



Code · Cowork · CLI

AUTONOMI

MULIGHETER OG RISIKO



✦ Login successful. Press **Enter** to continue

Kenneth Bareksten

Automatiseringsoversikt

```
import Claude from '@anthropic/agent',
// versjon 3.0.0 – autonomi: Muligheter og risiko

const bok = {

  // Del 1: Autonomi i praksis

  kap1 'Act without asking', // s. 4
  kap2 'Riktig innsats til riktig oppgave', // s. 7
  kap3 'Fortsett der du slapp', // s. 9

  // Del 2: Claude på avstand

  kap4 'Dispatch – én tråd, to skjermer', // s. 11
  kap5 'Sett Claude i kø', // s. 13
  kap6 'Live Artifacts', // s. 15

  // Del 3: Computer Use

  kap7 'Claude ser skjermen din', // s. 17
  kap8 'Computer Use i praksis', // s. 19

  // Del 4: Multi-agent

  kap9 'Sub-agents – del opp arbeidet', // s. 21
  kap10 'Orkestreering uten å miste oversikten', // s. 23

  // Del 5: Kontroll og ressursstyring

  kap11 'Token Economy i autonome sesjoner', // s. 25
  kap12 'Tillatelsesprofiler og trygge grenser', // s. 27
  kap13 'Sikkerhet og risiko', // s. 29

  // Avslutning

  avslutning 'Tillit er ikke fravær av kontroll', // s. 31

  // Appendiks

  appendiks A 'Nordheim-dokumenter', // s. 32
  appendiks B 'Hurtigreferanse', // s. 36
  appendiks C 'Sjekkliste', // s. 37
  appendiks D 'Ordliste', // s. 38
  appendiks E 'Brukseksampler', // s. 39
  appendiks F 'Kilder og ressurser', // s. 42

}
export default bok
```

Innledning

Når du stoler nok på verktøyet til å slippe taket

Bok 1, [Claude Code](#), [Cowork](#) og [CLI](#) forklart, handlet om å komme i gang. Bok 2, [Claude Code Skills](#), handlet om å jobbe smartere. Andre nedlastninger finner du på [nettstedet Lærerliv](#). Denne boken handler om noe annerledes: hva skjer når du slutter å holde Claude i hånden hele veien?

Det er et ubehagelig spørsmål for mange. Jeg merket det selv første gang jeg skrudde på Act without asking og så Claude begynne å ta beslutninger uten å spørre meg om lov. Det gikk fint. Nesten for fint. Og da satt jeg igjen med en ny bekymring som er mer interessant enn den første: ikke om Claude kan gjøre jobben, men om jeg faktisk vet hva jeg vil at den skal gjøre. Det er der denne boken starter.

De to foregående bøkene lærte deg kommandoer, struktur og arbeidsflyt. Alt det er fortsatt gyldig. Men autonomi krever noe mer. Det krever at du har tenkt gjennom hva Claude kan avgjøre på egenhånd og hva som alltid skal gjennom deg. Uten det er Act without asking bare kaos med høyere hastighet.

Gjennom boken bruker jeg ett gjennomgående prosjekt: du skal produsere en rapport, en presentasjon og en oppfølgingsplan fra det samme kildematerialet. Ikke fordi det er et spesielt spennende prosjekt, men fordi det er gjenkjennelig for de fleste som jobber med dokumenter og informasjon. Og fordi det er komplekst nok til å vise hva som faktisk skjer når du setter Claude i gang og trekker deg tilbake.

Boken handler om Claude-appen. Den har tre overflater: Chat, Cowork og Code. Cowork er primærverktøyet gjennom hele boken. Der det er relevant å skille mellom overflatene er det gjort tydelig i teksten. Noen av de nye funksjonene finnes også i CLI, og der det er vesentlige forskjeller er det merket med et eget avsnitt.

Hva som er nytt

Siden bok 2 ble skrevet har Anthropic lansert Act without asking, effort-sliden, Recap av sesjoner, Remote Control, Computer Use, sub-agents, Dispatch og Channels. Ingen av disse er dekket i de to første bøkene og vil bli forklart her.

Noen av dem er kraftige nok til å endre hvordan du jobber. Noen er fortsatt skjøre nok til at du bør vite hva du går inn i. Boken forteller deg begge deler.

Autonomi i praksis

Kapittel 1

Act without asking

Tre overflater, ett verktøy

Desktop-appen har tre hovedfaner som deler den samme underliggende teknologien, men som er bygget for ulike formål:

- **Chat:** Brukes til samtaler, skriving og analyse. Den har ingen direkte filtilgang eller autonome handlinger.
- **Cowork:** Primærverktøyet for kunnskapsarbeid. Den kan lese og skrive til lokale filer, koordinere sub-agenter og levere ferdige dokumenter.
- **Code:** Spesialtilpasset programvareutvikling med parallelle sesjoner, integrert terminal, fileditor og app-preview.

CLI (Command Line Interface) gir deg direkte tilgang til motoren bak Code-fanen fra terminalen. Det er mer fleksibelt for automatisering og avansert konfigurasjon

Hva Act without asking er

Dette er en funksjon som gir agenten begrenset autonomi. Den fungerer som en klassifikator som sorterer hver handling Claude ønsker å utføre i to kategorier:

1. Trygge handlinger: Lese filer, skrive utkast og opprette nye dokumenter. Disse kjøres uten å spørre.
2. Kritiske handlinger: Sletting, sending av informasjon eller endringer i systemfiler. Her stopper Claude og venter på din godkjenning.

