

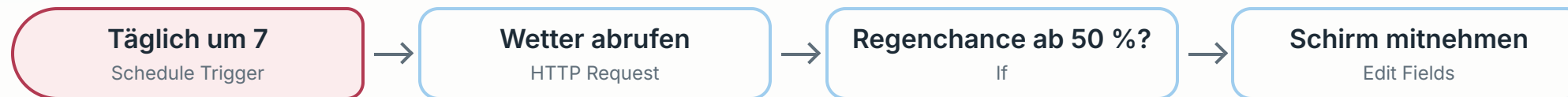
Vom Knopfdruck zur *Uhr*

In Teil 2 hast du deinen ersten Workflow auf Knopfdruck gebaut und getestet. Hier kommen zwei dazu: einer, an dem du Input und Output in Ruhe liest und Werte über den Node-Namen holst, und einer, der jeden Morgen von allein startet.

WORKFLOW 1: MEIN ERSTER FLOW



WORKFLOW 2: AUTOMATISCHER SCHIRMCHECK

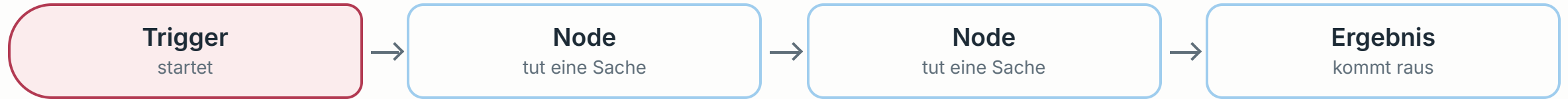


Kurz wiederholt: Wo finde ich was in n8n?

STELLE	WAS DORT IST
Linkes Menü, Overview	alle Workflows, Suche, Neu anlegen
Canvas in der Mitte	die Arbeitsfläche, hier liegen die Nodes
Plus oben rechts oder am Node-Ausgang	Node-Menü mit Suchfeld
Oben links	Workflow-Name, anklicken zum Umbenennen
Oben rechts	Publish, Drei-Punkte-Menü mit Settings und Duplicate, je nach Version oder Einstellung auch Save
Reiter Editor und Executions	bauen oder alte Läufe ansehen
Unten Mitte	Execute workflow

[Lektion 7 im Lernpfad](#) zeigt denselben Bildschirm zum Anklicken.

Ein Workflow ist eine Kette



- **Trigger:** startet die Kette. Immer der erste Node.
- **Node:** tut eine Sache: Daten anlegen, prüfen, verändern, holen, senden.
- **Verbindung:** vom Ausgang rechts zum Eingang links des nächsten Nodes.
- **Execute workflow:** die ganze Kette läuft einmal. **Execute step:** dieser Node, mit den Daten davor. Fehlen sie, laufen die Nodes davor mit.

Merk dir Grün heißt: ist durchgelaufen. Ob das Ergebnis stimmt, siehst du erst, wenn du in den Node schaust.

Kurz wiederholt: der Manual Trigger

- Im Menü: **Add first step**, dann „Trigger manually“.
- Heißt auf dem Canvas „When clicking ‘Execute workflow’“. Umbenennen ist erlaubt, etwa in „Manuell starten“.
- Startet nur, wenn du klickst. Nie von allein.
- Liefert selbst keine Daten. Die kommen vom nächsten Node.

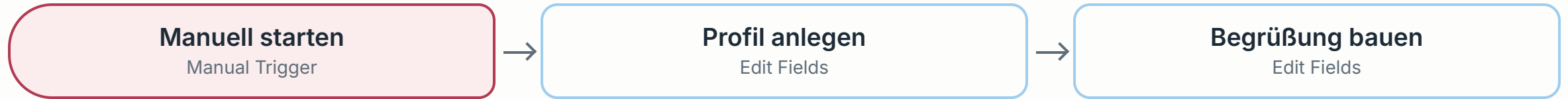
Merk dir Das ist der Trigger zum Lernen und Testen. Später in diesem Teil startet eine Uhr den Workflow.

Kurz wiederholt: Edit Fields

- Im Menü: Plus am Ausgang, Suche „Edit Fields“, dann **Edit Fields (Set)**.
- **Add Field:** Name, Typ, Wert. Für den Anfang reichen String (Text) und Number (Zahl).
- Jedes Feld hat einen Schalter **Fixed | Expression**. Fixed: fester Wert. Expression: wird ausgerechnet.
- **Include Other Input Fields:** an, wenn die eingehenden Felder mitkommen sollen.

Beispiel: `name` = Alex, `thema` = Automatisierung. Nach Execute step steht rechts ein Item mit zwei Feldern.

