## Ziel

Die Lernenden können eigene Funktionen mit JavaScript programmieren, verstehen deren Nutzen und können mindestens zwei Vorteile von Funktionen aufzählen.

## Ausgangslage

Funktionen erleichtern das Leben eines Entwicklers enorm. Sie sind ein zentrales Konstrukt jeder Programmiersprache und insbesondere für JavaScript; denn bei JavaScript wird oft mit sogenannten Callbacks gearbeitet. Es ist sehr wichtig für Ihre Zukunft als Programmierer, dass Sie verstehen, was Funktionen sind und was man damit tun kann.

## Hilfsmittel

Buch „Schrödinger lernt HMTL5, CSS3 und JavaScript“, Kapitel 11.

Internet:

* MDN (JavaScript Tutorial & Referenz): <https://developer.mozilla.org/de/docs/Web/JavaScript>
* JavaScript bei w3schools: <http://www.w3schools.com/js/default.asp>
* etc.

## Vorgehensweise

1. Lesen Sie im Buch "Schrödinger" im Kapitel 11 die Seiten 436 bis 446.
2. Beantworten Sie die Theorie-Fragen
3. Lösen Sie die Praxisaufgaben.

# Theorie-Fragen

1. Was verstehen Sie unter einer „Funktion“? Erklären Sie in Ihren eigenen Worten.

*Funktionen sind Aufgaben die mehrere Parameter übergeben können und am Schluss einen Rückgabewert haben.*

1. Was genau ist ein Parameter?

*Parameter sind die Dinge mit denen die Funktion arbeiten soll.*

1. Erklären Sie, wozu in einer Funktion der Befehl *return* verwendet wird.

*Der Befehl return gibt das Ergebnis der Funktion zurück.*

1. Nennen Sie mindestens zwei Vorteile von Funktionen (Stichworte genügen).

*- Wiederverwendbar, muss nicht alles erneut geschrieben werden*

*- kleine Arbeitsschritte werden zu einem, grossen zusammengefasst*

1. Wozu sind Funktionen gut? Erklären Sie, worin Ihrer Meinung nach der grösste Nutzen von Funktionen besteht.

*Code übersichtlicher zu halten.*

1. Was ist der Unterschied zwischen einer globalen Variable und einer lokalen Variable?

*Globale Variablen betreffen auch alle Variablen die ausserhalb der Funktion deklariert wurden.*

*Lokale Variablen hingegen betreffen nur die Variablen die innerhalb der Funktion deklariert wurden.*

1. JavaScript stellt bereits einige nützliche Funktionen zur Verfügung. Man spricht in diesem Zusammenhang auch von globalen, nicht an ein Objekt gebundenen Funktionen. Ein Beispiel für eine solche Funktion ist *parseInt()*, eine Funktion, die einen *String* in eine *Number* umwandelt und zurückgibt.   
   Notieren Sie drei weitere globale Funktionen inklusive einer kurzen Beschreibung, was die Funktionen tun. Recherchieren Sie dazu im Internet in der JavaScript-Referenz Ihrer Wahl.

|  |  |
| --- | --- |
| **Funktion** | **Beschreibung** |
| parseInt() | Liest ein String-Argument ein und gibt eine ganze Zahl zurück. |
| toString() | Formt eine Zahl zu einem String. |
| charAt() | Gibt einen beliebigen Charakter eines Textes aus. |
| parseFloat() | Liest ein String-Argument ein und gibt eine Kommazahl zurück |

1. Erklären Sie, was eine anonyme Funktion ist und machen Sie ein Beispiel für die Verwendung einer anonymen Funktion. Recherchieren Sie dazu ggf. im Internet, um eine Antwort zu finden.

*Anonyme Funktionen sind Funktionen ohne Namen, die sofort ausgeführt werden. In der Verarbeitung von Events werden häufig anonyme Funktionen verwendet*

*(function () {*

*var autor = 'Pratchett';*

*var titel = 'Schöne Scheine';*

*var buch = [autor, titel];*

*alert(buch.join(','))*

*})();*

1. Betrachten Sie die folgenden JavaScript-Beispiele und notieren Sie, was nach Durchlauf des Codes im Browserfenster ausgegeben wird.

**Beispiel 1**

var a = 20;

addieren();

document.write(a); // Ausgabe im Browser?

function addieren()   
{  
 a += 20;   
} *// 40*

**Beispiel 2**

var a = 20;

addieren(a);

document.write(a); // Ausgabe im Browser?

function addieren(zahl)   
{

zahl += 20;  
} *// 20*

**Beispiel 3**

var a = 20;

addieren(a);

document.write(a); // Ausgabe im Browser?

function addieren(zahl)   
{

zahl += 20;  
 a += 20;  
} *// 40*

**Beispiel 4**

var a = 20;

document.write(addieren(a)); // Ausgabe im Browser?

