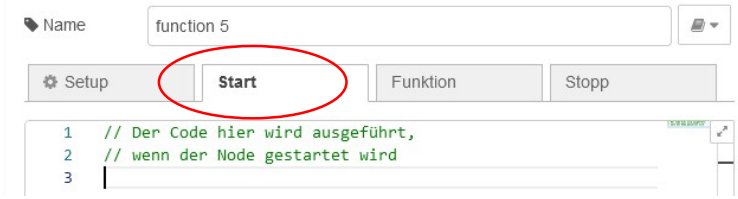
function

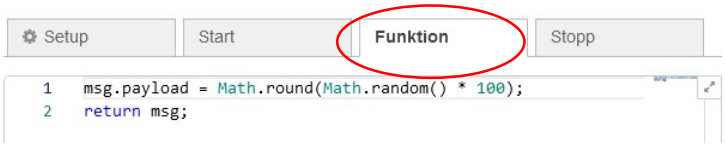
Function-Nodes können beliebigen Java-Script-Code enthalten und sie dienen zur Filterung und Manipulation von Nachrichten, die zwischen den Knoten ausgetauscht werden.

Name gibt den Namen der Node an.

Im Reiter Setup wird die Anzahl der Ausgänge festgelegt.



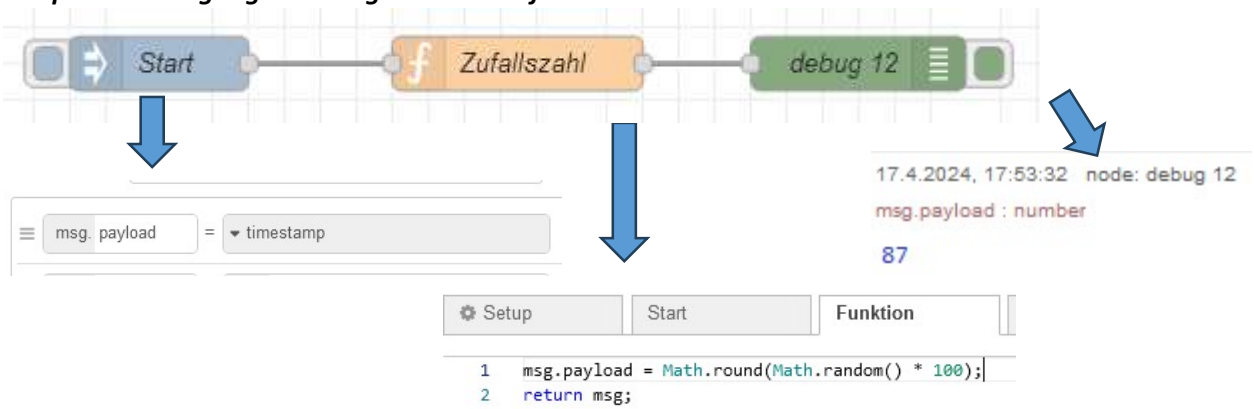
Hier liegt der Code, der bei einem Deployment des Node ausgeführt wird.



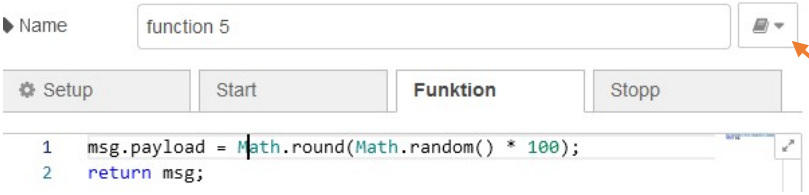
Im Reiter Funktion wird der Java-Script-Code eingefügt.

Hier steht der Code, der bei einem Stopp von Node-RED ausgeführt wird.

Beispiel 1: Erzeugung und Ausgabe einer Zufallszahl



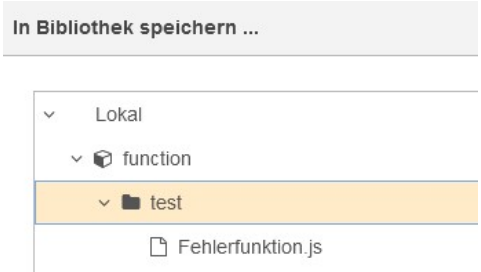
Hinweis: Funktionen speichern



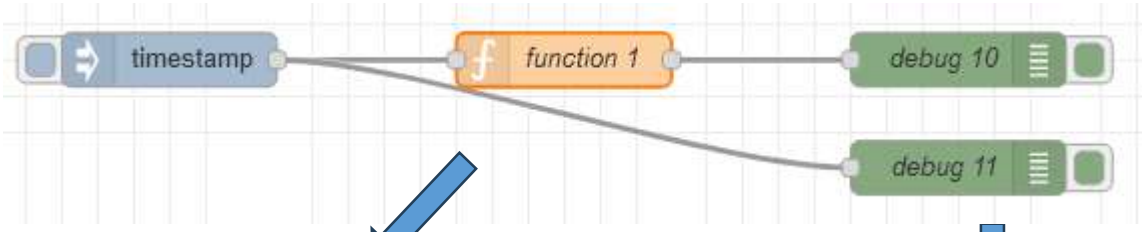
Einmal geschriebene Funktionen können in einer Bibliothek gespeichert werden. Dazu wird das Buch angeklickt und Bibliothek speichern ausgewählt.