Zum Nachbauen: Mein erster Flow



- Drei Nodes, kein Internet, keine Zugangsdaten.
- „Begrüßung bauen“ ist ein zweites Edit Fields mit dem Feld `nachricht`, gefüllt per Expression.
- Ergebnis: ein Satz, der den Namen aus dem Node davor enthält, etwa „Hallo Alex!“.

Merk dir Ändere in „Profil anlegen“ den Namen und führ den Workflow noch einmal aus. Der Satz ändert sich mit.

Input und Output lesen

LINKS: INPUT

Was in den Node reinkam: die Daten vom Node davor.

RECHTS: OUTPUT

Was der Node rausgibt: die Daten für den Node danach.

- Oben rechts im Output steht die **Item-Zahl**. In diesem Teil ist sie immer 1.
- Ansichten: **Table** zum Lesen, **JSON** zum Verstehen, **Schema** zum Ziehen.

Achtung Fehlt rechts ein Feld, das links da war? Dann ist **Include Other Input Fields** aus.

Expression mit Node-Namen

`$json` aus Teil 2 liest nur den Node direkt davor. Hier nennst du den Node beim Namen:

```
Hallo {{ $('Profil anlegen').item.json.name }}!
```

TEIL	BEDEUTUNG
<code>{{ und }}</code>	hier wird gerechnet, nicht getippt
<code>\$('Profil anlegen')</code>	hol das Item aus dem Node mit diesem Namen
<code>.item.json.name</code>	das Feld <code>name</code>

- Schalter auf **Expression**, `name` aus dem Input links ins Wertfeld ziehen. Steht dort `$json.name`, ersetze `$json` durch `$('Profil anlegen').item.json`.
- Unter dem Feld zeigt „Result“ den echten Wert. `[undefined]` heißt: Feld falsch geschrieben. Bei falschem Node-Namen meldet n8n einen Fehler.

Schedule Trigger: die Uhr startet den Workflow

- Im Menü: **Add first step**, dann „On a schedule“.
- **Trigger Interval:** Days. **Trigger at Hour:** 7. **Minute:** 0.
- Zeitzone: Drei-Punkte-Menü, Settings, Timezone, Europe/Berlin.
- Execute workflow testet sofort, egal wie spät es ist.
- **Von allein läuft er erst nach Publish.** Ein gespeicherter Workflow tut nichts.

Merk dir Beim Start liegen noch keine Daten vor. Die holt der nächste Node.

HTTP Request: Wetter aus dem Internet holen

FELD	WERT
Method	GET
URL	<code>https://api.open-meteo.com/v1/forecast</code>
Send Query Parameters	an
latitude	52.52
longitude	13.41
daily	<code>precipitation_probability_max,temperature_2m_max</code>
timezone	Europe/Berlin
forecast_days	1

Node umbenennen in „Wetter abrufen“: Unter diesem Namen holen If und Ergebnis-Nodes die Werte. Antwort: ein Item mit Listen unter `daily`. Keine Zugangsdaten nötig.

If: eine Regel, zwei Wege

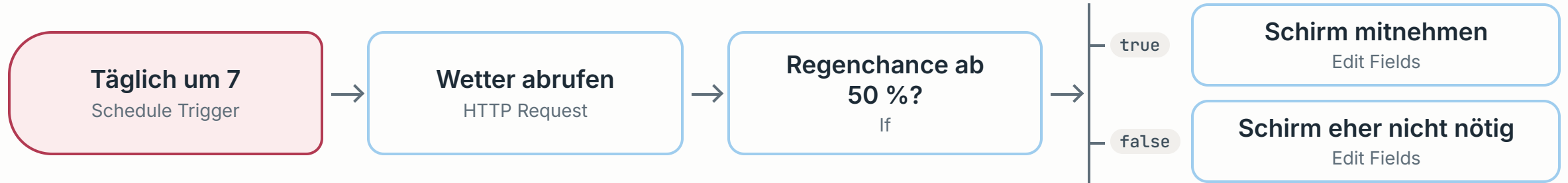
- Im Menü: Plus, Suche „If“. Linker Wert:

```
{{ $('Wetter abrufen').item.json.daily.precipitation_probability_max[0] }}
```

- In der Mitte: **Number, is greater than or equal to**. Rechts: 50.
- **true** oben: Regen chance ab 50 Prozent. **false** unten: darunter.
- An jeden Ausgang kommt ein Edit Fields mit dem Feld `empfehlung`.

Merk dir `[0]` heißt: der erste Eintrag der Liste. Gezählt wird ab null.

Zum Nachbauen: Automatischer Schirmcheck



- Fünf Nodes, eine echte Automation.
- Beide Wege testen: die Schwelle im If kurz auf 0 stellen (true), dann auf 101 (false), danach zurück auf 50.
- Am Ende steht ein Item mit Empfehlung, Ort, Regenchance, Höchsttemperatur und Datum.

