

Bisher haben wir Strings mit dem + Operator verbunden. Wir haben auch einen String als Array betrachtet und mit der `.length` Methode die Anzahl der Zeichen ermittelt.

Positionen von Text in einem String finden

JS



```
.indexOf("Textteil", start);
```

start ... gibt die Position an, ab welcher die Suche starten soll.
z. B. 4 → also nach dem vierten Zeichen.

Ist der Textteil nicht im String, wird `-1` ausgegeben.

Mit `.lastIndexOf` sucht man nach der letzten Übereinstimmung.

```
var eingabe = "O Genie, der Herr ehre dein Ego!";
var pos = eingabe.indexOf("ehre", 4);
window.alert(pos);
```

Text aus einem String ausschneiden

JS



```
.slice(start, ende);
```

start ... Startposition im Index z. B. 5 → also das fünfte Zeichen.

ende ... Ende im Index z. B. 12 → bis zum 12 Zeichen. Wird kein Wert für das Ende angegeben, dann gibt die `.slice` Methode den Rest des Strings zurück.

Gibt man eine negative Zahl als start Wert an, dann wird vom Ende des Strings die Startposition ermittelt. z. B. `eingabe.slice(-5)`; schneidet die letzten fünf Zeichen ab.

```
var eingabe = "O Genie, der Herr ehre dein Ego!";
var ausgabe = eingabe.slice(23, 31);
window.alert(ausgabe);
```



Die `.substr()` Methode schneidet ebenfalls einen Textteil heraus. Nur wird hier der Startwert und die Anzahl der Zeichen angegeben.

```
var ausgabe = eingabe.substr(23, 8);
```

UTF-16 Code ermitteln

JS



```
.charCodeAt(index);
```

index ... Die Position des Zeichens. Das erste Zeichen hat die Position 0.

Zurück gegeben wird der UTF-16 Code des Zeichens. z. B. `e = 101`, `D = 68`

```
var satz = "Dreh mal am Herd!";
var derUTFcode = satz.charCodeAt(2);
window.alert(derUTFcode);
```



Mit `String.fromCharCode()` dreht man die Sache um.

Man gibt den UTF-16 Code ein und bekommt das Zeichen zurück.

```
var ausgabe = String.fromCharCode(83, 116, 114, 105, 110, 103);
window.alert(ausgabe);
```