

Hinzufügen, was fehlt, ohne anzutasten, was da ist

Kit de documentation

Dokumentation — Das Kit erweitern — DE

1 Gegenstand

Dieses Dokument ist eine sinngemäße Übersetzung. Maßgeblich ist das Originaldokument auf Französisch; Erweiterungen und Änderungen werden stets dort eingepflegt.

Das Kit deckt eine endliche Menge an Darstellungen und Dokumentarten ab. Früher oder später trifft ein Projekt auf einen Bedarf, der nicht darin vorkommt: eine fehlende Gestaltung, ein besonderer Blocktyp, eine Regel aus seinem eigenen Fachgebiet.

Zwei Wege stehen offen. Die Dateien des Kits zu ändern deckt den Bedarf sofort und zwingt bei der ersten Aktualisierung zur Wahl zwischen der Anpassung und der neuen Fassung. Eigene Dateien danebenzulegen kostet anfangs etwas mehr Arbeit und lässt beides nebeneinander bestehen.

Dieses Dokument beschreibt den zweiten Weg: wohin Ihre Dateien gehören, wie sie benannt werden, wie ein Generator sie mit denen des Kits verbindet, und was bricht, wenn das Kit sich weiterentwickelt.

2 Was dem Kit gehört und was Ihnen gehört

Die Grenze liegt bei einem Präfix. Jede Datei, deren Name mit “Kit -” oder “kit_” beginnt, stammt aus dem Kit und wird bei der nächsten Aktualisierung ersetzt. Ihre Dateien tragen eines von zwei anderen Präfixen, je nach Reichweite.

- **Kit-X —, für Ihre Erweiterungen des Kits:** ein zusätzliches *Stylesheet*, eine Farbpalette, ein eigenes Werkzeug. Diese Dateien gelten für alle Ihre Projekte und folgen Ihnen von einem zum anderen, ohne umbenannt zu werden. Das X steht für Erweiterung.
- **[Präfix] —, für das, was nur einem Projekt gilt:** seine Dokumente, sein *Registry*, seine *Datenmodule*, seine ausführbaren Artefakte. Die vollständige Liste steht in Gemeinsame Struktur §2.

Hinweis: Ihre Kit-X-Dateien bestehen nur in einem einzigen Exemplar. Sichern Sie sie regelmäßig.

- **Dateien des Kits, in einem Zug ersetzt:** die vier Stylesheets und ihre Reference, die ausführbaren Artefakte `kit_*.py` und `kit_*.js`, Kit - *Registry.js* und die Dokumente des Kits selbst. Das Vorgehen beim Ersetzen steht im Aktualisierungsleitfaden.
- **Dateien des Projekts, nie angetastet:** Ihr *Registry*, Ihr *Prompt*, Ihre Dokumente, Ihre *Datenmodule*, Ihre ausführbaren Artefakte — Gemeinsame Struktur §12 — und die Stylesheets, die dieses Dokument Sie zu schreiben lehrt.

Hinweis: Eine an Ort und Stelle geänderte Datei des Kits verliert ihre Änderung bei der nächsten Aktualisierung, ohne Warnung: das Ersetzen ist ein Überschreiben. Deshalb führt die Erweiterung über das Hinzufügen.

3 Die Farben ersetzen

Die Farben der Kit-Stylesheets sind Vorgaben, und drei Ebenen überlagern sich. Kit-X - Colors.js trägt die Farben, die Ihnen von Projekt zu Projekt folgen. [Präfix] - Colors.js trägt die, die nur für ein Projekt gelten, und geht den vorigen vor. Jede Datei erklärt nur die Schlüssel, die sie ändert; die übrigen behalten den Wert der Ebene darüber.

Die Projektdatei wird vom Flag `requires.colors` ihres *Registry* gesteuert: setzen Sie es auf `true`, sonst lädt der Generator sie nicht. Die von Kit-X wird von keinem Flag gesteuert — sie ist vorhanden oder nicht. Jede Datei exportiert `colorEntries`, ein Objekt mit zwei Tabellen: `docx` für die Dokumente, `html` für die Webseiten. Die Trennung erlaubt, den Kontrast auf gedrucktem Papier unabhängig vom Bildschirm einzustellen. Die Schlüsselnamen und ihre Vorgaben sind in den Reference der beiden Stylesheets tabelliert, der Vertrag der Datei in Gemeinsame Struktur §6.5.