und ein Name für die Funktion vergeben werden.
Der Abruf der Funktion erfolgt dann über Bibliothek öffnen…

Im folgenden Fenster kann nun ein Verzeichnis



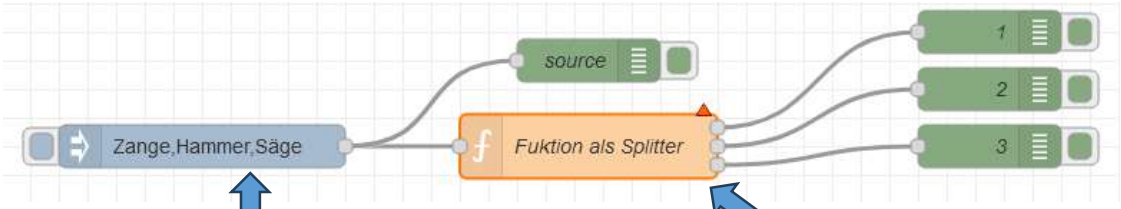
Beispiel 2: Ausgabe von Timestamp (Anzahl der Millisekunden seit dem 01.01.1970) und der umgerechneten Echtzeit.

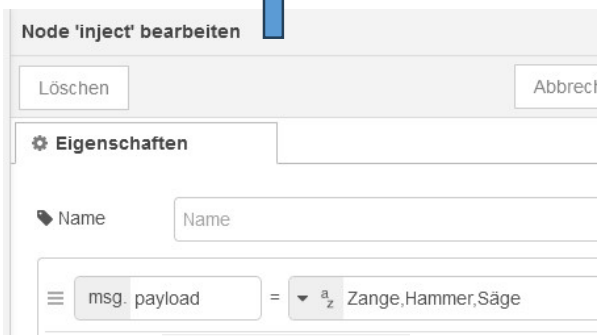


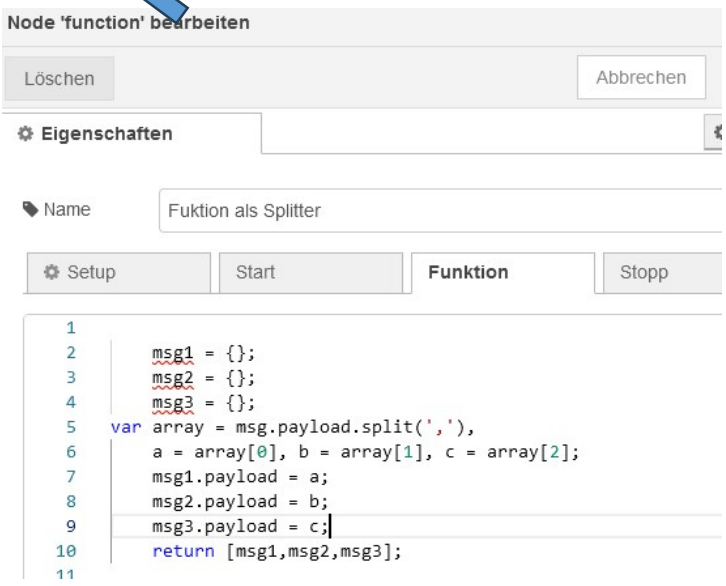

```
const timestamp = new Date(msg.payload);
const year = timestamp.getFullYear();
let month = timestamp.getMonth();
month = (month < 9 ? '0' : '') + (month + 1);
let day = timestamp.getDate();
day = (day < 10 ? '0' : '') + day;

let timestring = year + '-' + month + '-' + day + ' ';
timestring += timestamp.toLocaleTimeString();
msg.payload = timestring;
return msg;
```

```
17.4.2024, 17:45:40   node: debug 11
msg.payload : number
1713368740008
17.4.2024, 17:45:40   node: debug 10
msg.payload : string[19]
"2024-04-17 17:45:40"
```

