Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską

SZKOLENIE AI · MATERIAŁ DLA UCZESTNIKÓW

Claude w praktyce

Chmura i Desktop, Chat i Code, projekty, modele, skills, plugins, mody, artefakty i narzędzia codziennej pracy

Adam Stecyk
Prowadzący

Listopad 2026
Wersja 1.1

Spis treści

01 Gdzie działa Claude: chmura czy desktop; Chat czy Code

02 Projekty w Chat a folder w Code; wybór modelu

03 CLAUDE.md i SKILL.md: stałe instrukcje i umiejętności

04 Skills, plugins, mody i konektory (MCP)

05 Artefakty: Docs, Slides, Sheets, Design, Data, Whiteboards

06 Przeglądarka, Claude in Chrome, computer use, Research, pamięć

07 Usage: limity i jak je oszczędzać

08 Claude Code: Environment, session rail, side chat, split view, sekcje, panele

09 Claude Code: równoległe sesje, agenty i zadania cykliczne

> Interfejs Claude zmienia się co kilka tygodni. Nazwy przycisków i dostępność funkcji zależą od planu (Free, Pro, Max, Team, Enterprise) i wersji aplikacji. Ten materiał opisuje stan z listopada 2026.

1. Gdzie działa Claude

Ten sam Claude jest dostępny w kilku miejscach. Różnią się tym, do czego ma dostęp: tylko do internetu i Twoich wgranych plików, czy także do Twojego komputera.

Claude w chmurze

claude.ai w przeglądarce
aplikacja mobilna (iOS, Android)

Claude Desktop

aplikacja na Windows i macOS
zakładki: **Chat** i **Code**

	Chmura (claude.ai, mobile)	Claude Desktop
Instalacja	Nic nie instalujesz, logujesz się w przeglądarce	Instalujesz aplikację, logujesz się tym samym kontem
Dostęp do plików	Tylko pliki, które sam wgrasz	Może czytać i zapisywać pliki w folderach, które wskażesz
Twój komputer	Brak	Lokalne konektory MCP, computer use, terminal (w Code)
Historia	Wspólna: rozmowy z Chat widać i w przeglądarce, i w aplikacji	

Desktop: Chat czy Code?

	Chat	Code
Do czego	Rozmowa, pisanie, analiza, pomysły, artefakty	Praca na plikach w folderze: kod, dokumenty, dane, automatyzacja
Kontekst	Rozmowa + Projekt (instrukcje i pliki wiedzy)	Wskazany folder + plik CLAUDE.md
Jak działa	Odpowiada w oknie rozmowy	Agent: sam czyta pliki, uruchamia polecenia, edytuje, sprawdza wynik
Uprawnienia	Nie dotyka Twojego dysku bez konektora	Pyta o zgodę na edycję i polecenia (zależnie od trybu uprawnień)
Jednostka pracy	Rozmowa	Sesja (z własnym folderem, historią i zmianami)
Dla kogo	Każdy	Każdy, kto pracuje na plikach, nie tylko programiści

Zasada praktyczna: jeśli wynik ma trafić do pliku na Twoim dysku lub dotyczy wielu plików naraz, użyj Code. Jeśli chcesz porozmawiać, przemyśleć coś lub szybko wygenerować tekst, użyj Chat.

2. Projekty, folder i modele

Chat i Code inaczej „wiedzą", nad czym pracujesz.

Chat: Projekty

- Projekt grupuje rozmowy na jeden temat.
- **Instrukcje projektu:** stały opis, jak Claude ma pracować (rola, styl, format).
- **Wiedza projektu:** wgrane pliki (PDF, DOCX, notatki), dostępne we wszystkich rozmowach projektu.
- Projekt można udostępnić zespołowi (plany Team i Enterprise).

Code: folder zamiast projektu

- Zakładka Code nie ma (jeszcze) Projektów jak w Chat.
- Przy nowej sesji **wskazujesz folder** roboczy. To on jest „projektem".
- Instrukcje zapisujesz w pliku `CLAUDE.md` w tym folderze.
- Wiedza to pliki w folderze. Claude sam je przeszukuje i czyta.

Odpowiedniki: **Instrukcje projektu** w Chat ≈ **CLAUDE.md** w Code. **Pliki wiedzy** w Chat ≈ **pliki w folderze** w Code.

Wybór modelu

Model wybierasz z listy pod polem wiadomości (w Chat i w Code). Lista zależy od planu.