Das Kit prüft weder Kontrast noch Harmonie: eine unlesbare Farbe ergibt ein gültiges und mühsam zu lesendes Dokument. Ein Wert, der keine Farbe ist, hält dagegen die *Erzeugung* an, mit dem Namen des fehlerhaften Schlüssels.

```
// Kit-X - Colors (2026-08-23 - 17h00).js
const colorEntries = {
  docx: { HEADING_COLOR: '8B0000' },
  html: { HEADING_COLOR: '#8B0000', CODE_BG: '#FAFAFA' },
};
module.exports = { colorEntries };
```

Der Generator lädt beide Ebenen und wendet sie der Reihe nach an. Das *Stylesheet* importiert weder das *Registry* noch diese Dateien: der Generator schlägt die Brücke, wie bei der Sprache und beim Glossar.

```
// setColors fusionne : deux appels successifs cascudent.
if (kitX) style.setColors(kitX.docx);
if (couleurs) style.setColors(couleurs.docx);
```

4 Ein eigenes Stylesheet schreiben

Ein eigenes *Stylesheet* exportiert Funktionen, die Blöcke erzeugen, genau wie die des Kits. Es ergänzt das *Stylesheet* General, statt es zu ersetzen: Ihr Generator importiert beide und ruft je nach Block das eine oder das andere auf.

4.1 Benennung und Paar

Der Name folgt der Konvention der Kit-Stylesheets, mit dem Präfix, das der Reichweite der Datei entspricht: “Kit-X - *Stylesheet* - [Thema] - Code.js” für ein *Stylesheet*, das Ihnen von Projekt zu Projekt folgt, “[Präfix] - *Stylesheet* - [Thema] - Code.js” für eines, das nur einem Projekt eigen ist. Kit-X passt in den meisten Fällen: eine einmal nützliche Darstellung ist es meist auch anderswo.

Die `.docx`-Reference trägt dasselbe Präfix, dasselbe Thema und denselben *Zeitstempel* wie ihre.js. Beide werden zusammen geliefert — die *Paar*-Regel, *Prompt* §6, gilt für Ihre Stylesheets wie für die des Kits.

Die Reference hält fest, was der Code allein nicht sagt: die Formatierungswerte, die Rückgabeverträge, die zwingenden Abfolgen. Sie wird bei jeder Änderung der .js neu erzeugt, nie aus dem Code allein rekonstruiert.

4.2 Was Ihr Stylesheet importiert

Die Geometriekonstanten des Kits — Einzüge, Tabellenbreiten, Schriften, Farben — gehören zum *Stylesheet* General und werden dort gelesen. Sie bei sich abzuschreiben erzeugt eine Abweichung, die erst bei der ersten Änderung der Geometrie sichtbar wird.

```
const style = require('./Kit - Stylesheet - General - Code.js');

// La geometrie vient de General, jamais d'une copie locale.
const { INDENT_L1, INDENT_L2, TABLE_WIDTH_L1, FONT, BODY_TEXT_COLOR } = style;
```

Hinweis: Das *YAML-Stylesheet* des Kits bildet die Ausnahme und importiert nichts: es muss allein benutzbar bleiben. Seine Geometriewerte sind daher abgeschrieben und als solche vermerkt. Diese von Hand gepflegte Abhängigkeit hat eine Abweichung auf Ebene 3 mehrere Monate schlummern lassen — der Import ist die Voreinstellung.

4.3 Was Ihr Stylesheet exportiert

Jede Funktion gibt entweder ein Paragraph zurück oder ein Array von Elementen, wie die Funktionen des Kits. Der Rückgabevertrag wird in Ihrer Reference festgehalten: was die Funktion zurückgibt, was ihr vorangehen muss, was ihr folgen muss.