Beispiel 3: gesplittete Ausgabe einer msg.payload in mehrere msg.payload´s



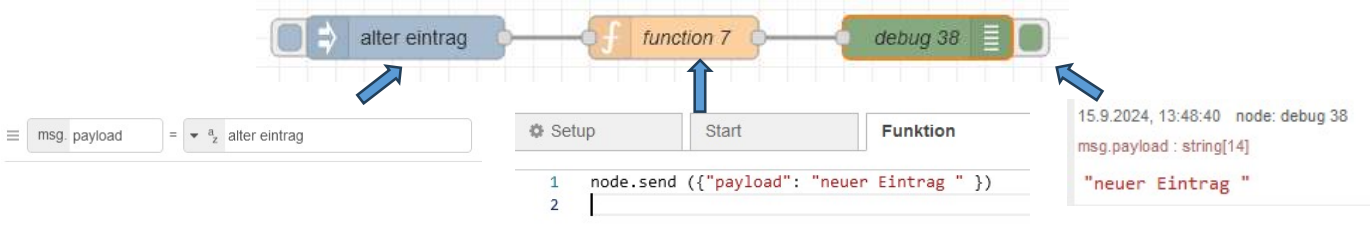




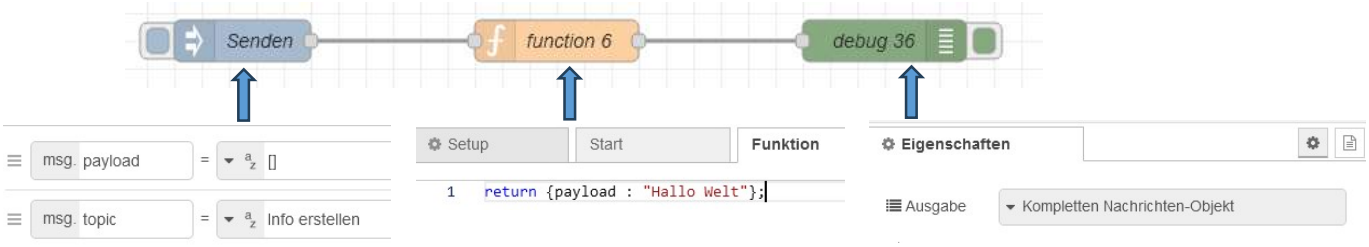
Ausgabe:

```
9.9.2024, 11:14:08   node: source
msg.payload : string[17]
"Zange,Hammer,Säge"
9.9.2024, 11:14:08   node: 1
msg.payload : string[5]
"Zange"
9.9.2024, 11:14:08   node: 2
msg.payload : string[6]
"Hammer"
9.9.2024, 11:14:08   node: 3
msg.payload : string[4]
"Säge"
```

Beispiel 4: andere Ausgabe von msg.payload

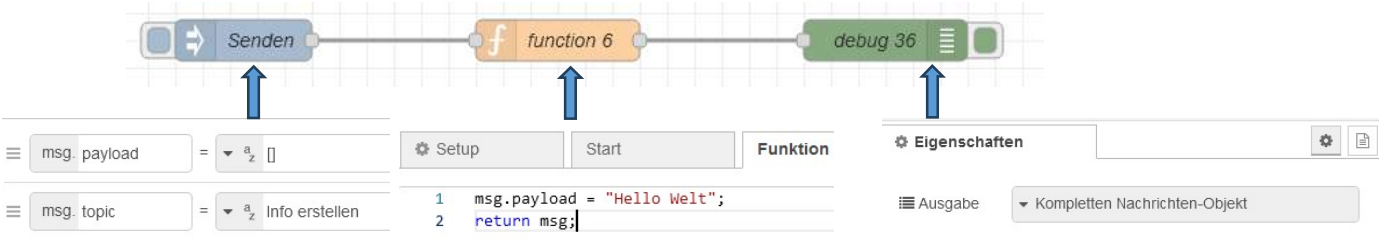


Beispiel 5: Nachrichten erstellen Version 1 (Verlust von msg.topic)



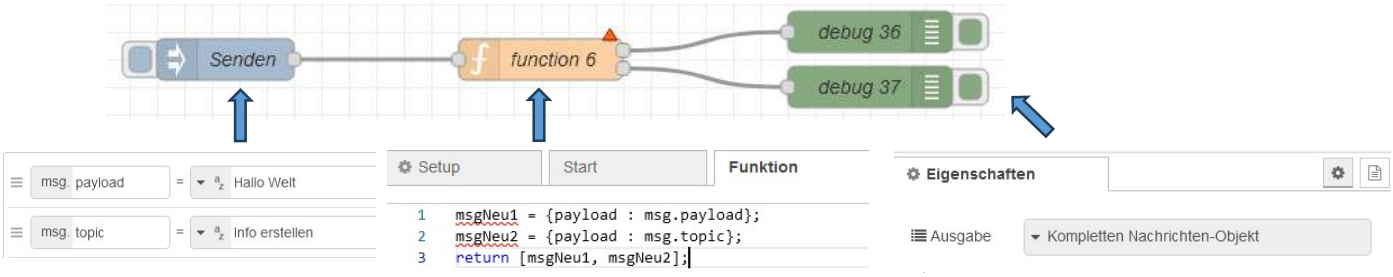
Der msg.payload bekommt einen Platzhalter und wird in der function-Node beschrieben.

