F = °C \* 1,8 + 32

|  |
| --- |
|  |

1. **Deklarieren Sie eine Variable, weisen Sie dieser Variable danach ein Lambda/Arrow Funktion zu, welche aus 2 Parametern das Modulo bestimmt. Rufen Sie dann die Funktion mit verschiedenen Parametern auf. Welcher Unterschied besteht zwischen einer regulären Funktion und einer Lambda/Arrow Funktion? Verwenden Sie dazu Google.**

|  |
| --- |
|  |

1. **Was können Sie zum Schlüsselwort «this» in JavaScript sagen? Verwenden Sie dazu Google.**

|  |
| --- |
|  |

**Aufgabe 9)**

Erstellen Sie einen github Account. Sie werden in diesem Kurs alle Ihre Aufgaben welche Source Code beinhalten auf diese Plattform hochladen. Achten Sie dabei immer, dass Ihre Repositories öffentlich zugänglich sind.

1. **Erstellen Sie einen Account. Um was handelt es sich bei Github?**

<https://github.com/signup>

|  |
| --- |
|  |

1. **Installieren Sie git auf Ihrem Hostsystem. Um was handelt es sich bei git?**

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

ℹ️ Tipp: «git init»

|  |
| --- |
|  |

1. **Was ist der Unterschied zwischen git und github?**

|  |
| --- |
|  |

1. **Initialisieren Sie lokal ein neues git Repository und commiten Sie den Source Code, der in diesem Arbeitsblatt angefallen ist. Achten Sie darauf, dass Sie nur die Dateien commiten welche relevant sind. Alle anderen Dateien sollen in der Datei «.gitignore» aufgelistet werden damit diese nicht aus Versehen commited werden.**ℹ️ Tipp: «git add . && git commit -m ‘Initial commit’»

|  |
| --- |
|  |

1. **Erstellen Sie unter Ihrem github Account ein neues passendes Repository mit sprechender Bezeichnung. Verknüpfen Sie dieses Repository und pushen Sie den Source Code auf github. Fügen Sie danach den Link zu Ihrem Repository in die untenstehende Textbox ein.**

|  |
| --- |
|  |

**Aufgabe 10) (Optional)**

Falls Sie mit dem Arbeitsblatt bereits fertig sind, können Sie sich den nachfolgenden Onlinekurs anschauen und selbstständig anfangen die Übungen zu erledigen. Das kann Ihnen zusätzlich helfen, in das ganze Thema hereinzukommen.

<https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/>

💡 Um die Übungen und den Fortschritt speichern zu können, müssen Sie einen gratis Account auf freecodecamp erstellen.