FUN·C·I für Lehrkräfte – Klassenraum und Aufgaben-Editor

Inhalt

1. Teil A – Klassenraum in FUN·C·I . . . 1
   1.1. Klassenraum erstellen oder öffnen . . . 1
   1.2. Die Schüler-Übersicht . . . 2
2. Teil B – Eigene Aufgaben mit dem Aufgaben-Editor . . . 4
   2.1. Überblick . . . 4
   2.2. Eine Aufgabe bearbeiten . . . 5
   2.3. Musterlösung als echte Blockly-Werkstatt bauen . . . 5
   2.4. Prüf-Funktion schreiben und testen . . . 6
   2.5. Aufgabensatz-Bericht . . . 6
   2.6. PRIMM-Zyklen erstellen (optional) . . . 7
   2.7. Veröffentlichen . . . 7
3. Tipps . . . 7

Diese Anleitung zeigt die beiden Bereiche für Lehrkräfte: den **Klassenraum** in `funci.html` (Fortschritt der Klasse verfolgen) und den **Aufgaben-Editor** (`aufgaben_editor.html`), mit dem sich eigene Aufgaben ohne Programmierkenntnisse erstellen lassen. Wie die Schüler:innen arbeiten, zeigt die Schüler-Anleitung (`FUNCI_Anleitung_Schueler.pdf`).

1. Teil A – Klassenraum in FUN·C·I

1.1. Klassenraum erstellen oder öffnen

Ein Klick auf **Klassenraum** oben rechts in `funci.html` öffnet das Klassenraum-Fenster mit vier Reitern: **Als Schüler** und **Schülercode** (für Schüler:innen), **Lehrer** sowie **Moodle-Anbindung** (für die Schul-IT, siehe `FUNCI_Anleitung_Admin_Moodle.pdf`).

Im Reiter **Lehrer**:

- **Neuen Raum erstellen** – optional ein Ablaufdatum (Standard: ein Schuljahr) und über **Auswählen** die Aufgabensätze, die die Klasse sehen soll. Danach erscheinen der **Raumcode** (für die Schüler:innen), der **Lehrercode** und das Ablaufdatum.
- **Vorhandenen Raum verwalten** – Raumcode und Lehrercode eingeben, dann **Schüler-Übersicht anzeigen**.

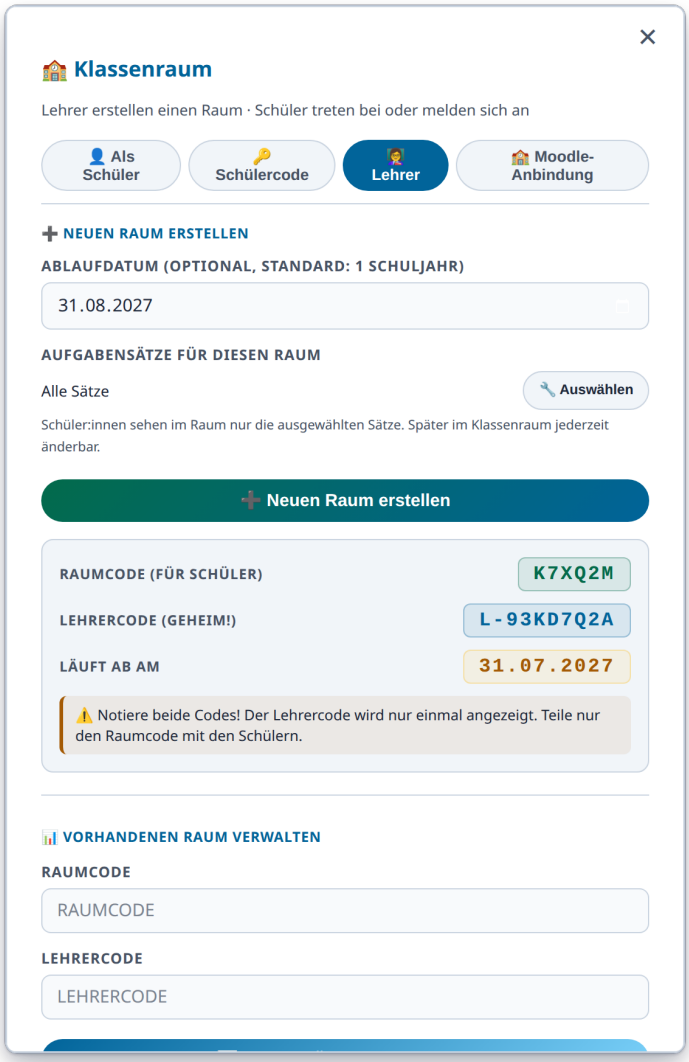


Abbildung 1. Reiter „Lehrer" nach dem Anlegen eines Raums



Raum- und Lehrercode nach dem Anlegen sicher notieren – der Lehrercode wird aus Sicherheitsgründen nur **einmal** angezeigt. Den Raumcode geben Sie an die Klasse weiter.