Zwei Regeln des Kits gelten für Ihre Funktionen. Die Benennung ist sprechend, ohne Abkürzung und ohne zu entschlüsselndes Kürzel — *Prompt* §12. Und jede angezeigte Zeichenkette geht durch eine Lokalisierungstabelle, auch wenn Ihr Projekt heute nur eine Sprache spricht — *Prompt* §4.1.

5 Ihr Stylesheet eintragen

Ein Kit-X-*Stylesheet* folgt Ihnen von Projekt zu Projekt; seine Version gehört daher in Kit-X - *Registry.js*, Ihr eigenes Verzeichnis, und nicht in das *Registry* eines bestimmten Projekts. Legen Sie diese Datei an, wenn Sie Ihr erstes *Stylesheet* schreiben.

```
// Kit-X - Registry.js
module.exports = {
  stylesheets: {
    planches: { version: '1.00', ts: '2026-08-23 - 14h00' },
  },
};
```

Ihre Generatoren lesen dort die aktive Version, statt sie fest einzutragen. Drei Verzeichnisse bestehen dann nebeneinander, jedes mit seinem Bereich: das des Kits

für die Kit-Stylesheets, das von Kit-X für Ihre, das des Projekts für seine Dokumente und *Zeitstempel*.

6 Einen Generator schreiben, der beide verbindet

Ein Generator importiert das *Stylesheet* des Kits und Ihres, dann ruft er je nach zu erzeugendem Block das eine oder das andere auf. Die Reihenfolge der Aufrufe bestimmt die Reihenfolge des Dokuments, wie bei jedem Generator.

```
const style    = require('./Kit - Stylesheet - General - Code.js');
const planches = require('./Kit-X - Stylesheet - Planches - Code.js');
const registry = require('./Monprojet - Registry.js');

style.setLanguage(registry.project.docLanguage);
planches.setLanguage(registry.project.docLanguage);

const corps = [
  style.h1('1', 'Montage'),
  style.paragraph('Le montage se fait en trois passes.'),
  planches.schema(image, 'Vue eclatee'), // votre fonction a vous
  style.releaseParagraph(),
];
```

Die Abschnittsebene hält das *Stylesheet* General, gesetzt von h1, h2 und h3 und gelesen von den übrigen Funktionen. Ein eigenes *Stylesheet*, das die aktuelle Ebene braucht, liest sie über `style.currentLevel()` und schreibt sie nie: nur drei Funktionen setzen sie, und eine vierte hinzuzufügen öffnet wieder den Zuordnungsfehler, den diese Einrichtung schließt.

7 Was die Prüfung abdeckt

Die Prüfwerkzeuge des Kits gelten für Ihre Dokumente ohne Anpassung. Sie lesen die erzeugte Datei, nicht den Code, der sie erzeugt hat.

- **kit_validate_docx.js:** prüft den Aufbau der erzeugten.docx — Ebenen, Abfolgen, Versionsabdruck. Es erwartet den Abdruck des Stylesheets General, den Ihre Dokumente tragen, da dieses sie baut.
- **kit_check_setters.py:** die einzige Ausnahme zum vorstehenden Satz: es liest den Generator, nicht das Dokument. Es prüft, ob die von den *Registry*-Flags verlangten *Setter* am Anfang aufgerufen werden — ein ausgelassener *Setter* schaltet die zugehörige Darstellung stillschweigend ab. Es gilt also für den Generator, den dieses Dokument Sie zu schreiben lehrt.
- **kit_check_markup.py:** vergleicht die Auszeichnung zwischen einer Quelldatei und dem neu erzeugten Dokument. Nützlich, sobald Sie ein bestehendes Dokument wieder aufgreifen.
- **kit_check_ossature.py:** vergleicht das Gerüst zweier Sprachfassungen.
- **kit_extract_map.py und kit_check_fidelite.py:** die Extraktion kennt die Blöcke des Kits. Ein von Ihrem *Stylesheet* erzeugter Block ist ihr unbekannt und kommt beschädigt zurück. Die Treueprüfung sagt es Ihnen — das ist ihre

Aufgabe. Zwei Antworten sind möglich: die Extraktion in Ihrer eigenen Arbeitskopie erweitern, oder den Bedarf an das Kit melden.

