Jak piszemy strony: formuła v3

Poziom: średnio **Dotyczy: autorzy bazy** **Czas: ~15 min**

Jedna recepta na każdą stronę, żeby cała baza wyglądała i czytała się tak samo. Strona dla tych, którzy dopisują albo poprawiają treść.

Obowiązuje od 27 września 2026 dla wszystkich nowych stron i dla każdej strony, którą przepisujemy. Zastępuje `02-szablon-strony.md` (v2) w części o układzie. Głos autora z `06-styl-konio.md` obowiązuje dalej bez zmian.

Dla kogo piszemy

Czytelnikiem jest **osoba, która pierwszy raz otwiera półkę i nigdy nie pracowała w IT**. Nie zna skrótów, nie wie, co to terminal, nie wie, dlaczego serwer ma dwa dyski. Jeżeli ona zrozumie stronę, doświadczony administrator też ją zrozumie, tylko szybciej przewinie do Fiszki.

Test przed oddaniem strony: czy ktoś, kto zna tylko Worda i przeglądarkę, po przeczytaniu pierwszych trzech zdań wie, **o czym jest strona i po co mu to**?

Trzy rodzaje stron

Rodzaj	Tag	Kiedy	Szkielet
Opowieść	`typ/opowiesc`	pierwsza strona każdej książki	historia narzędzia, po co powstało, jak trafiło do Zielonej, mapa książki
Narzędzie	`typ/objasnienie`	komenda, program, usługa	Fiszka → Z życia wzięte → Jak to działa → Częste błędy → Ćwiczenia
Procedura	`typ/procedura`	doprowadzenie do konkretnego stanu	Zanim zaczniemy → Przebieg → Weryfikacja → Wycofanie zmian → Gdy coś nie zadziała

Dla aplikacji dochodzi zawsze komplet **wdrożeniowy** (patrz niżej).

Szkielet strony narzędziowej

````
# Tytuł strony                                ← bez myślnika, dwukropek zamiast

> **Poziom:** podstawy · **Dotyczy:** ... · **Czas:** ~15 min
>
> Trzy zdania, nie więcej. Co to jest, po co, kiedy tu wracamy.

## Fiszka                                     ← zawsze pierwsza sekcja

### Czynność rzeczownikiem: flaga
```bash
komenda
````

Jedno lub dwa zdania, co robi i kiedy po nią sięgamy.

Przykład:

```
wynik z Zielonej: docker01, jan.nowak, CDN_Zielona
```

Z życia wzięte

> **□□ Z ŻYCIA WZIĘTE**
>
> Piątek, 16:10. Pani Halina dzwoni, że Optima wisi przy zapisie faktury...

Jak to działa ← po 250 słowach zwija się sam
```

Pojęcie: ... ramki pojęć, analogie, schemat

Częste błędy ← tabela: objaw | przyczyna | co robimy

□□ Ćwiczenia

```
## Co robi za nas konwerter

Nie piszemy tego ręcznie. `v3.py` dokłada to przy każdym przebudowaniu strony:

| Element | Skąd | Wygląd |
|---|---|---|
| Chipy metadanych | linia `**Poziom:** · **Dotyczy:** · **Czas:**` | kolorowe pigułki pod tytułem |
| Słowniczek strony | każdy skrót ze `slowniki/skroty.json`, który padł w tekście | zwijany blok „□□Skróty użyte na tej stronie" nad Fiszką |
| Podpowiedź skrótu | pierwsze wystąpienie skrótu w prozie | kropkowane podkreślenie, rozwinięcie po najechaniu myszą |
| Karta fiszki | każde `###` pod `## Fiszka` | ramka z kolorem ryzyka i chipem |
| Rozwinięcie skrótów w fiszce | skróty z nagłówka i opisu, nazwy komend ze `slowniki/komendy.json` | linia „□□df = disk free · SAN = Subject Alternative Name" na dole karty |
| Zwinięcie długiego wyniku | blok wyniku dłuższy niż 25 linii | „Pokaż pełny przykład (N linii)" |
| Zwinięcie teorii | `## Jak to działa` powyżej 250 słów | „□□Rozwiń: jak to działa pod spodem" |

### Kolory kart fiszek

Kolor niesie informację, nie ozdobę. Konwerter rozpoznaje go z treści komendy.

| Kolor | Chip | Kiedy | Przykłady |
```

|---|---|---|---|
□□zielony	TYLKO ODCZYT	nic nie zmienia, można wkleić bez stresu	`df -h`, `SELECT`, `kubectl get`, `Get-Service`
□□pomarańczowy	ZMIENIA SYSTEM	coś tworzy, zmienia albo restartuje	`systemctl restart`, `CREATE`, `docker compose up`, `-out plik`
□□czerwony	OSTROŻNIE · NIEODWRACALNE	usuwa, formatuje, zabija	`rm -rf`, `DROP`, `mkfs`, `kubectl delete`, `--force`
□□niebieski	PLIK KONFIGURACYJNY	blok to plik do wklejenia	`compose.yaml`, `nginx.conf`, `main.cf`

Gdy konwerter się pomyli, dopisujemy w Fiszce pod nagłówkiem komentarz `<!-- ryzyko: zmiana -->` (do wdrożenia w v3.1).