Hat Ihre Schul-IT FUN·C·I über Moodle eingebunden, entfällt dieser Schritt: FUN·C·I wird dann als Moodle-Aktivität geöffnet, beim ersten Start entsteht automatisch ein Klassenraum, und jeder weitere Start findet ihn wieder.

Im Reiter **Lehrer** führt der Link **Erste Schritte für Lehrkräfte** zu einer Übersichtsseite mit allen Aufgabensätzen, direkten Links und QR-Codes zum Ausdrucken – praktisch, um eine Klasse gleich auf den richtigen Satz zu schicken.

1.2. Die Schüler-Übersicht

Nach dem Öffnen eines Raums sehen Sie den Fortschritt Ihrer Klasse.

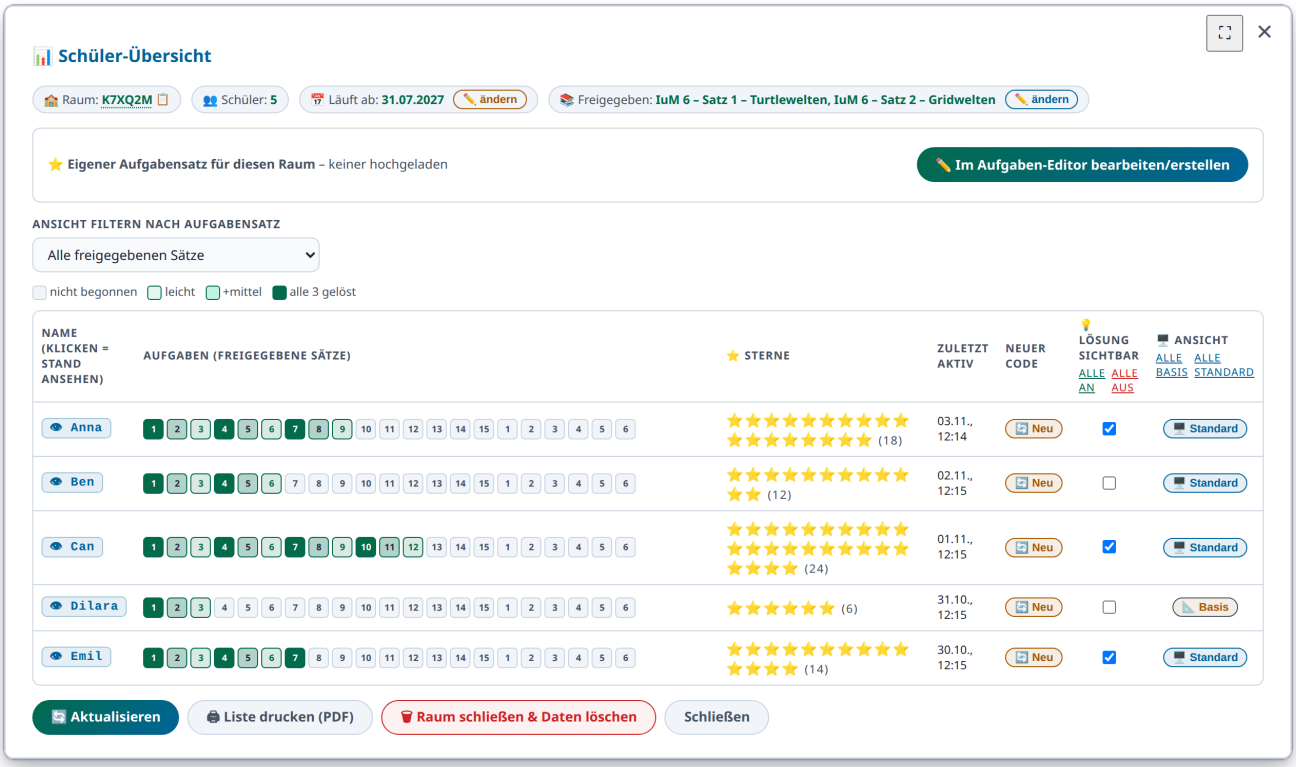


Abbildung 2. Schüler-Übersicht eines Raums mit fünf Schüler:innen

Oben stehen Raumcode, Anzahl der Schüler:innen, das Ablaufdatum und die freigegebenen Aufgabensätze – beides lässt sich über **ändern** anpassen. Darunter können Sie einen eigenen Aufgabensatz für diesen Raum im Aufgaben-Editor erstellen oder bearbeiten (siehe Teil B) und die Tabelle auf einen Satz filtern.

In der Tabelle:

- **Aufgaben** – ein Kästchen je Aufgabe mit ihrer Nummer im Satz. Die Farbe zeigt, wie viele Stufen gelöst sind (siehe Legende über der Tabelle). Sind es viele Aufgaben, sind die Kästchen auf zwei Zeilen eingeklappt.