Hinweis: *Die Extraktion gehört zum Kit und folgt seinen Aktualisierungen. Eine lokale Erweiterung wird also überschrieben. Ein Blocktyp, der Bestand haben soll, wird eher an das Kit gemeldet als daneben gepflegt — Gemeinsame Struktur §13 beschreibt den Weg.*

8 Was bricht, wenn das Kit sich weiterentwickelt

Das Kit sichert keine Verträglichkeit zu, weder aufwärts noch abwärts. Eine Weiterentwicklung kann eine Funktion umbenennen, eine *Signatur* ändern oder einen Helfer entfernen. Ihr *Stylesheet*, das das des Kits aufruft, trägt die Folgen.

- **Ein Funktionsname ändert sich:** Ihre Aufrufe schlagen beim ersten Start laut fehl. Das ist der bequemste Fall — der Fehler ist sofort da und örtlich bestimmt.
- **Ein Geometriewert ändert sich:** Ihre Blöcke verschieben sich, ohne dass etwas fehlschlägt, sofern Sie den Wert abgeschrieben statt importiert haben. Das ist der Grund für die Regel in §4.2.
- **Ein Abfolgevertrag ändert sich:** ein zwingend gewordener *Nachfolger* ergibt ein fehlerhaft aufgebautes Dokument, das der *Validator* meldet. Lesen Sie die Reference des Stylesheets nach jeder Aktualisierung.

Die Sitzung, in der eine Aktualisierung aufgenommen wird — beschrieben im Aktualisierungsleitfaden — ist der Zeitpunkt, an dem diese Abweichungen behandelt werden. Ihre Stylesheets werden dort ebenso gelesen wie Ihre Generatoren.

9 Modelle außerhalb des Kits schaffen

Ein Projekt, dessen Stoff es verlangt — ein Rezept, ein Feldblatt, ein der Dokumentation fremdes Seitenformat —, darf eigene Modelle außerhalb der Formen des Kits schaffen. Das Recht wird im *Registry* über `allowNonKitTemplates` erklärt, standardmäßig false. Es ist eine schriftliche Erlaubnis, kein Riegel: Keine Prüfung setzt es durch, und ohne es schafft ein Projekt kein *Modell außerhalb des Kits*.

Das Recht betrifft die Formen, nie die Regeln. Das Projekt behält dieselben Reflexe: Quelldatei zuerst, *Registry* als maßgebliche Quelle, frischer *Zeitstempel*, Versionsnummer, die bei jeder Dateiänderung steigt, von Grund auf neu geschriebener Generator, übereinstimmende Couplets, Prüfkette vor jeder Lieferung, kein Inhaltsverlust, keine Formatierung aus dem Gedächtnis, Entscheidungen, die zugleich mit dem Code in den Dokumenten festgehalten werden. Was eine gemeinsame Regel ändern würde, wird an das Kit gemeldet — Gemeinsame Struktur §13 — statt an Ort und Stelle behoben.

Was das Kit nicht mehr gewährleistet: Es steht weder für die Darstellung eines Modells außerhalb des Kits ein noch für dessen Bestand bei einer Weiterentwicklung eines Stylesheets. Es verlangt weiterhin die Reflexe, die Prüfkette und die Identität der *Subsite*.

10 Wann der Bedarf an das Kit zu melden ist

Eine lokale Erweiterung passt zu einem Bedarf aus Ihrem eigenen Fachgebiet. Ein Bedarf, dem andere Projekte ebenfalls begegnen würden, gehört ins Kit, wo er gepflegt und geprüft wird.

- Der Bedarf lässt sich formulieren, ohne Ihr Fachgebiet zu nennen: er ist allgemein.
- Sie haben dieselbe Funktion zweimal für zwei verschiedene Dokumente geschrieben.
- Ihr *Stylesheet* schreibt einen Geometriewert des Kits ab, weil keine Funktion ihn bereitstellt.
- Die Extraktion erkennt Ihren Block nicht und Sie pflegen eine lokale Kopie des Werkzeugs.

La voie de remontée est décrite dans Gemeinsame Struktur §13. Un besoin remonté et accepté devient une fonction du Kit, et votre extension locale disparaît au profit de celle-ci.