

Mit einer Schleife wiederholt man einen Anweisungsblock. **while** Schleifen eignen sich für Durchläufe, deren Anzahl noch nicht bekannt ist.

### Kopfgesteuerte Schleife

JS



```
while (Bedingung) {Anweisung; Anweisung; usw.}
```

Für die Bedingungen gelten die selben Regeln wie bei einer IF-ELSE Verzweigung. Siehe 7.0 Verzweigung (Vergleichsoperatoren). Die Anweisungen werden solange wiederholt, solange auch die Bedingung erfüllt ist (true).

```
<p>Du musstest <span id="ausgabe"></span> mal raten!</p>
<script>
  var eingabe = "";
  var versuch = 0;
  while (eingabe != "Innsbruck") {
    eingabe = window.prompt("Hauptstadt von Tirol?");
    versuch ++;
    document.getElementById("ausgabe").innerHTML = versuch;
  }
</script>
```



**versuch++**; ist ein Zähler. Dabei wird pro Durchgang eines hinzuaddiert. Es entspricht einem **versuch = versuch + 1**;

### Fussgesteuerte Schleife

JS



```
do {Anweisung; Anweisung; usw.}
while (Bedingung);
```

Der Anweisungsblock wird einmal sicher durchlaufen. Am Ende werden die Bedingungen mit `while ()`; definiert, die ein weiteres durchlaufen bestimmen sollen. Wird die Bedingung erfüllt (true), dann wird der Anweisungsblock nochmals ausgeführt.

```
var eingabe = "";
var zaehler = 0;
do {eingabe = window.prompt("Mathequiz: 4 x 3 = ?");
  zaehler ++;
  if (eingabe == 12) {
    window.alert("Bravo! " + zaehler + " Versuche");
    break;}
}
while (eingabe != 12);
```



Mit **break**; verlässt man sofort eine Schleife!



**!!! ACHTUNG !!!** Wenn die Bedingungen falsch gesetzt sind, kann eine Endlosschleife entstehen. Hier ein Beispiel für eine Endlosschleife:

```
var zaehler = 5;
do {zaehler++; window.alert(zaehler);}
while (zaehler > 5);
```