Model	Kiedy użyć	Koszt limitu
Opus	Trudne zadania: analiza, długie dokumenty, złożony kod, planowanie	Najwyższy
Sonnet	Codzienna praca: dobry balans jakości i szybkości	Średni
Haiku	Proste, szybkie zadania: streszczenia, klasyfikacja, poprawki	Najniższy

Aktualne wersje: Opus 5.5, Sonnet 5.5, Haiku 5.5 oraz Fable 5.2. Nowe wersje pojawiają się regularnie, a lista w aplikacji jest zawsze aktualna.

Ustawienia obok modelu

- **Extended thinking / effort:** model dłużej „myśli” przed odpowiedzią. Lepsza jakość przy trudnych zadaniach, większe zużycie limitu.
- **Fast mode (Code):** szybsze odpowiedzi tego samego modelu.
- **Style (Chat):** zwięzły, formalny, objaśniający lub własny.

3. CLAUDE.md i SKILL.md

Dwa zwykłe pliki tekstowe (Markdown), dzięki którym nie musisz za każdym razem tłumaczyć Claude tego samego.

CLAUDE.md: „pamięć projektu”

- Wczytywany **automatycznie na start każdej sesji** w Code.
- Zawiera: cel projektu, strukturę folderów, zasady, styl, czego unikać.
- Miejsca: `folder/CLAUDE.md` (ten projekt) albo `~/.claude/CLAUDE.md` (wszystkie projekty).
- Polecenie `/init` tworzy pierwszą wersję na podstawie zawartości folderu.

SKILL.md: „umiejętność”

- Opis jednej powtarzalnej czynności, np. „raport miesięczny”, „artykuł w moim stylu”.
- Leży w folderze skilla, obok mogą być szablony i skrypty.
- Claude wczytuje go **tylko wtedy, gdy zadanie pasuje** do opisu (`description`).
- Działa w Chat i w Code.

Przykład CLAUDE.md

```
# Projekt: Szkolenie AI
Materiały szkoleniowe dla zespołu.

## Zasady
- Piszemy po polsku, prostym językiem, bez żargonu.
- Dokumenty zapisuj w folderze /materialy jako PDF i DOCX.
- Nie usuwaj plików z folderu /archiwum.
- Przed większą zmianą przedstaw plan i poczekaj na akceptację.
```

Przykład SKILL.md

```
---
name: raport-miesieczny
description: Tworzy raport miesięczny sprzedaży z pliku Excel.
  Użyj, gdy użytkownik prosi o raport lub podsumowanie miesiąca.
---
# Raport miesięczny
1. Wczytaj plik dane/sprzedaz.xlsx, arkusz z bieżącym miesiącem.
2. Policz sumy per region i zmianę względem poprzedniego miesiąca.
3. Wstaw wyniki do szablonu szablon_raportu.docx.
4. Zapisz jako raporty/RRRR-MM.pdf.
```

> **Różnica w jednym zdaniu:** CLAUDE.md jest zawsze w kontekście (więc ma być krótki), a SKILL.md jest wczytywany na żądanie (więc może być szczegółowy).

4. Skills, plugins, mody i konektory

Cztery sposoby rozszerzania Claude. Łatwo je pomylić, więc warto znać różnicę.

Co to jest	Co daje	Przykład
Skill	Wiedza „jak coś zrobić”: instrukcja + szablony + skrypty	Tworzenie PDF, DOCX, XLSX, PPTX; własny skill „artykuł w moim stylu”
Plugin	Paczka gotowa do instalacji: kilka skills, polecenia `/...`, agenty, hooki i konektory razem	Plugin działu sprzedaży, plugin do przeglądu kodu
Konektor (MCP)	Dostęp do zewnętrznego systemu: czytanie i zapisywanie danych	Gmail, Google Drive, Kalendarz, Notion, Canva, Slack, baza danych, Power BI
Mod	Zmiana wyglądu i działania samej aplikacji Claude	Własny motyw kolorów, panel obok czatu, pasek statusu, polecenie `/raport`

> Skill to **umiejętność**, konektor to **dostęp**, plugin to **pudełko**, w którym jedno i drugie można dostarczyć zespołowi, a mod **zmienia samo okno** Claude.