- **Name anklicken** – öffnet den Stand dieser Schülerin oder dieses Schülers zur Ansicht. Bearbeiten Sie dort selbst etwas, wird das als „von Lehrkraft bearbeitet" markiert.
- **Neuer Code** – erzeugt einen neuen Schülercode (zum Beispiel, wenn er verloren ging); der Fortschritt bleibt erhalten.
- **Lösung sichtbar** – legt je Person (oder über **Alle an** / **Alle aus**) fest, ob der Knopf „Lösung" angeboten wird.
- **Ansicht** – **Standard** oder vereinfachte **Basis**-Ansicht, je Person oder für alle. Neue Schüler:innen starten in der Basisansicht. Sie zeigt nur das Nötigste: unten **Ausführen** und **Tipp** (Stop und Geschwindigkeit nur während eines Laufs), **Lösung** und Schriftgröße im Menü der Aufgabe, im Kopf nur die Sterne; ohne Schrittweise, Code und Visualisierung. Nach dem ersten Baustein leuchtet **Ausführen** kurz auf, nach einem Fehlversuch **Tipp**.

Unten: **Aktualisieren**, **Liste drucken (PDF)** und **Raum schließen & Daten löschen**.

2. Teil B – Eigene Aufgaben mit dem Aufgaben-Editor

2.1. Überblick

`aufgaben_editor.html` öffnet zunächst leer. Über **Importieren** laden Sie eine bestehende `aufgaben.json` mit `aufgaben_funktionen.js` (zum Beispiel den Standardsatz als Vorlage) oder beginnen mit **Neue Aufgabe**.