Målet er å fjerne de unødvendige spørsmålene som bremser fremdriften, uten at du mister kontrollen.

Grensen er ikke alltid der du tror. Mer om det i kapittel 11 om tillatelsesprofiler.

Ikke det samme som å la Claude gjøre hva den vil

Det er et viktig skille å ha klart fra starten. **--dangerously-skip-permissions**, som bok 2 nevnte kort, og som man har muligheten til i Claude CLI. Den fjerner nesten alle sperrer. Act without asking gjør ikke det. Det justerer terskelen for hva som krever et tastetrykk fra deg og hva som ikke gjør det.

Tenk på det som forskjellen på en kollega du ber om å ordne noe mens du er i møte, og en kollega du gir nøklene til kontoret og sier gjør det du tror er riktig. Act without asking er det første.

Slik skruer du det på i Cowork

Åpne Cowork. Under samtaleboksen, ved siden av filer og modellvelger, finner du Act without asking. Skru det på.

Claude begynner å bruke klassifikatoren med en gang på den neste oppgaven du gir den. Innstillingen huskes mellom sesjoner, men det er lurt å sjekke at den er der du vil ha den når du starter en ny samtale.

I Code

Code har en modusvelger med to valg: Plan og Accept Edits.

Plan lar Claude lese filer og utforske kodebasen, men ikke endre noe. Nyttig når du vil se hva Claude har tenkt å gjøre før du slipper det løs.

Accept Edits godkjenner filredigeringer og vanlige filsystemkommandoer automatisk, men stopper fortsatt og spør ved alt annet. Act without asking i Cowork er det samme som auto mode i code. Du aktiverer det i innstillingene.

Accept Edits passer når du jobber i et avgrenset prosjekt og vil se endringene i editoren etterpå heller enn å godkjenne dem én og én underveis.

I CLI

CLI har flere moduser enn desktop-grensesnittet, som denne boka fokuserer mest på.

Du bytter med Shift+Tab under en sesjon, eller setter modus ved oppstart. -- **dangerously-skip-permissions** gir Claude tilgang og mulighet til å utføre nesten alt den mener er hensiktsmessig, og tilsvarer det Anthropic kaller **bypassPermissions**, som må velges i innstillingene. Bruk det bare i isolerte miljøer som containere eller virtuelle maskiner. Mer om dette i kapittel 12.

Når du *ikke* skal bruke Act without asking

Act without asking egner seg dårlig når oppgaven innebærer noe som er vanskelig å angre. Sender du Claude av gårde for å rydde opp i en mappestruktur, kan filer havne steder du ikke hadde tenkt. Ber du den oppdatere et dokument andre bruker aktivt, kan du overskrive noe som ikke burde overskrives.

Tommelfingerregelen er enkel: skru på Act without asking når du kan tåle å angre på det Claude gjør. Skru det av når du ikke kan det.

Prosjektet i boka: Nordheim AS

I denne boken bruker jeg ett gjennomgående prosjekt. Du jobber som ekstern konsulent og skal hjelpe det fiktive selskapet Nordheim AS med å oppsummere, presentere og følge opp arbeidet med den nye møtестrukturen deres.

Materialet ligger i Appendiks A. Fire dokumenter: et møtereferat, en statusrapport, en e-posttråd og et notat med uløste spørsmål. Til sammen gir de deg et bilde av en organisasjon midt i en prosess, med friksjon og noen uavklarte spørsmål. Resultatet du skal levere når vi er ferdige, er en rapport, en presentasjon og en oppfølgingsplan. Alle tre fra det samme kildematerialet.

Tips: Gjør dette før du går videre. Gå til Appendiks A. Opprett en ny mappe på maskinen din og kall den nordheim. Kopier de fire dokumentene inn i mappen. Legg mappen til i Cowork.

Øvelse 1: Første oppgave med Act without asking på

1. Åpne Cowork og skru på Act without asking under Permissions.
2. Legg nordheim-mappen til i Cowork hvis du ikke har gjort det.
3. Skriv: "Les de fire dokumentene i nordheim-mappen og lag et kort sammendrag av hva Nordheim AS holder på med og hva som er uavklart."
4. Observer hva Claude gjør uten å spørre deg om lov, og hva den eventuelt stopper opp ved.
5. Sjekk sammendraget. Er det presist? Er noe viktig utelatt?

Ikke forsøk å lage rapporten ennå. Det kommer i kapittel 3. Denne øvelsen handler utelukkende om å se Act without asking i arbeid på en trygg oppgave.

Husk dette



- Act without asking sorterer handlinger i trygt og vent, ikke i alt eller ingenting.
- I Code velger du mellom Plan og Accept Edits. Accept Edits er ikke det samme som å fjerne alle sperrer.
- I CLI gir `--dangerously-skip-permissions` full autonomi. Bruk det bare i isolerte miljøer.
- Nordheim-prosjektet starter i Appendiks A med fire dokumenter du kopierer inn i en mappe kalt nordheim.

Autonomi i praksis

Kapittel 2

Riktig innsats til riktig oppgave

Ikke alle oppgaver fortjener det samme. Et raskt sammendrag av ett dokument krever ikke det samme som en fullstendig analyse av fire kildefiler med motstridende informasjon. Hvis Claude bruker maksimal innsats på alt, tar det lengre tid, koster mer og gir deg ikke nødvendigvis bedre resultater på de enkle tingene. Innsatsstyring handler om å matche ressursbruk med oppgavens faktiske kompleksitet.