Mody: Claude „pod siebie”

- **Motyw**: własne kolory i czcionki aplikacji, np. w barwach firmy.
- **Panel**: stałe okienko obok czatu, które się aktualizuje, np. licznik zadań, status raportu, notatki.
- **Pasek statusu**: u dołu okna widać np. model, zużycie limitu i folder roboczy.
- **Polecenie** `/…`: skrót do często powtarzanej prośby, np. `/podsumuj`.
- **Hook**: automatyczna reakcja na działanie Claude, np. powiadomienie po zakończeniu zadania albo blokada usuwania plików z wybranego folderu.

Moda nie trzeba programować. Wystarczy poprosić Claude, np. *„zrób mi mod z panelem, który pokazuje dzisiejsze spotkania”*. Mody można też rozpowszechniać w pluginach.

Skills w praktyce

- Włączasz w **Ustawienia** → **Capabilities / Skills**. Wbudowane skills Anthropic obsługują m.in. pliki Office i PDF.
- Własny skill możesz wgrać jako ZIP (Chat) albo położyć w `~/.claude/skills/` lub `.claude/skills/` w projekcie (Code).
- Najprościej: poproś Claude „zrób mi skill do…”. Pomoże w tym wbudowany skill *skill-creator*.

Plugins

- Instalowane z katalogu (marketplace): publicznego lub firmowego.
- Organizacja może udostępnić własne pluginy całemu zespołowi. Wszyscy pracują według tych samych procedur.

Konektory (MCP, Model Context Protocol)

- **Zdalne** (w chmurze): dodajesz w Ustawienia → Connectors, logujesz się do usługi. Działają w przeglądarce i w aplikacji.
- **Lokalne** (Desktop): działają na Twoim komputerze, np. dostęp do plików, Power BI Desktop, lokalnych baz.
- Claude zawsze działa z **Twoimi uprawnieniami** w danej usłudze. Przed wysłaniem maila czy usunięciem danych zapyta o zgodę.

> Instaluj tylko pluginy i konektory z zaufanych źródeł. Mają one dostęp do Twoich danych.

5. Artefakty

Artefakt to samodzielny wynik pracy Claude, wyświetlany obok rozmowy: dokument, prezentacja, arkusz, projekt graficzny, dashboard lub tablica. Można go edytować, wersjonować i udostępnić linkiem.

Docs

Dokumenty do wspólnej edycji i komentowania: raporty, procedury, notatki. Eksport do Word, PDF, Markdown, Google Docs.

Slides

Prezentacje z motywem graficznym (design system). Pobierasz jako PPTX lub PDF.

Sheets

Arkusze do wypełniania, sortowania i liczenia: budżety, listy, proste trackery.

Design

Projekty wizualne i makiety: plakaty, ekrany aplikacji, grafiki, prototypy.

Data

Dashboardy i strony z danymi: wykresy, KPI, tabele. Mogą czytać dane na żywo z konektorów.

Whiteboards

Tablice do burzy mózgów i schematów: karteczki, diagramy, mapy procesów.

Jak pracować z artefaktami

- Wystarczy poprosić: „zrób prezentację…", „przygotuj dashboard…". Claude sam dobierze typ.
- Artefakt jest **domyślnie prywatny**. Udostępniasz go linkiem, kiedy zechcesz.
- Poprawki zlecasz w rozmowie („zmień kolor wykresu", „dodaj slajd o kosztach") lub komentarzem w samym artefakcie.
- Artefakty mogą być **interaktywne**: formularze, ankiety, kalkulatory, z danymi wspólnymi dla wszystkich oglądających.
- Wszystkie artefakty znajdziesz w galerii (claude.ai, sekcja Artifacts). W Code: polecenie `/artifacts`.
- Artefakty działają w Chat i w Code.

Artefakt czy plik? Gdy chcesz coś pokazać, udostępnić lub dalej edytować razem z Claude, wybierz artefakt. Gdy potrzebujesz konkretnego pliku (.docx, .xlsx, .pdf) na dysk, powiedz to wprost: „zapisz jako plik PDF".

6. Internet, przeglądarka i pamięć

Claude może sięgać do internetu i stron WWW na kilka sposobów. Różnią się zakresem i tym, czyje konto jest używane.

