

1AHWII, 2. Praktische Leistungsfeststellung

14. Dezember 2023, Lehrer: Georg Graf

Aufgabe 1: Installation node

Am Ende dieser Aufgabe wirst Du in der Lage sein, auf Deinem PC Javascript-Programme ohne einen Webbrowser auszuführen.

Um das Installationspaket herunterzuladen, gehe auf <https://nodejs.org/en/download> und wähle "LTS" sowie das "Installer" Paket.

Installiere das Paket und lasse dabei alle vorgeschlagenen Einstellungen wie es vorgeschlagen wird, also einfach immer auf "weiter" klicken bis die Installation fertig ist.

Schließe jetzt alle Terminalfenster sowie VSCode, denn bei der Installation werden Umgebungsvariablen gesetzt, damit die Node Installation auch gefunden wird.

Öffne VSCode erneut und stelle z.B. so fest, daß die Installation erfolgreich war:

```
$ node --version  
v20.9.0.  
$
```

Aufgabe 2: Node erkunden

Starte hierzu node ohne Kommandozeilenargumente und gib einfache, wie im Unterricht gelernte, Befehle ein, z.B.:

```
$ node
Welcome to Node.js v20.9.0.
Type ".help" for more information.
> const a = 3;
undefined
> a = 7;
Uncaught TypeError: Assignment to constant variable.
> let b = 3;
undefined
> b = 7;
7
> b
7
> function print(argument) {
... console.log(argument);
... }
undefined
> print("Hallo!");
Hallo!
undefined
```

Mit **CTRL-D** kannst Du node wieder verlassen und auf die System-Kommandozeile zurückkehren.

Anmerkung zu den folgenden Aufgaben

Wichtig: Die hier erstellten Funktionen müssen **exakt** (Groß-/Kleinschreibung!) so heißen wie hier in der Angabe vorgegeben, sonst werden sie nicht bewertet!

Aufgabe 3: app.js

Stelle im Terminal sicher, daß Du im Angabe-Ordner bist mit dem Kommando **pwd** (Print Working Directory). Dort ist die Datei **app.js**, welche Javascript Code enthält. Ansonsten verwende im Terminal das Kommando **cd <pfad>**, um in den gewünschten Ordner zu wechseln.

```
$ node app.js
app.js beginnt.
app.js endet.
$
```

Für die folgenden Aufgaben muss die Datei **app.js** von Dir verändert werden.

Aufgabe 4: Funktion **plus(a,b)**

Erstelle eine Funktion mit dem Namen **plus**, welche 2 Parameter annimmt und deren Summe zurückliefert ("return").

Erstelle jetzt einen "git commit" und "push"-e diesen Zustand gleich auf github hoch.

Aufgabe 5: Funktion **kleinerAls5(n)**

Erstelle eine Funktion `kleinerAls5`, welche `true` liefert, wenn der übergebene Wert kleiner als 5 ist, und im anderen Fall den Wert `false`.

Committe und pushe wieder.

Aufgabe 5: Funktion `wurzel(r)`

Eine Funktion zum Wurzelziehen wird mit `Math.sqrt()` zur Verfügung gestellt. Schreibe selbst eine Funktion `wurzel()` und `return`-e die Quadratwurzel der übergebenen Zahl.

Committe und pushe wieder.

Aufgabe 6: Schleife

Nur falls ein Sehr Gut angestrebt wird, dies haben wir im Unterricht noch nicht besprochen: Erstelle eine Funktion `schleife()`, welche nichts retourniert, sondern alle natürlichen Zahlen von 1 (inklusive) bis 10 (inklusive) auf die Konsole ausgibt (`console.log()`).

Committe und pushe wieder.

Gutes Gelingen!