Hva innsats egentlig er

Når Claude tenker mer, bruker den flere tokens på resonnering før den svarer. Det betyr lengre ventetid, høyere kostnader og ofte bedre resultater på komplekse oppgaver. På enkle oppgaver er det sløsing. Et sammendrag av et møtereferat trenger ikke dyp analyse. En vurdering av fire kildefiler med motstridende informasjon gjør det.

Innsatsstyring er ikke en mirakelknapp. Det endrer ikke hva Claude kan gjøre, men hvor grundig den tenker før den gjør det.

Adaptive Thinking i Cowork

I Cowork finner du Adaptive Thinking som en toggle i modellvelgeren. Den er tilgjengelig for alle modeller. Når den er på, vurderer Claude selv om oppgaven krever utvidet resonnering før den svarer.

Tommelfingerregelen: skru på Adaptive Thinking når oppgavene varierer mye i kompleksitet i løpet av en sesjon, eller når du ikke vet på forhånd hvor krevende noe er. Skru den av når du vil ha raske, forutsigbare svar på enkle oppgaver.

I Code

I Code-fanen finner du innsatsnivå som et valg i modellvelgeren. Du velger mellom low, medium, high og max. Valget gjelder for hele sesjonen inntil du endrer det. Bruk low når Claude skal lese eller sammenfatte noe uten å vurdere. Bruk high når oppgaven krever at Claude veier alternativer, identifiserer motsetninger eller produserer noe som skal stå seg under kritisk gjennomgang.

I CLI

Skriv **!effort** uten argumenter for å åpne en interaktiv slider. Bruk piltastene til å velge nivå og trykk Enter for å bekrefte. Du kan også sette det direkte, for eksempel med `!effort low` eller `!effort high`. Innsatsnivået vises i statuslinjen mens du jobber.

Når høy innsats ikke virker

Mer tenking hjelper ikke hvis spørsmålet er upresist. Skriver du "hva synes du om dette dokumentet?" med høy innsats, får du et lengre svar, ikke nødvendigvis et bedre. Innsats forsterker kvaliteten på et godt spørsmål. Det kompensere ikke for et dårlig ett.

Nordheim AS: samme materiale, ulik innsats

Du har fire kildefiler fra Nordheim AS. Noen oppgaver i dette prosjektet er enkle. Andre er ikke det.

Et sammendrag av møtereferatet er en lav-innsats-oppgave. Claude leser, trekker ut hovedpunkter og leverer. Ingen avveininger, ingen motstridende informasjon å veie opp mot hverandre.

En vurdering av hva som er uavklart på tvers av alle fire dokumentene er noe annet. E-posttråden og notatet med uløste spørsmål kan si forskjellige ting om samme sak. Det krever at Claude faktisk tenker, ikke bare sammenfatter.

Øvelse 2: Samme oppgave, to innsatsnivåer

Bruk Cowork med Adaptive Thinking av for denne øvelsen, slik at du selv styrer innsatsen.

1. Still lav innsats og skriv: "Gi meg en liste over de viktigste beslutningene som er tatt i møtereferatet fra Nordheim AS."
2. Still høy innsats og skriv: "På tvers av alle fire dokumentene, hva er uavklart, og er det motsetninger mellom det som står i e-posttråden og notatet med uløste spørsmål?"

Sammenlign ikke bare svarene, men også ventetiden. Det er forskjellen mellom lav og høy innsats i praksis.

Husk dette



- Adaptive Thinking i Cowork lar Claude vurdere innsatsbehovet selv. Nyttig når oppgavene varierer.
- I Code og CLI setter du innsatsnivå manuelt.
- Høy innsats gir bedre resultater på komplekse oppgaver. Et presist spørsmål med lav innsats slår ofte et upresist spørsmål med høy innsats.

Autonomi i praksis

Kapittel 3

Fortsett der du slapp

Langvarige prosjekter lever ikke i én sesjon. Du starter, kommer et stykke, lukker maskinen og er tilbake neste dag. Spørsmålet er ikke om du kan fortsette, men hvor mye du må forklare på nytt før du faktisk er i gang igjen.

Cowork: prosjekter med innebygd hukommelse

I de første ukene etter lansering startet hver Cowork-sesjon fra null. Du forklarte oppdraget på nytt og håpet resultatet ble konsistent. Prosjekter fikset det.

Et Cowork-prosjekt er et dedikert arbeidsrom med instruksjoner og filer som overlever mellom sesjoner. Hukommelsen er innebygd. Claude husker hva som er gjort i prosjektet og bygger på det neste gang du åpner det.

I Code

Code har ingen tilsvarende prosjektfunksjon. Det som gir kontinuitet der er CLAUDE.md, en fil du oppretter i prosjektmappen og fyller med instruksjoner Claude laster ved sesjonstart.

CLAUDE.md er ikke automatisk hukommelse. Det er det du selv skriver. Men det er forutsigbart: det samme lastes hver gang, uansett hva som skjedde sist. For Nordheim betyr det en CLAUDE.md som beskriver oppdraget og hva som er levert. Du oppdaterer den etter hver arbeidsøkt.

I CLI

claude --continue gjenopptar siste sesjon i gjeldende mappe uten å spørre. Samtalehistorikken og alt Claude leste og kjørte sist er fortsatt i kontekst. Bruk den når du nettopp la fra deg arbeidet

Øvelse 3a: Sett opp Nordheim som Cowork-prosjekt

1. Gå til Cowork-fanen. Klikk Projects i venstremenyen, deretter +. Velg "Use an existing folder" og pek på nordheim-mappen fra kap. 1.
2. Du får mulighet til å gi prosjektet et navn og skrive instruksjoner. Skriv noe slikt: "Jeg er ekstern konsulent for Nordheim AS. Oppdraget er å levere en rapport, en presentasjon og en oppfølgingsplan basert på fire kildefiler i denne mappen. Leveransen er ikke ferdig."
3. Klikk Create.