Narzędzie	Co robi	Gdzie
Web search	Szybkie wyszukiwanie w internecie, odpowiedź ze źródłami	Chat, Code
Research	Głębokie badanie: kilkadziesiąt źródeł, kilka minut pracy, raport z przypisami	Chat
Wbudowana przeglądarka	Osobny panel przeglądarki w aplikacji. Claude czyta strony, klika, testuje Twoją aplikację. Odizolowana od Twojego Chrome.	Desktop (Code)
Claude in Chrome	Rozszerzenie do Twojego Chrome: działa w Twoich kartach i na Twoich zalogowanych kontach	Chrome + Chat lub Code
Computer use	Claude widzi ekran i steruje myszą oraz klawiaturą w aplikacjach (Excel, Notatnik, eksplorator plików)	Desktop

Bezpieczeństwo: Claude nie wpisuje haseł, numerów kart ani nie rozwiązuje CAPTCHA. Przed wysłaniem wiadomości, zakupem czy usunięciem danych zawsze pyta o zgodę. Treść stron traktuje jako dane, nie jako polecenia.

Pamięć (Memory)

W Chat

- Claude zapamiętuje fakty i preferencje z rozmów (np. „piszę po polsku, pracuję w dziale finansów”).
- Pamięć można przeglądać, edytować i wyłączyć w Ustawieniach.
- Projekty mają osobną pamięć. Rozmowa incognito nie zapisuje niczego.
- Możesz zaimportować pamięć z innego asystenta AI.

W Code

- **CLAUDE.md**: instrukcje, które piszesz Ty (lub Claude na Twoją prośbę).
- **Auto-memory**: Claude sam zapisuje w osobnych plikach, czego się nauczył o Tobie i projekcie.
- Wystarczy powiedzieć: „zapamiętaj, że…”.

> Pamięć to **notatki Claude o Tobie**. CLAUDE.md to **Twoje instrukcje dla Claude**. Ważne zasady zapisuj w CLAUDE.md, bo masz nad nimi pełną kontrolę.

7. Usage: limity i jak je oszczędzać

Każdy plan ma limit użycia. Chat i Code korzystają z tej samej puli.

Jak działają limity

- **Limit sesyjny**: odnawia się co kilka godzin (okno ok. 5 godzin od pierwszej wiadomości).
- **Limit tygodniowy**: dodatkowy, zwłaszcza dla najmocniejszych modeli.
- Stan sprawdzisz w **Ustawienia** → **Usage**. W Code: polecenie `/usage` lub pytanie „wyjaśnij moje zużycie”.
- Na planach Pro i Max można dokupić dodatkowe użycie po przekroczeniu limitu.

Co zużywa limit najszybciej

Czynnik	Dlaczego
Długie rozmowy	Przy każdej wiadomości Claude czyta całą dotychczasową historię
Duże pliki	Długie PDF, arkusze i zrzuty ekranu zajmują dużo kontekstu
Najmocniejszy model	Opus i wysoki poziom „myślenia” kosztują więcej niż Sonnet czy Haiku
Agenty i sesje równoległe	Kilka agentów naraz to kilka razy większe zużycie

Research i przeglądarka	Wiele stron do przeczytania

Dobre nawyki

- Nowy temat to nowa rozmowa lub sesja.
- Do prostych zadań wybieraj Sonnet lub Haiku.
- Wgrywaj tylko potrzebne fragmenty dokumentów.

- Stałe informacje trzymaj w Projekcie lub CLAUDE.md, a nie w każdej wiadomości.
- W Code: `/compact` streszcza długą sesję, `/clear` czyści kontekst.
- Precyzyjne polecenie oznacza mniej poprawek.

8. Claude Code: interfejs

Zakładka Code w aplikacji Desktop to wygodne okno na Claude Code: bez terminala, z panelami na pliki, zmiany i przeglądarkę.

Environment (środowisko)

Przy tworzeniu sesji wybierasz, **gdzie** Claude pracuje:

- **Local**: na Twoim komputerze, we wskazanym folderze.
- **Worktree**: osobna kopia repozytorium Git, więc zmiany nie mieszają się z Twoimi.
- **Cloud**: w chmurze Anthropic, działa nawet przy wyłączonym komputerze (repozytorium z GitHub).
- **SSH**: na zdalnym serwerze.

Session rail (pasek sesji)

- Lista sesji po lewej stronie, każda z tytułem i statusem.
- Widać, która sesja pracuje, która czeka na Twoją decyzję, a która skończyła.
- Sesje można przypinać, oznaczać jako zakończone i archiwizować.