function addieren(zahl)   
{  
 zahl += 20;   
} *//*

1. JavaScript bietet die Möglichkeit zu prüfen, ob eine bestimmte Funktion existiert oder nicht. Recherchiere im Internet wie das gemacht werden kann und vervollständige untenstehenden Code-Block entsprechend, so dass die Funktion „addieren()“ nur dann aufgerufen wird, wenn sie auch wirklich existiert.

var a = 0;

if (typeof addieren == "function"){

a = addieren();

document.write(a);

}

function addieren()

{

a += 20;

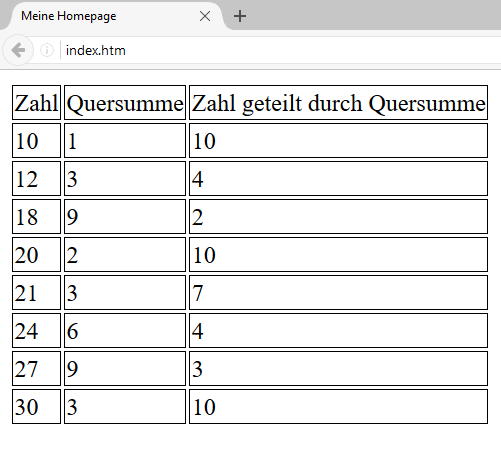
return a;

# }Praxisaufgaben

1. Schreiben Sie eine Funktion, welche als Parameter eine beliebige positive Ganzzahl (Integer) erhält und für diese Zahl die Quersumme berechnet. Das Resultat (die berechnete Quersumme) soll an den Aufrufer der Funktion zurückgegeben werden.   
   Prüfen Sie in Ihrer Funktion auch, ob wirklich eine Ganzzahl als Parameter übergeben wurde und geben Sie 0 oder NaN zurück, falls keine Ganzzahl übergeben wurde.   
   Teste Sie Ihre Funktion indem Sie sie aufrufen und das zurückgegebene Resultat am Browser ausgeben (document.write()).
2. Programmieren Sie eine Funktion mit dem Namen „durchQuersummeTeilbareZahlen“ , welche zwei Parameter erhält. Beide Parameter sollen eine Ganzzahl sein und die Funktion soll folgendes tun.
   * für alle Zahlen, die sich im Bereich der ersten bis zur zweiten Zahl befinden, wird geprüft, ob sie durch ihre eigenen Quersumme ohne Rest teilbar sind
   * alle Zahlen, die durch ihre Quersumme ohne Rest teilbar sind, werden in eine HTML-Tabelle verpackt; diese Tabelle soll 3 Spalten haben, in der 1. Spalte soll die Zahl stehen, in der 2. Spalte die Quersumme und in der 3. Spalte das Resultat der Division der Zahl durch ihre Quersumme
   * die komplette HTML-Tabelle wird am Schluss an den Aufrufer der Funktioen als String zurückgeben

Zur Veranschaulichung ein Beispiel: Der Aufruf der Funktion mit den Zahlen 10 und 30 muss zu unten abgebildetem Ergebnis führen.

*z.B. document.write(durchQuersummeTeilbareZahlen(10, 30));*



1. Erstellen Sie eine Funktion, die für einen bestimmten Text zählt, wie viele Vokale (a, e, i, o, u) er enthält. Das heisst, für den Text „Es geht mir gut.“ gibt die Funktion „4“ zurück.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>höhö</title>

</head>

<body>

<script type="text/javascript">

function Quersumme(zahl) {

var summe = 0;

var segs = (zahl+'').split('');

for (var i=0; i<segs.length; i++)

summe += parseInt(segs[i]);

return summe;

}

function durchQuersummeTeilbareZahlen(zahl1, zahl2) {

var returnvar = "<table><tbody>";

for (var i = zahl1; i < zahl2; i++) {

var zahl3 = Quersumme(i);

if (i % zahl3 === 0){

var zahl4

}

var segs = (zahl+'').split('');

for (var i=0; i<segs.length; i++)

summe += parseInt(segs[i]);

return summe;

returnvar += "</tbody></table>"

}

document.write(durchQuersummeTeilbareZahlen(10, 30));

</script>.

</body>

</html>

3.

<script type="text/javascript">

function zahler(str) {

var anz = 0;

for (var i = 0; i < str.length; i++) {

if (str.toLowerCase().charAt(i)=="a" || str.toLowerCase().charAt(i)=="e" ||

str.toLowerCase().charAt(i)=="i" || str.toLowerCase().charAt(i)=="o" ||

str.toLowerCase().charAt(i)=="u") {

++anz;

}

}

return(anz);

}

document.write(zahler("aeiou"));

</script>

# Erwartete Resultate

1. Alle Theoriefragen sind schriftlich beantwortet und das Dokument mit den Antworten ist an dem dafür vorgesehenen Platz abgelegt/gespeichert.
2. Die Praxisaufgaben sind gelöst und getestet, so dass sie einwandfrei funktionieren. Die Lösung sind im dafür vorgesehenen Verzeichnis abgelegt.
3. Erledigte Arbeiten und gemachte Erfahrungen sind im Arbeitsjournal geloggt.