Neste gang du vil tilbake: klikk søkeikonet i Cowork, finn en tidligere samtale i prosjektet og klikk på den. Du er inne igjen med full kontekst.

claude **--resume** åpner en velger med alle tidligere sesjoner. Filtrer på gjeldende mappe, alle prosjekter eller git-gren. Bruk den når du vil tilbake til en sesjon fra noen dager siden.

/recap gir en situasjonsoppdatering inne i en aktiv sesjon. Nyttig etter lengre fravær.

Øvelse 3b: Gjenopta en CLI-sesjon

1. Start en ny CLI-sesjon i nordheim-mappen. Gi Claude én oppgave: "Lag et kort sammendrag av møterefateratet." Avslutt sesjonen.
2. Start terminalen igjen og skriv claude **--continue**. Sjekk at Claude vet hva den holdt på med. Skriv deretter **/recap** og les oppdateringen. Stemmer den?

Selve testen er ikke om Claude husker. Det gjør den. Testen er om du stoler nok på det til å ikke forklare alt på nytt.

Husk dette



- Cowork krever at du selv vedlikeholder kontekst mellom sesjoner via prosjekter og kontekstfiler.
- Code lagrer sesjonsminner automatisk og lar deg løfte dem til permanente instruksjoner med **/remember**.
- I CLI gir **--continue** deg raskest tilbake til der du var, **--resume** gir deg kontroll over hvilken sesjon du vil tilbake til, og **/recap** orienterer deg etter lengre fravær.

Claude på avstand

Kapittel 4

Dispatch — én tråd, to skjermer

Hvor mange ganger har du sittet i et møte og husket at noe ikke er gjort?

Med Dispatch sender du oppgaven fra telefonen: hva som skal gjøres, innen når, og hvor resultatet skal lagres. Desktop-appen tar det derfra.

Dette er Ikke en ny modell. Ikke et eget abonnement. Det er et arbeidsflytlag i Cowork som kobler mobilappen til desktop-appen.

Hva Dispatch er og ikke er

Dispatch er en fjernkontroll, ikke en frittstående agent. Claude kjører på maskinen din. Telefonen sender instruksjoner og tar imot resultater. Lukker du laptopen, slutter Claude å jobbe.

Det finnes én persistent tråd mellom telefon og desktop. Du kan starte en oppgave fra telefonen, åpne maskinen og fortsette i samme samtale, og sende nye instruksjoner fra telefonen igjen. Konteksten følger med hele veien. Det som ikke fungerer er oppgaver som krever godkjenning underveis. Claude stopper og venter, men du får ikke alltid varsel.

Påliteligheten varierer. Enkle henting- og oppsummeringsoppgaver fungerer som regel. Krysser oppgaven flere apper er det mer usikkert. Dispatch er i research preview.

Sender du en koderelatert oppgave via Dispatch, ruter den automatisk til Code-fanen og starter en sesjon der med et Dispatch-merke. Oppgaven havner der den hører hjemme uten at du trenger å gjøre noe.

Slik setter du den opp

Klikk Dispatch i venstremenyen i Cowork og klikk "Get started". Gi Claude tilgang til filene dine og skru på "hold maskinen våken". Skann QR-koden med Claude-appen på telefonen. Etter paring dukker Dispatch opp som en egen samtale i mobilappen.

Remote Control: se hva Claude gjør

Dispatch er asynkront. Du sender en oppgave og venter på resultatet. Noen ganger vil du heller følge med på hva Claude gjør mens det skjer.

Remote Control gir deg et fullstendig claude.ai-grensesnitt inn i en kjørende CC-sesjon (Claude Code), åpent fra telefon eller nettleser. Du ser alt i sanntid og kan gripe inn når du vil. Start en Code-sesjon og skriv **/remote-control**. Du får en URL og en QR-kode. Åpne URL-en på telefonen.

Ingen ekstra oppsett og ingen tredjepartsapper. Remote Control er det naturlige valget når du vil ha oversikt og muligheten til å korrigere underveis.

Channels: CLI fra telefonen

Channels er et alternativ for de som jobber i CLI, men en viktig presisering: Channels er i research preview og ikke tilgjengelig for alle Pro-brukere ennå. Sjekk om du har tilgang før du prøver.

Channels kobler en kjørende Code-sesjon til Telegram, Discord eller iMessage via en MCP-plugin. Du sender en melding til boten din, Code plukker den opp og svarer tilbake i samme kanal. Det krever Bun og Claude Code v2.1.80 eller nyere.

Start med `claude --channels plugin:telegram@claude-plugins-official`. Første gang du sender en melding, får du en paringskode du bekrefter i CLI. Koblingen er deretter låst til din konto. Channels har fått permission relay, som videresender godkjenningsforespørsler til telefonen i stedet for å stoppe sesjonen. Du godkjenner direkte fra Telegram.

Channels fungerer bare mens sesjonen er åpen. Trenger du noe som kjører etter at CLI er lukket, er tmux eller en VPS løsningen.

Øvelse 4: Dispatch i praksis

1. Sett opp Dispatch og skann QR-koden hvis du ikke har gjort det. Gå fra maskinen med telefonen.
2. Åpne Claude-appen, finn Dispatch-samtalen og skriv: "Les møtereferatet fra Nordheim AS og lag tre konkrete oppfølgingspunkter. Lagre dem som oppfølging.md i nordheim-mappen."
3. Kom tilbake etter ti minutter. Er filen der? Er innholdet presist? Måtte Claude stoppe og vente på noe?

