

## Musterlösungen zur Implementierung der Sortierverfahren

→ Prinzipien siehe PWiki unter Punkt 3 Bereich Algorithmierung...

### Sortieren durch Auswählen (Select)

```
x 1 function select(){
2   var zahl = [50,61,7,8,9,1,2,3]; // Feld mit 8 vorgegebenen Werten
3   for (var i = 0; i <=7; i++){
4     var mini=i;           // Setze Merker
5     for (var j = i + 1; j < 8; j++){
6       if (zahl[j] < zahl[mini]) { // Vergleich Zahl an Stelle j mit Merker
7         mini=j;}
8     }
9     var h = zahl[i];       // vertauschen
10    zahl[i] = zahl[mini];
11    zahl[mini] = h;
12    // console.log(zahl[i]); // Ausgabe
13  }
14  console.log(zahl)
15 }
16
```

```
x 1 select()
2
```

**speichern & ausführen**

> 1,2,3,7,8,9,50,61

## Sortieren durch Einfügen (Insert)

```
x 1 function insert(){
  2 var zahl = [450,61,17,9,9,-1,2,3];
  3 for (var i = 1; i <8; i++){
  4   var merke=zahl[i];
  5   var j=i;
  6   while (merke<zahl[j-1]) {
  7     zahl[j]=zahl[j-1];
  8     j=j-1;
  9   }
 10   zahl[j] = merke;
 11 }
 12 /*for (var i = 0; i < 8; i++){
 13 console.log(zahl[i]);
 14 }*/
 15
 16
 17 console.log(zahl);
 18 }
 19
```

```
x 1 insert()
  2
```

**speichern & ausführen**

```
> -1,2,3,9,9,17,61,450
```

## Sortieren durch Austauschen (Bubble)

```
x 1 function bubble(){
2   var n = 8;
3   var zahl = [5,61,7,8,9,1,2,3];
4   do {
5     var vertauscht = false;
6     for (var i = 0; i < n-1; i++){
7       if (zahl[i+1] < zahl[i]) {
8         //-----Tausch-----
9         var h = zahl[i];
10        zahl[i] = zahl[i + 1];
11        zahl[i + 1] = h;
12        vertauscht = true;
13      }
14    }
15    n--;
16  }
17  while (vertauscht == true);
18  /*for (var i = 0; i < 8; i++){
19    console.log(zahl[i]);
20  }*/
21
22  console.log(zahl)
23 }
24
```

```
x 1 bubble()
2
```

**speichern & ausführen**

> 1,2,3,5,7,8,9,61

## Quicksort – beachte den **rekursiven** Aufruf dabei

```
x 1 zahl = [7, -2, 4, 211, 16, 5, 0, -4, 2]
  2
  3 function Teilung(feld, start, end){
  4
  5     const pivotWert = feld[end];
  6
  7     let pivotIndex = start;
  8
  9     for (let i = start; i < end; i++) {
 10         if (feld[i] < pivotWert) {
 11             // tauschen
 12             [feld[i], feld[pivotIndex]] = [feld[pivotIndex], feld[i]];
 13
 14             pivotIndex++;
 15         }
 16     }
 17
 18
 19     [feld[pivotIndex], feld[end]] = [feld[end], feld[pivotIndex]]
 20     return pivotIndex;
 21 };
 22
 23
 24
 25 function quickSort(feld, start, end) {
 26     // Base case or terminating case
 27     if (start >= end) {
 28         return;
 29     }
 30
 31     let index = Teilung(feld, start, end);
 32
 33     // rekursiver Aufruf
 34     quickSort(feld, start, index - 1);
 35     quickSort(feld, index + 1, end);
 36 }
 37
```

```
x 1 quickSort(zahl, 0, zahl.length - 1)
  2
  3 console.log(zahl)
  4
```

**speichern & ausführen**

```
> -4,-2,0,2,4,5,7,16,211
```