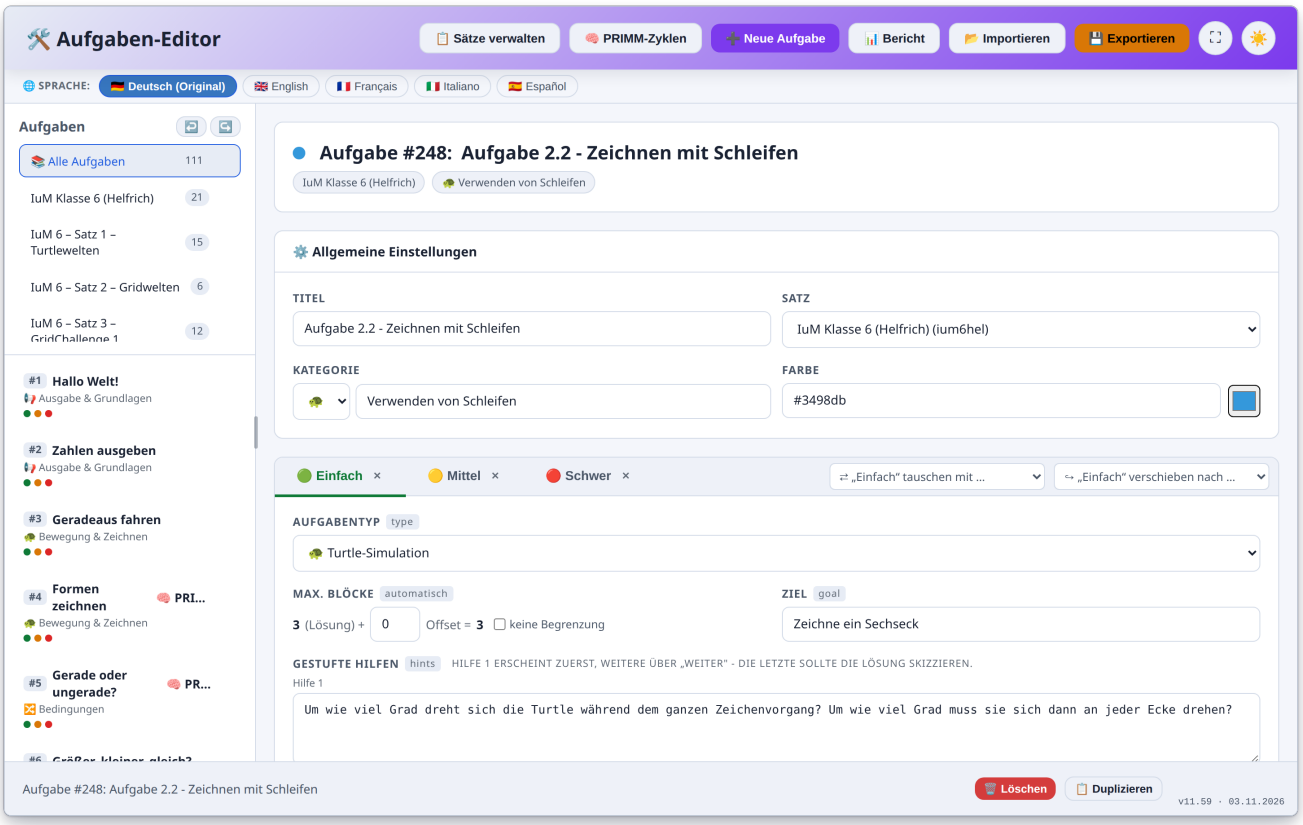


Abbildung 3. Aufgaben-Editor nach dem Laden des Standardsatzes

- **Oben:** Sätze verwalten, PRIMM-Zyklen, Neue Aufgabe, Bericht, Importieren, Exportieren; darunter die **Sprache** – in den anderen Sprachen übersetzen Sie nur die Texte, alles andere bleibt wie im deutschen Original.
- **Links:** die Aufgabensätze und Aufgaben, oben mit Rückgängig und Wiederholen. Die Breite der Liste ändern Sie mit der Leiste an ihrem rechten Rand (Ziehen, Pfeiltasten, Doppelklick = Standardbreite).
- Änderungen werden automatisch im Browser gesichert.

Sind schon Aufgaben geladen, fragt das Import-Fenster, was mit ihnen geschehen soll: **Ersetzen** (der Import tritt an ihre Stelle) oder **Ergänzen** (der importierte Satz kommt dazu; doppelte Aufgabennummern werden automatisch neu vergeben, ein gleichnamiger Satz bekommt den Zusatz „(2)").

2.2. Eine Aufgabe bearbeiten

Ein Klick auf eine Aufgabe öffnet das Formular: Titel, Satz, Kategorie und Farbe, darunter je Stufe (**Einfach**, **Mittel**, **Schwer**) Aufgabentyp (Turtle, Grid-Welt, Nur-Text oder Spielwelt), Block-Grenze, Ziel und gestufte Hilfen. Stufen lassen sich tauschen oder in eine andere Aufgabe verschieben.