Husk dette



- **Dispatch** er en fjernkontroll for Cowork, ikke en frittstående agent. Maskinen må være våken og appen åpen. Koderelaterte oppgaver ruter automatisk til Code-fanen.
- **Remote Control** gir deg sanntidsoversikt inn i en kjørende Code-sesjon via URL eller QR-kode.
- **Channels** er kun for CLI-brukere med research preview-tilgang, og fungerer bare mens sesjonen er åpen.

Claude på avstand

Kapittel 5

Sett Claude i kø

Hvor mange oppgaver vet du om på forhånd at skal være klare til et bestemt tidspunkt?

Scheduled Tasks lar deg sette opp jobben i forkant. Kildefilene ligger der, tidspunktet er satt, og Claude kjører det ferdig mens du sover eller sitter i møte. Du trenger ikke være til stede for å starte det.

Scheduled Tasks i Cowork

Scheduled Tasks lar deg beskrive en oppgave én gang og velge når den skal kjøre. Daglig, ukentlig eller månedlig. Claude kjører oppgaven som en ny Cowork-sesjon og leverer resultatet til filsystemet ditt, klar til å åpne når du er tilbake.

Oppsettet tar ett minutt. Skriv **/schedule** i en Cowork-oppgave, eller klikk "Scheduled" i venstremenyen. Beskriv hva Claude skal gjøre og velg kadense. Claude avklarer detaljer ved behov, oppretter oppgaven og viser deg en bekreftelse. Deretter dukker den opp under "Scheduled" i sidefeltet fra og med neste kjøring.

En begrensning å kjenne til: oppgaven kjører bare mens maskinen er våken og appen er åpen. Sover maskinen når oppgaven skal kjøre, hopper Claude over og kjører neste gang du åpner appen. Du får varsel om hoppet kjøring.

Scheduled Tasks har tilgang til lokale filer og koblede verktøy. Det skiller dem fra Routines, som vi kommer til om et øyeblikk.

I Code

Code har tilsvarende funksjonalitet. Klikk Schedule i Code-sidefeltet, deretter "New task" og "New local task". Fyll inn prompt og velg kadense. Du kan også si det direkte i en aktiv sesjon: "sett opp en daglig gjennomgang av nordheim-mappen klokka 07:00". Samme begrensning som Cowork: maskinen må være åpen

Routines: Claude kjører uten at maskinen er på

Routines ble lansert 14. april 2026 i research preview. De kjører på Anthropic's skyinfrastruktur, ikke på maskinen din. Du kan lukke laptopen, legge deg, ta et fly. Rutinen kjører likevel.

Du oppretter dem fra Code ved å klikke "New task" og velge "New remote task", eller direkte på claude.ai/code/routines.

En Routine har ikke tilgang til lokale filer. Den jobber med det du har koblet til kontoen din: Gmail, Slack, Google Drive. Trenger Claude lese noe som bare finnes på maskinen din, er Scheduled Tasks riktig valg. Trenger Claude bare koble seg til eksterne tjenester, kan Routines kjøre uavhengig av om maskinen er på.

Det er en daglig grense på antall kjøring per konto. Sjekk forbruk på claude.ai/settings/usage.

I CLI

`/loop` er for oppgaver du vil repetere mens en sesjon er åpen. Den dør når du lukker CLI. For planlagte oppgaver som skal overleve en omstart, bruk Code. For sky-baserte rutiner, skriv `/schedule` i CLI for å opprette en Routine direkte fra terminalen.

Øvelse 5: Sett opp en planlagt Cowork-oppgave

1. Åpne Cowork og skriv `/schedule`. Beskriv oppgaven slik:
2. "Sjekk nordheim-mappen for endringer siden siste kjøring. Oppdater filen `status.md` med et kort sammendrag av hva som er nytt og hva som gjenstår av leveransene."
3. Velg daglig som kadense. La Claude bekrefte og klikk Schedule. Gå til "Scheduled" i sidefeltet og bekreft at oppgaven er registrert.
4. Maskinen må være åpen for at den kjører. Vil du at oppsummeringen skal skje uavhengig av maskinen, er det en oppgave for en Routine med Google Drive eller en annen kobling i stedet for lokale filer.

Husk dette



- Scheduled Tasks krever åpen maskin og har tilgang til lokale filer.
- Routines kjører i skyen uavhengig av maskinen, men bare med koblede tjenester. Code har planleggingsfunksjonen i Schedule-sidefeltet.
- I CLI er `/loop` for sesjoner og `/schedule` for sky-rutiner.

Claude på avstand

Kapittel 6

Live Artifacts, oppdatert dashboard

Hvor mange ganger har du laget et sammendrag som var utdatert dagen etter? Noen la til et notat. En e-posttrådfikk et nytt svar. Statusrapporten ble oppdatert. Det du laget stemte ikke lenger, og du måtte starte på nytt.

Live Artifacts løser det. Ikke ved å oppdatere dokumentet kontinuerlig, men ved å hente ferske data hver gang du åpner det.

Hva et Live Artifact er

Et vanlig Cowork-output er et øyeblikksbilde. Claude kjører, leverer og er ferdig. Live Artifacts fungerer annerledes. De er dashboards og trackere koblet til datakildene dine. Åpner du et Live Artifact i morgen, leser det dataene på nytt og viser deg det som er aktuelt nå.

Det er en lagret instruksjon med tilgang til det du har koblet inn. Hvert Artifact vet hvilke kilder det skal lese og hva det skal presentere. Du åpner det og det gjør jobben.