Specjalne bloki w markdownie

Zapis w pliku	Co wychodzi na stronie
`> □□Scenka...`	brązowa karta „Z ŻYCIA WZIĘTE"
`> □□Co ma być na zrzucie...`	przerywana ramka „ZRZUT EKRANU DO UZUPEŁNIENIA"
`> □□Wskazówka...`	zielony callout
`> ⚠□ Uwaga...`	pomarańczowy callout
`![Podpis](@@IMG:klucz@@)`	obrazek z galerii z podpisem

Skróty

Każdy skrót rozwijamy. Zasady:

1. Skrót trafia do `slowniki/skroty.json` z rozwinięciem i jednym zdaniem po polsku, co to jest. Nie tłumaczymy dosłownie, tłumaczymy funkcję: `DNS` to nie „Domain Name System", tylko „książka telefoniczna internetu, zamienia nazwy na adresy IP".
2. Nazwa komendy, która jest skrótem, trafia do `slowniki/komendy.json`: `df` to „disk free", `ps` to „process status".
3. W tekście piszemy skrót normalnie. Resztę robi konwerter.
4. Skrót wieloznaczny (`SAN`, `CSV`, `SID`, `PV`) ma w słowniku oba znaczenia z kontekstem.

Wizualia

Każda strona narzędziowa ma **przynajmniej jeden schemat**, każda Opowieść ma **mapę**. Schematy rysujemy sami (graphviz albo HTML renderowany przez Chromium) w palecie okładek, wgrywamy do galerii i wstawiamy przez `@@IMG:`. Zrzutów z cudzej dokumentacji nie kopiujemy, bo mają własną licencję. Gdzie zrzut ekranu jest potrzebny, stawiamy blok `□□ z dokładnym

opisem, co ma na nim być, i autor dostarcza go po instalacji.

Architektura zawsze ze schematem

Każda strona albo sekcja, która opisuje architekturę (z czego składa się system, co z czym rozmawia, którędy idzie żądanie), ma **co najmniej jeden schemat, wstawiony zaraz po pierwszym akapicie tej sekcji**. Zwykle dwa: jeden „z czego to się składa", drugi „którędy idzie żądanie albo dane". Opis słowami zostaje, schemat go nie zastępuje. Tak samo w Gitea, Nginx, Zabbixie i każdym silniku bazy: czytelnik nie może trafić raz na rysunek, a raz na ścianę tekstu o tym samym rodzaju treści.

Komplet wdrożeniowy aplikacji

Każda aplikacja na półce Aplikacje ma te strony, w tej kolejności:

1. **Opowieść**: skąd się wzięła, kto ją robi, dlaczego Zielona ją wybrała, mapa książki
2. **Instalacja w Dockerze**: pełny `compose.yaml` do skopiowania, ścieżki według konwencji, pierwszy start, weryfikacja
3. **Instalacja bez Dockera**: pakiet albo binarka, usługa systemd, użytkownik systemowy
4. **Pierwsza konfiguracja**: reverse proxy, certyfikat, konto administratora
5. **Integracja z domeną**: LDAP albo SSO, jeżeli aplikacja to umie
6. **Codzienna praca**: to, co robimy w niej najczęściej
7. **Kopia zapasowa i aktualizacja**: procedura
8. **Częste problemy**: objaw, przyczyna, co robimy

Na każdej stronie instalacyjnej stoi notka o konwencji ścieżek:

> □□W naszym środowisku dane aplikacji trzymamy w `/docker/apps/<nazwa>`, a pliki Compose w `/docker/compose/<nazwa>`. To konwencja, nie wymóg Dockera. Można ją zmienić, byle w całej firmie była jedna.

Opowieść na początku książki

Wzorem jest półka „Opowieści historyczne". Pół technicznie, pół narracyjnie:

- **Skąd się wzięło**: kto, kiedy, jaki problem rozwiązywał, jedna ciekawostka
- **Jak to działa w jednym obrazku**: schemat, zero komend
- **Dlaczego Zielona tego używa**: scenka z postaciami, problem sprzed wdrożenia
- **Mapa książki**: tabela: strona, na jakie pytanie odpowiada

Długość: 400 do 900 słów. Czyta się ją jak artykuł, nie jak dokumentację.

```
## Pisownia

- **Bez półpauz i myślników** w nowym tekście, tytułach i podpisach. Zamiast „Docker:
instalacja" piszemy „Docker: instalacja". Tak pisze autor.
- Pierwsza osoba liczby mnogiej, bez trybu rozkazującego (bez zmian względem v2).
- Akapity maksymalnie cztery zdania. Lista zamiast akapitu, gdy wyliczamy więcej niż trzy
rzeczy.
- Tabela zamiast akapitu, gdy porównujemy.

## Wszystko, o co zapytano

Każde pytanie, które padło w naszych rozmowach choć raz, zasługuje na stronę albo sekcję.
Źródłem jest pamięć rozmów w projekcie. Temat z rozmowy przepisujemy do świata Zielonej:
prawdziwe adresy, odciski certyfikatów i nazwy maszyn z pracy autora nie trafiają do bazy.
```

Revision #31
Created 2026-09-27 11:39:03 UTC by Tomasz Konieczny
Updated 2026-10-26 14:18:01 UTC by Tomasz Konieczny