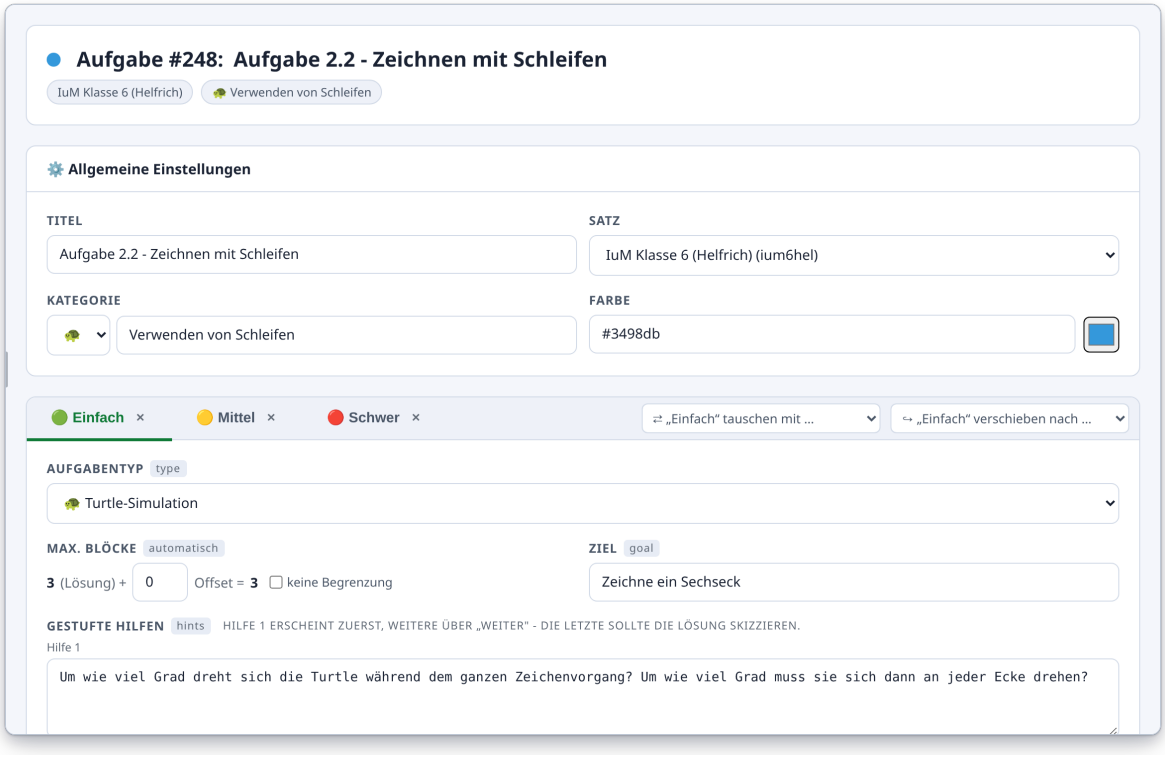


Abbildung 4. Formular einer Aufgabe

Weiter unten folgen Aufgabentext, der Werkzeugkasten (welche Blöcke die Schüler:innen bekommen) und bei Grid-Welten der Welt-Editor mit Testwelten.

2.3. Musterlösung als echte Blockly-Werkstatt bauen

Die Musterlösung bauen Sie per Ziehen und Ablegen in einer echten Blockly-Werkstatt – mit denselben Blöcken, die auch die Schüler:innen haben. Mit **Ausführen** sehen Sie sie direkt laufen.

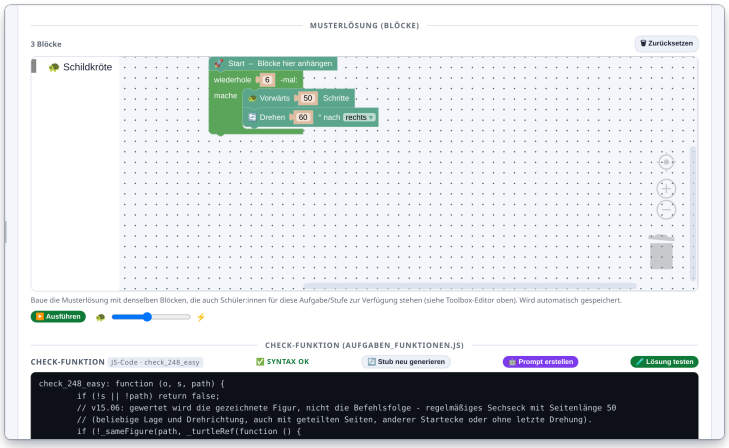


Abbildung 5. Musterlösung und Prüf-Funktion

2.4. Prüf-Funktion schreiben und testen

Darunter steht die **Check-Funktion**: Sie entscheidet, ob eine Schülerlösung als richtig gilt, und wird beim Tippen auf Syntaxfehler geprüft. **Prompt erstellen** erzeugt einen ausführlichen Auftrag für eine KI, falls Sie sich beim Formulieren helfen lassen möchten.

Im Bereich **Rückmeldungen** können Sie, statt Code zu schreiben, Prüfregeln zusammenklicken, eigene Rückmeldetexte für typische Fehler hinterlegen und Fehlversuche speichern, die die Prüfung **nicht** bestehen dürfen.