Slik lager du et

Klikk "Live Artifacts" i venstremenyen i Cowork. Du får to valg: lag et nytt fra bunnen av, eller velg blant ferdige maler. "Week at a glance" gir deg en uksoversikt basert på kalender og e-post. "What needs my attention" trekker ut det som venter på handling fra koblede tjenester.

Vil du lage noe tilpasset, klikk "New artifact" og beskriv hva du vil se. Claude spør hvilke kilder som skal inn og setter det opp.

Over hundre koblinger er tilgjengelig under "Browse connectors", fra Google Workspace og Microsoft 365 til Slack, Notion og Jira. Live Artifacts finnes bare i Cowork. Det finnes ingen tilsvarende funksjon i Code eller CLI.

Nordheim AS: en oversikt som holder seg aktuell

Nordheim-prosjektet i denne boken har fire statiske kildefiler. Et Live Artifact er ikke det riktige verktøyet for det. Men tenk deg at prosjektet hadde gått over måneder, at referater kom til underveis og e-posttråder vokste. Da hadde et Artifact som viste hva som er uavklart og hva som er levert, spart deg for å kjøre samme prompt på nytt etter hver oppdatering.

Pågående arbeid med informasjon som lever og endrer seg. Det er der Live Artifacts henter seg inn.

Øvelse 6: Lag ditt første Live Artific

1. Klikk "Live Artifacts" i Cowork-sidefeltet. Velg "Week at a glance" og koble til kalenderen din. Åpne artifactet.
2. Stemmer det med hva du faktisk har på agendaen? Juster beskrivelsen hvis noe mangler, og åpne det igjen i morgen.
3. Da er det neste uke som vises.

Husk dette



- Live Artifacts henter ferske data ved åpning, ikke kontinuerlig.
- Velg mellom ferdige maler eller bygg fra bunnen av.
- Tilgjengelig bare i Cowork. Nyttig for pågående arbeid der informasjonen endrer seg over tid, ikke for statiske enkeltprosjekter.

Computer use

Kapittel 7

Claude ser skjermen din

Første gang jeg ga Claude tilgang til skjermen min, ba jeg den åpne et nettskjema og fylle inn noen felter. Jeg så markøren bevege seg over skjermen, klikke på riktig felt og skrive inn teksten. Det tok lengre tid enn om jeg hadde gjort det selv. Men jeg trengte ikke å gjøre det selv.

Det er Computer Use.

Ikke første utvei, men siste

Når du gir Claude en oppgave, jobber den seg gjennom en prioritert liste. Finnes det en direkte kobling til appen, Gmail, Slack eller Google Drive, bruker Claude den. Er det ingen kobling, åpner Claude Chrome og navigerer der i stedet. Bare hvis ingen av disse fungerer, tar Claude kontroll over skjermen.

Skjermkontroll er langsommere og mer feilbarlig enn en direkte kobling. Systemet er bygget for at Claude skal unngå det der det er mulig.

Hva du trenger

Computer Use er tilgjengelig i Cowork og Code på macOS og Windows. Du trenger Pro eller Max. Team og Enterprise har ikke tilgang ennå. Funksjonen er i research preview.

På Windows krever Cowork at Virtual Machine Platform er aktivert og at du installerte appen med administratortilgang. Uten admin-rettigheter er Cowork utilgjengelig uansett abonnement.

Tillatelser og kontroll

Det er ingen sandkasse mellom Claude og appene dine. Claude klikker og navigerer direkte i det du har på skjermen. Første gang Claude trenger tilgang til en ny app, spør den om lov. Du godkjenner eller avviser.

Investeringsplattformer og kryptovaluta-apper er blokkert som standard og kan ikke åpnes av Claude uansett hva du sier.

Du kan legge til egne apper i en blokkerte-liste under innstillinger. Gjør det for apper du ikke vil at Claude skal berøre.

Hva det egner seg til

Computer Use gir mening der det ikke finnes en direkte integrasjon. Et internt HR-system uten API. Et nettskjema som ikke kan fylles ut på annen måte. En app som bare finnes som skrivebordsversjon.

Det er ikke egnet for oppgaver som krever fart eller høy presisjon. En oppgave du gjør på 30 sekunder selv, kan ta Claude tre minutter via skjermen. Tålmodighet er ikke det samme som effektivitet.

I Code

Computer Use gir mening der det ikke finnes en direkte integrasjon. Et internt HR-system uten API. Et nettskjema som ikke kan fylles ut på annen måte. En app som bare finnes som skrivebordsversjon.

Øvelse 7: Aktiver Computer Use og test tillatelsessystemet

1. Gå til Settings i Claude-appen og aktiver Computer Use under General. Åpne Cowork med nordheim-prosjektet og gi Claude denne oppgaven:
2. "Nordheim AS jobber med å innføre en ny møtestruktur. Søk opp hva forskning og fagmiljøer sier om effektive møtestrukturer i mindre organisasjoner. Lagre et kort sammendrag i nordheim-mappen som bakgrunn.md."
3. Observer hva Claude gjør. Bruker den en direkte kobling, eller åpner den Chrome og navigerer? Ber den om tillatelse til nettlesertilgang?

Resultatet i bakgrunn.md kan du bruke som kontekst når du skriver rapporten i kap. 3.

Husk dette



- Computer Use krever Pro eller Max. Team og Enterprise har ikke tilgang ennå.
- Claude bruker skjermkontroll bare som siste utvei. Første gangs tilgang til en app krever godkjenning.
- Investeringsplattformer og kryptovaluta er blokkert som standard.
- Funksjonen er i research preview.