Die zwei Ergebnis-Nodes ausfüllen

- `empfehlung`, String: oben „Schirm mitnehmen“, unten „Schirm eher nicht nötig“
- `ort`, String: Berlin, fest eingetippt
- `regen_prozent`, Number: `daily.precipitation_probability_max[0]`
- `temperatur_max`, Number: `daily.temperature_2m_max[0]`
- `datum`, String: `daily.time[0]`

Hinter dem If holst du die Werte mit dem Node-Namen, wie im If selbst:

```
{{ $('Wetter abrufen').item.json.daily.time[0] }}
```

Testen, speichern, aufräumen

WANN

WAS

nach jedem Node	Execute step, Output prüfen
nach jeder Änderung	speichern: neuere Versionen tun es selbst, sonst Save, Cmd+S oder Strg+S
bevor du aufhörst	der Name sagt, was der Workflow tut: „Schirmcheck Berlin“ statt „My workflow“
bei jedem Workflow	eine Sticky Note oben links: Was tut er, wie teste ich ihn?
bei jedem Node	umbenennen: „Wetter abrufen“ statt „HTTP Request“

Merk dir Publish nur, wenn der Zeitplan wirklich laufen soll. Zum Üben nicht.

Wenn es beim Bauen hakt

SYMPTOM	URSACHE	WAS TUN
[undefined] in der Vorschau	Feldname falsch geschrieben	aus dem Input ziehen statt tippen
Fehler „Referenced node doesn't exist“	Node-Name in \$('...') falsch geschrieben oder Node noch nicht umbenannt	Node umbenennen oder Namen genau wie auf dem Canvas schreiben
roter Node	die Fehlermeldung steht im Node	Meldung lesen, Input und Einstellung prüfen
Feld fehlt im Output	Include Other Input Fields aus	Schalter prüfen
links steht kein Input	Node nicht verbunden, oder das Item lief über den anderen Ausgang des If	Verbindung prüfen, zum Testen die Schwelle ändern
If rot: „Wrong type“	die Zahl kommt als Text an, etwa "51" statt 51	Typ prüfen, mehr in Lektion 5

DIE REGEL FÜR DIE FEHLERSUCHE

Such den ersten Node von links, dessen Output nicht stimmt. Nicht den letzten roten.

Der rote Node ist meistens nur die Folge. Das Fehlerlabor trainiert genau diesen Blick.

Dein Auftrag

- 1 **Mein erster Flow:** ein eigener Workflow mit drei Nodes wie oben. Läuft er, kommen zwei eigene Felder dazu, eines davon in der Nachricht per Expression.
- 2 **Schirmcheck nachbauen:** ein zweiter Workflow mit fünf Nodes. Beide Wege mit den Schwellen 0 und 101 testen.
- 3 **Eigene Variante:** anderer Ort, andere Regel. Schreib vorher drei Testfälle auf, mit dem, was rauskommen soll.

Koordinaten zum Ausprobieren: Hamburg 53.55 und 9.99, Köln 50.94 und 6.96, München 48.14 und 11.58.

Merk dir Eine Regel als Beispiel: Jacke mitnehmen, wenn die Höchsttemperatur unter 12 Grad liegt.

Was bleibt hängen?

1 Was macht Execute step anders als Execute workflow?

Execute step führt den geöffneten Node aus, bei Bedarf auch die Nodes davor. Execute workflow lässt die ganze Kette laufen.

2 Woher weiß „Begrüßung bauen“ den Namen?

Aus „Profil anlegen“, über `$( 'Profil anlegen' ). item.json.name . $json.name` geht hier auch, weil der Node direkt davor liegt.

3 Warum startet der Schirmcheck nicht von allein, obwohl er gespeichert ist?

Ein Zeitplan läuft erst nach Publish. Speichern allein reicht nicht.

4 Wo schaust du zuerst, wenn im Ergebnis ein Feld fehlt?

Im ersten Node, in dessen Output es fehlt. Meist ist dort Include Other Input Fields aus.

Weiter im Kurs

LERNPFAD

Lektionen zum Nachklicken

- 5 · Mapping und Datentypen
- 6 · Fehler finden
- 7 · Das n8n-Menü
- 8 · Nodes ausfüllen
- 9 · Erste eigene Flows

EXPRESSION-BAUKASTEN

Kurze Lektion: Expressions in fünf Schritten

WISSENSBOX

Die wichtigsten Dinge, in Bewegung

ÜBUNG 1

Fehlerlabor: grün heißt nicht richtig

NÄCHSTER TEIL

Teil 4: Daten von außen

Erik



Du willst mehr als Selbstlernen? Dann lernst du n8n mit mir 1:1. Auf Anfrage bringe ich es auch Unternehmerinnen, Unternehmern und ganzen Teams bei. Erzähl mir im Erstgespräch, was du vorhast.

15 Minuten mit Erik >