

Aufgabe 1

Das nachfolgende Programmstück erstellt ein indiziertes Array und manipuliert es anschliessend.

```
<html><body><pre><script type="text/javascript">
var namen = ["A.N.Mut", "B.E.Dauern", "H.O.Sen", "N.I.Emals", "E.L.Efa"];
window.alert(namen);

</script></pre></body></html>
```

- Geben Sie mittels der Eigenschaft `length` die Anzahl der Arrayelemente aus.
- Fügen Sie mit der Methode `push("L.E.Tzter")` ein neues Arrayelement am Ende an.
- Fügen Sie mit der Methode `unshift("E.R.Ster")` ein neues Arrayelement am Anfang hinzu.
- Entfernen Sie nun mittels der `pop()`-Methode das letzte Arrayelement und ...
- ... mittels `shift()` ebenfalls das erste Arrayelement.

Geben Sie das veränderte Array jeweils mit der Methode `window.alert` aus.

Aufgabe 2

Die Bearbeitung aller Arrayelemente der Reihe nach erfolgt vorzugsweise mittels einer `for`-Schleife. Hier wird die Summe aller Zahlen eines Arrays gebildet.

```
var z = [-12, 38, 7, 42, -22, 17, -5, 9, 27], sum = 0;
for (var i=0; i < z.length; i++) {
  document.write(sum + " + " + z[i] + " = ");
  sum += z[i];
  document.write(sum + "\n");
}
```

- Ermitteln Sie nun das Minimum aller Arrayelemente. Setzen Sie hierzu eine Variable `var zmin = 10000`; auf einen hinreichend grossen Wert und vergleichen dann jedes Arrayelement mit Diesem. Ist ein Element kleiner als das Minimum, so wird das Minimum auf den Wert jenes Arrayelements gesetzt.
- Bestimmen Sie auf die gleiche Weise das Maximum.

Aufgabe 3

Wir wollen nun die Wahrscheinlichkeit simulieren, nach der von 50 Personen in einem Raum Einige denselben Geburtstag haben. Dazu erzeugen wir in einer Schleife mittels `Math.random` Pseudozufallszahlen zwischen 0 und 1. Wir skalieren diese anschliessend auf 365, schneiden mittels `Math.floor` die Nachkommastellen ab und speichern die Werte in ein Array. Das Ganze sieht so aus:

```
var z = [];
for (var i=0; i < 50; i++) {
  z[z.length] = Math.floor(Math.random()*365);
}
window.alert(z);
```

- Wir wollen nun diejenigen Geburtstage ermitteln, die mehrfach vorkommen. Hierzu betrachten wir in einer **weiteren äusseren Schleife** alle Zahlen im Array. In einer **inneren Schleife** vergleichen wir alle jeweils **nachfolgenden** Zahlen mit dieser. Im Falle einer Gleichheit geben wir sie aus.
- Geben Sie nicht nur die *Zahlenzwillinge*, sondern auch die *Anzahl* der Zwillinge aus.

(Hinweis: Legen Sie hierzu eine geeignete Variable *count* an und inkrementieren Sie diese jeweils im Falle eines Treffers.)

Aufgabe 4

Wir erzeugen nun ein Array von 2D-Vektoren (*assoziative Arrays*) und geben diese unmittelbar in einer Schleife wieder aus. Beachten Sie bitte die Zugriffsweise mittels des "."-Operators.

```
var v = [{x:15,y:-8},{x:12,y:9},{x:4,y:17},{x:-18,y:-7},{x:22,y:-6}];
for (var i=0; i<v.length; i++) {
  document.write("{x:"+v[i].x+",y:"+v[i].y+"}, ");
}
```

- Verwenden Sie bitte statt des "."-Operators den "[]"-Operator und weisen Sie so die Äquivalenz nach.
- Ermitteln Sie nun in einer weiteren Schleife die Länge eines jeden Vektors und geben Sie diese aus (*document.write*). Verwenden Sie hierbei den *Satz des Pythagoras* $l = \sqrt{x_2 + y_2}$.
Hinweis: [Math.sqrt\(arg\)](#) zur Berechnung der Quadratwurzel mit dem jeweiligen numerischen Argument verwenden.
- Suchen Sie in Analogie zur vorangegangenen Aufgabe den Vektor maximaler Länge.

Aufgabe 5

Für Ambitionierte

Wir geben nun eine Matrix als mehrdimensionales Array vor und multiplizieren die auf 3D erweiterten Vektoren aus der vorangegangenen Aufgabe (*diesmal als indiziertes Array*) damit von links.

$$\begin{bmatrix} 0.86 & -0.5 & 2 \\ 0.5 & 0.86 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Die Matrix und das Vektorarray sehen folgendermassen aus:

```
var v = [[15,-8,1],[12,9,1],[4,17,1],[-18,-7,1],[22,-6,1]];
var m = [[0.86, -0.5, 2],
         [0.5, 0.86, -1],
         [0, 0, 1]];
```

Die Matrix/Vektormultiplikation basiert auf folgendem Pseudocode:

```
variable vs for transformed vector
for every Matrixrow i {
  vs[i] = 0;
  for every Matrixrowelement and Vectorelement j {
    vs[i] = vs[i] + m[i][j]*v[j];
  }
}
```

Wandeln Sie den Pseudocode in lauffähiges Javascript.