Der wichtigste Knopf ist **Lösung testen**: Er führt die Musterlösung gegen die Check-Funktion aus und zeigt das Ergebnis mit den Werten, die die Prüfung gesehen hat.



Abbildung 6. Testergebnis der Musterlösung

2.5. Aufgabensatz-Bericht

Über **Bericht** sehen Sie auf einen Blick, ob alle Stufen eine Musterlösung und eine eigene Check-Funktion haben und ob noch etwas fehlt – auch in den PRIMM-Zyklen.



Abbildung 7. Aufgabensatz-Bericht ohne offene Punkte

2.6. PRIMM-Zyklen erstellen (optional)

Über **PRIMM-Zyklen** legen Sie zu einer Aufgabe einen Predict–Run–Investigate–Modify–Make-Zyklus an: verknüpfte Aufgabe, Titel und Zyklus-Typ (Grid-Welt, Turtle oder Nur-Text), dann je Phase die Inhalte.

Unter **Zyklus endet nach** legen Sie fest, nach welcher Phase der Zyklus mit „Fertig" endet – etwa nur bis Investigate, ohne Modify und Make. Die Inhalte späterer Phasen bleiben gespeichert, falls Sie sie wieder zuschalten.

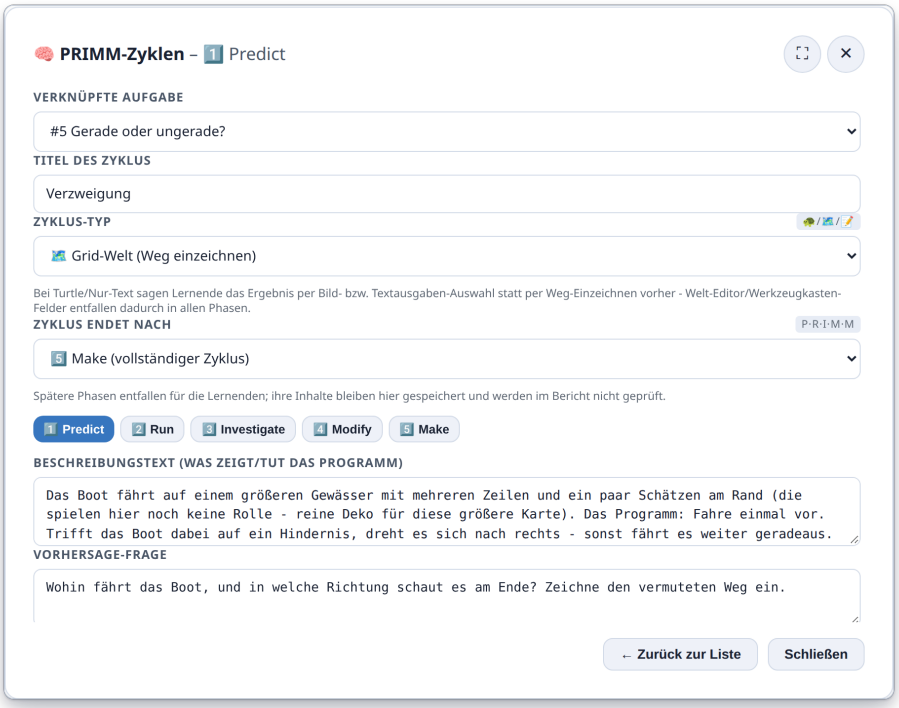


Abbildung 8. Einen PRIMM-Zyklus bearbeiten

2.7. Veröffentlichen

Zwei Wege machen einen fertigen Satz nutzbar:

- **Direktübernahme** – Öffnen Sie den Editor aus der Schüler-Übersicht (**Im Aufgaben-Editor bearbeiten/erstellen**), genügt ein Klick, und der Satz steht im Raum bereit.
- **Export** – `aufgaben.json` und `aufgaben_funktionen.js` (auch als ZIP) herunterladen und später importieren oder weitergeben.

3. Tipps

- Die mitgelieferten Standardsätze sind gute Vorlagen für Turtle-, Grid- und PRIMM-Aufgaben (über **Importieren**).
- Vor dem Veröffentlichen immer den **Bericht** ansehen.
- Zur Passung zum Bildungsplan siehe `FUNCI_Funktionsueberblick_und_Vergleich.pdf`.