Side chat (rozmowa boczna)

- Szybkie pytanie „na boku", bez przerywania i zaśmiecania głównej sesji.
- Np. „co znaczy ten błąd?" albo „jak działa ta funkcja?", podczas gdy główna sesja dalej pracuje.

Split view (widok podzielony)

- Dwie sesje obok siebie w jednym oknie.
- Przeciągnij sesję z paska w wybrane miejsce okna lub kliknij prawym przyciskiem → Open in → Split view.
- Przydatne przy porównaniu wyników lub pracy nad dwoma zadaniami naraz.

Sekcje i grupowanie

- **Sekcje**: własne nagłówki w pasku sesji, np. „Szkolenie", „Raporty", „Artykuły".
- **Grupowanie**: sesje można grupować automatycznie (np. według folderu lub statusu) albo przenosić ręcznie.
- Rozdziały (chapters) dzielą długą sesję na etapy. Spis rozdziałów pozwala przeskakiwać między nimi.

Panele w Code

Oprócz rozmowy możesz otworzyć z boku okna dodatkowe panele:

Panel	Co pokazuje	Kiedy się przydaje
Diff (zmiany)	Pliki zmienione przez Claude w tej sesji: dodane linie na zielono, usunięte na czerwono	Przed zatwierdzeniem: sprawdzasz, co dokładnie się zmieniło
Terminal	Okno poleceń (np. PowerShell), w którym Ty lub Claude uruchamiacie polecenia i skrypty	Gdy chcesz zobaczyć wynik skryptu albo sam coś uruchomić
Browser (przeglądarka)	Wbudowana przeglądarka, w której Claude otwiera strony, czyta je i klika	Podgląd tworzonej strony, sprawdzanie informacji w sieci
Tasks (zadania)	Lista kroków bieżącego zadania, odhaczanych na bieżąco	Przy dłuższym zadaniu: widzisz, na jakim etapie jest Claude
Plan	Plan działania przygotowany w trybie Plan, przed jakąkolwiek zmianą	Przy większych zmianach: najpierw akceptujesz plan, potem Claude działa

9. Claude Code: praca równoległa i automatyzacja

Code potrafi prowadzić kilka zadań naraz i wykonywać zadania według harmonogramu.

Równoległe sesje

- Kilka niezależnych zadań w osobnych sesjach naraz, np. jedna pisze raport, druga porządkuje dane.
- Claude może sam zaproponować wydzielenie zadania do nowej sesji. Pojawia się wtedy karta „New task".
- Przy pracy na tych samych plikach używaj **worktree**, żeby sesje sobie nie przeszkadzały.

Agenty (subagenty)

- Wewnątrz jednej sesji Claude może uruchomić pomocników do wycinka pracy: przeszukania plików, researchu, przeglądu.
- Wynik wraca do głównej sesji, a jej kontekst zostaje czysty.
- Własne agenty definiujesz w `.claude/agents/`.

Zadania cykliczne

Rodzaj	Jak działa	Przykład
Scheduled tasks (lokalne)	Uruchamiane w aplikacji Desktop o stałej porze, gdy komputer jest włączony	Co rano: przegląd skrzynki i kalendarza, poranny brief
Routines (w chmurze)	Agent w chmurze według harmonogramu (cron), działa bez Twojego komputera	Co poniedziałek: raport z repozytorium, kontrola danych
/loop	Powtarzanie polecenia w bieżącej sesji co określony czas	Co 10 minut sprawdzaj, czy skończył się import

Wystarczy powiedzieć: *„codziennie o 8:00 przygotuj podsumowanie nowych maili"*, a Claude utworzy zadanie.

Ściąga: od czego zacząć

1. Rozmowa, pomysł, tekst: **Chat**, najlepiej w **Projekcie**.
2. Praca na plikach: **Code**, wskaż folder, uruchom `/init` i uzupełnij **CLAUDE.md**.
3. Powtarzasz coś co tydzień? Zrób **skill**. Ma się dziać samo? Zrób **zadanie cykliczne**.
4. Potrzebujesz danych z Gmaila, Drive czy CRM? Podłącz **konektor**.
5. Wynik do pokazania lub udostępnienia: **artefakt**. Plik na dysk: poproś o konkretny format.
6. Pilnuj **usage**: dobieraj model do zadania i zaczynaj nowe rozmowy.

Fundusze Europejskie dla Rozwoju Społecznego
Rzeczpospolita Polska
Dofinansowane przez Unię Europejską