Computer use

Kapittel 8

Computer use i praksis

Det er en ting å forstå hva Computer Use er. En annen å bruke det på noe som faktisk betyr noe.

Kap. 7 forklarte prioriteringskjeden: kobling først, nettleser neste, skjerm sist. Her ser vi hva som skjer i praksis når Claude tar kontroll over skjermen, og like viktig: når det ikke burde gjøre det.

Hva som egner seg

Skjermkontroll gir mening der verken kobling eller nettleser strekker til. Et program som bare finnes som skrivebordsapp. Et internt system uten API. En arbeidsflyt som krever at Claude navigerer mellom flere apper på skrivebordet.

Nordheim-prosjektet er ikke det. Det meste av arbeidet der løses fint med direkte koblinger eller nettleseknavigering. Computer Use frister til å bli brukt bare fordi det er mulig. Det er sjelden godt nok grunnlag.

Hva som ikke egner seg

Presisjon. Claude leser skjermbilder og gjetter på hva som er klikkbart, og den gjetningen er ikke alltid riktig. I Code og CLI gjelder egne tilgangsnivåer per app-kategori: nettlelere er kun for visning, terminaler og IDE-er kun for klikk. Resten får full kontroll. Kjenn disse grensene før du sender Claude av gårde.

Fart er heller ikke Computer Use sin sterkeste side. En oppgave du gjør på 30 sekunder selv, kan ta Claude tre minutter via skjermen.

Fallgruver

Prompt-injeksjon er den viktigste. Når Claude leser skjermen, leser den alt som er synlig. En nettside med skjult tekst som forsøker å gi Claude instruksjoner utenom det du har bedt om, er en reell risiko. Anthropic scanner for slike forsøk. Det er ikke en garanti. Se hva Claude gjør, ikke bare hva det leverer.

Den nest viktigste er å gi Claude tilgang til for mye. Første gang Claude ber om tilgang til en app, er det fristende å bare si ja. Apper med bred tilgang til filsystem eller systeminnstillinger vises med en ekstra advarsel i tillatelsesdialogen. Les den.

I CLI

Computer Use i CLI krever per nå i research preview Claude Code v2.1.85 eller nyere og er foreløpig kun tilgjengelig på macOS. Det fungerer ikke i ikke-interaktiv modus, hvilket vil si muligheten til å brukes i automatiserte scripts. Men dette kan endres fort.

Når Claude tar kontroll, skjules andre apper slik at Claude bare samhandler med det du har godkjent. Terminalvinduet forblir synlig og utelates fra skjermbilder. Trykk Esc for å avbryte umiddelbart. Appene vises igjen med en gang.

Øvelse 8: Computer Use med lokal fil og nettleser

1. Åpne Cowork med nordheim-prosjektet og gi Claude denne oppgaven:
2. "Les rapporten i nordheim-mappen og sjekk på ssb.no om det finnes relevante statistikker om møtekultur eller produktivitet i norske virksomheter. Legg til en kort faktaboks i rapporten med det du finner."
3. Observer hva Claude gjør. Bruker den Chrome, eller tar den kontroll over skjermen? Når er det du selv ville ha gjort det annerledes?
4. Trykk Esc hvis noe ikke ser riktig ut.

Husk dette



- Computer Use i CLI krever v2.1.85 og fungerer bare på macOS.
- Skjermkontroll er siste utvei.
- Nettlesere er kun for visning i CLI og Code.
- Prompt-injeksjon er en reell risiko. Trykk Esc for å avbryte.

Multiagenter

Kapittel 9

Sub-agents, del opp arbeidet

Nordheim-prosjektet handler om tre leveranser: rapport, presentasjon og oppfølgingsplan, alle fra det samme kildematerialet. Jeg har laget det slik at det kan ligne på en reell situasjon. Du kan gi Claude alle tre i én instruksjon og la den jobbe seg gjennom dem i rekkefølge. Det fungerer. Men rapport og presentasjon er uavhengige av hverandre. De trenger ikke vente på hverandre.

Det er der sub-agenter kommer inn.

Hva sub-agenter er

En sub-agent er Claude i et eget kontekstvindu med en avgrenset oppgave. Mens du snakker med Claude i hovedsamtalen, kan den spinne opp sub-agenter som jobber parallelt med spesifikke deloppgaver. Resultatene kommer tilbake til hovedsamtalen når de er ferdige.

Det gir to fordeler. Parallellitet: to sub-agenter halverer ventetiden for oppgaver som ikke avhenger av hverandre. Kontekst: en lang samtale der Claude leser kildefiler, skriver sammendrag og produserer presentasjon fyller opp kontekstvinduet fort. Sub-agenter holder deloppgavene i egne vinduer og sender bare det som er relevant tilbake.

I Cowork og Code

Cowork koordinerer sub-agenter automatisk. Du trenger ikke sette dem opp eller navngi dem. Når du gir Claude en kompleks oppgave, vurderer den selv om parallelle sub-agenter er hensiktsmessig. Du ser det i fremdriftsindikatoren mens oppgaven kjører.

Code har tre innebygde sub-agenter den alltid har tilgang til. Explore brukes til filsök og kartlegging. Plan brukes til å forstå kontekst før endringer. Den tredje er en generell sub-agent for sammensatte oppgaver. Cowork bruker de samme under panseret.

I CLI

CLI gir full kontroll. Skriv **/agents** for å åpne agentpanelet. Running-fanen viser sub-agenter som kjører akkurat nå. Du kan åpne en aktiv agent og se hva den gjør, eller stoppe den. Library-fanen viser alle tilgjengelige agenter: innebygde, egne og plugin-baserte.

