Codex – Vászon (📄) & YAML (🛠️) Gyorsútmutató

> **Cél**: Egyetlen összefoglaló arról, hogyan írj **vásznat** (markdown) és a hozzá tartozó **YAML-lépéslistát** úgy, hogy a Codex gond nélkül végrehajtsa.

---

1 | Alapfogalmak

Fogalom	Mit jelent?
Vászon (`/canvases/*.md`)	Ember-barát leírás: mi a feladat, miért kell, mik a lépések, és mikor tekintjük késznek.
YAML (`/codex/goals/*.yaml`)	Gépi-barát recept: pontos fájl-patch, fájl-létrehozás, teszt-futtatás, stb.
Codex	Automatizált build-ügynök: a YAML-t hajtja végre, CI-ben tesztel, és PR-t nyit.

> **Ökölszabály**: *Soha ne írd közvetlenül a kódot*. Csak a két fájlt szerkeszd – a Codex minden mást elintéz.

---

2 | Vászon felépítése

```
# <Sprint / Modul> – <Funkció rövid neve>

## Kontextus
Röviden: *miért* kell ez a módosítás? Mi a kiinduló állapot?

## Cél (Goal)
Egy mondatban: *hova szeretnénk eljutni?*

## Feladatok
- [ ] Pontosan felsorolt, fejlesztői szemmel is értelmezhető lépések.
- [ ] …

## Acceptance Criteria / Done Definition
- [ ] `flutter analyze` hibamentes
- [ ] Teszt-lefedettség ≥ 80 %
- [ ] Új string → ARB fájlban, l10n generálva
- [ ] … (a témához illeszkedő szabályok)

## Hivatkozások
- Canvas → `/codex/goals/<ugyanaz-a-név>.yaml`
- Kapcsolódó issue / design doc / Figma
```

Írási tippek

- Minden vászon **egyetlen feature** vagy refaktor; ne legyen „zsákbamacska”.
- Használj **csekklistát** – a Codex is ezt expectálja.
- Rövid, 80 karakternél nem hosszabb sorokat írj; a diff átláthatóbb.

---

3 | YAML felépítése (példa-séma)

```yaml
# fájl: /codex/goals/<vászon-név>.yaml
meta:
  canvas: <vászon-név>.md     # Kapcsolat a vászonnal
  priority: P1                # P0–P3 a priority_rules.md szerint
steps:
  - patch_file:               # ⇊ egy konkrét módosítás ⇊
      target: lib/.../file.dart
      patch: |
        @@
        - régisor
        + újsor

  - create_file:              # új fájl teljes tartalommal
      target: test/.../file_test.dart
      content: |
        import 'package:flutter_test/flutter_test.dart';
        ...

  - write_test:               # rövidítés, ha csak tesztet adunk
      name: validates_emails_correctly
      location: test/validators/email_validator_test.dart
      content: |
        test('...', () {
          ...
        });

  - run: flutter test         # parancs futtatása (analyze, gen-l10n, stb.)
```

Fontos szabályok

Kötelező	Tiltott
Teszt minden új/érintett logikához	Tiltott útvonal módosítása (`android/` , `ios/` , `pubspec.yaml` , ...)
Hard-coded színek helyett `Theme` / `FlexColorScheme`	"Kitalált" kódrészlet, ami nem létezik a repo-ban
Új szöveg → ARB + `flutter gen-l10n`	Push-based navigáció (`Navigator.push`)

---

4 | Codex által betöltött szabályfájlok

Fájl	Rövid leírás
``	Globális workflow; minden futáskor betöltődik.
``	Engedélyezett / tiltott módosítható utak.
``	Projekt-architektúra, névkonvenciók.
``	P0–P3 beosztás.
``	YAML szintaxis-guide.
``	Pre-commit ellenőrzések.
``	≥ 80 % coverage, emulátor-szabályok.
``	Tilalom a hard-coded színekre.
``	GoRouter-kötelező.
``	i18n menetrend.
``	Megengedett DI-gráf.

(További /docs markdown-ok – pl. `localization_best_practice.md`*,* `golden_and_accessibility_workflow.md` *– témaspecifikus részleteket adnak.)*

---

5 | Leggyakoribb hibák & Gyors ellenőrzőlista

1. **Tiltott fájlhoz nyúltam?** → Nézd meg `.codex/config.yaml`-t.
2. **Van teszt minden logikához?** → `testing_guidelines.md`.
3. **Hard-coded színt írtam?** → `codex_theme_rules.md`.
4. **Új stringet hozzáadtam ARB-hez?** → `localization_logic.md`.
5. **Navigator.push-t használtam?** → `routing_integrity.md`.
6. **Priority szerepel a YAML-ban?** → `priority_rules.md`.

---

6 | Workflow lépésről lépésre

1. **Vászon megírása** a `/canvases` mappában.
2. **YAML létrehozása** ugyanazzal a névvel a `/codex/goals` alatt.
3. `[git add]` → commit → push.
4. **Codex fut** (patch, `flutter pub get`, `flutter analyze`, `flutter test`, golden, coverage).
5. **PR érkezik**: ha minden zöld, csak review + merge.

   Tipp: lefuttathatod lokálisan is a *Codex Dry-Run* parancsot (ld. `codex_dry_run_checklist.md`) mielőtt pusholsz.

Készen is vagy! 💡 A fenti sablonok betartásával elkerülheted a visszadobott futásokat, és gyorsan zöldre hozhatod a buildet.