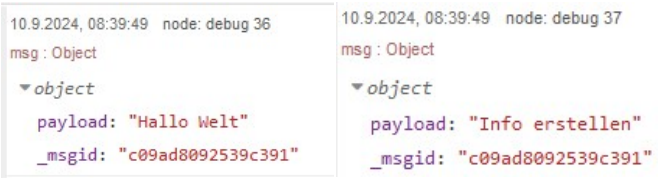
Beispiel 6: Nachrichten erstellen Version 2 (msg.topic wird ebenfalls ausgegeben)



Wie Beispiel 5. Allerdings werden alle weiteren Nachrichten ausgegeben.

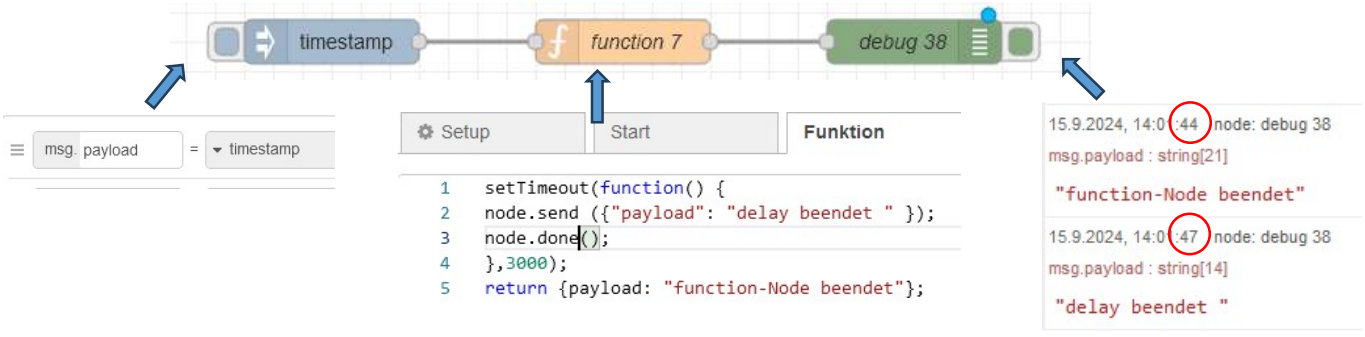
Beispiel 7: eingehende Nachricht auf mehrere Ausgangsports





Die beiden Nachrichten können nun getrennt voneinander weiterbearbeitet werden.

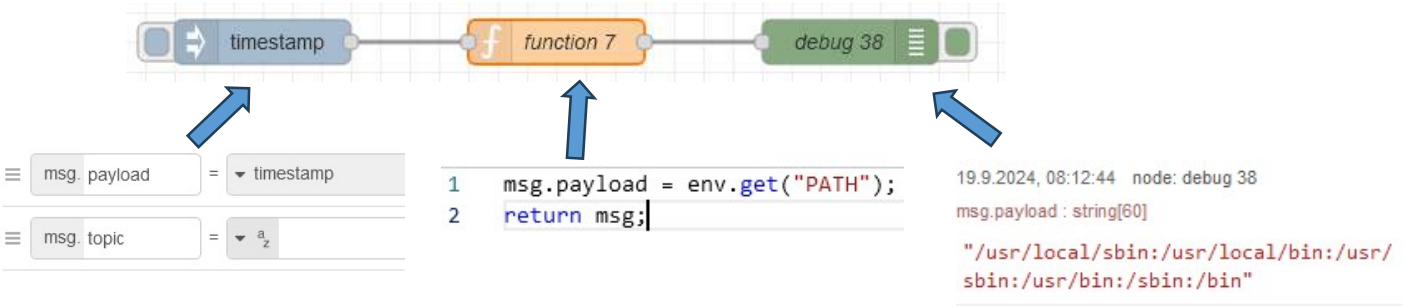
Beispiel 8: Eine Nachricht verzögern



setTimeout() bietet eine einstellbare Startverzögerung (3000 ms = 3 Sekunden).

Beispiel 9: auf Umgebungsvariablen zugreifen

Mit dem Befehl env können aktuelle Umgebungsvariablen anzeigt werden.



Beispiel 10 Abfrage für mehrere Ausgänge