Egne sub-agenter defineres som Markdown-filer i `~/claude/agents/` for global tilgang eller `prosjektnavn/claude/agents/` for prosjekt-spesifikk tilgang. En enkel agent for Nordheim-prosjektet kan være dette:

name: nordheim-sammendrag

description: Leser kildefiler fra Nordheim AS og produserer strukturerte sammendrag. Bruk når kildefiler skal analyseres.

tools: Read, Write

model: sonnet

Du er en analytiker som leser dokumenter og produserer presise sammendrag. Fokuser på beslutninger tatt, uavklarte spørsmål og fordeling av ansvar.

Lagre filen som nordheim-sammendrag.md i `.claude/agents/`. Claude delegerer til den automatisk når en oppgave passer beskrivelsen, eller du kaller den eksplisitt: "Bruk nordheim-sammendrag-agenten på møtereferatet.". Anthropic anbefaler 2-5 agenter på middels til store oppgaver.

Øvelse 9: Send to oppgaver parallelt i Cowork

1. Åpne Cowork med nordheim-prosjektet og skriv:
2. "Les de fire kildefilene i nordheim-mappen. Produser rapport.md med en strukturert analyse og anbefalinger. Produser samtidig presentasjon.md med fem nøkkelpunkter egnet for en presentasjon. Leveransene skal være uavhengige av hverandre."
3. Se på fremdriftsindikatoren. Kjører Claude to arbeidsstrømmer parallelt? Sjekk begge filene når de er ferdige. Er innholdet konsistent på tvers, men tilpasset hvert format?

Husk dette



- Cowork koordinerer sub-agenter automatisk uten konfigurasjon.
- I CLI gir `/agents` full kontroll.
- Egne sub-agenter er Markdown-filer i `.claude/agents/`. Opptil 10 kan kjøre parallelt.

Multiagenter

Kapittel 10

Orkestrering uten å miste oversikten

To agenter som jobber parallelt er ikke dobbelt så effektivt som én. Det er dobbelt så mange ting som kan gå galt og dobbelt så mye output å vurdere. Det endrer ikke at multi-agent er nyttig. Det er bare et argument for å vite hva du faktisk holder på med.

Når multiagent er verdt det

Et to-agent-oppsett gir mening når oppgavene er uavhengige av hverandre og store nok til å fylle et kontekstvindu. Rapport og presentasjon fra Nordheim-materialet er et godt eksempel. De deler kildemateriale men trenger ikke koordinere underveis.

Det gir ikke mening når oppgavene er sekvensielle. Skal agent B bruke resultatet fra agent A, må A være ferdig først. Da er parallellitet ikke en fordel.

Start med én agent. Treffer du kontekstgrensen eller ser at deler av oppgaven er genuint uavhengige, legg til en agent til. Et fungerende to-agent-oppsett lærer deg mer enn et perfekt arkitekturert fem-agent-system som krasjer ved første kjøring.

Hva som kan gå galt

Den vanligste feilen er for bred instruksjon. En agent som skal "hjelp til med Nordheim-rapporten" vet ikke hva den skal gjøre. En agent som skal "lese møtereferatet og liste alle beslutninger med ansvarlig person" vet det. Jo smalere oppgaven er, jo mer forutsigbar er agenten.

Den nest vanligste er at en agent mister fokus underveis. Kontekstvinduet fylles, agenten kompenserer og mister tråden. Det merkes på outputen: vagere tekst, kortere sammendrag. Er du der, del oppgaven i to. Ikke gi agenten flere instruksjoner.

Oversikt i Cowork

Cowork viser fremdrift mens oppgaven kjører. Du kan følge med, gå fra maskinen og komme tilbake, eller avbryte. Vil du stoppe en oppgave, klikk treprikkmenyen ved siden av den og velg Avbryt. Sletting krever alltid eksplisitt godkjenning, uansett hva du har sagt tidligere i sesjonen.

Globale instruksjoner og mappeinstruksjoner gjelder også for agenter som kjører parallelt. Har du skrevet at Claude alltid skal lagre til nordheim-mappen, husker alle sub-agentene det

I CLI

Agent Teams er eksperimentelt og deaktivert som standard. Aktiver det ved å legge til `CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS=1` i settings.json. Du trenger Claude Code v2.1.32 eller nyere.

Én sesjon er teamleder. Den koordinerer via en delt oppgaveliste og spawner teammedlemmer som jobber i egne kontekstvinduer. Forskjellen fra sub-agenter er at teammedlemmene kan kommunisere direkte med hverandre. Bruk Shift+Down for å veksle mellom dem.

Agent Teams bruker betydelig mer tokens enn sub-agenter. Det er ikke et naturlig neste steg fra sub-agenter, men et verktøy for de tilfellene der teammedlemmene faktisk trenger å samkjøre underveis.

Øvelse 10: Kjør oppfølgingsplanen som én agent

1. Rapport og presentasjon ligger i nordheim-mappen fra forrige øvelse. Åpne Cowork og skriv:
2. "Les rapport.md og presentasjon.md i nordheim-mappen. Lag en oppfølgingsplan basert på anbefalingene i rapporten og nøkkelpunktene i presentasjonen. Lagre den som oppfølgingsplan.md."
3. Observer at Claude arbeider sekvensielt. Sammenlign med øvelse 9. Var én agent riktig valg her?

Husk dette



- Multi-agent lønner seg bare når oppgavene er genuint uavhengige. Sekvensielle oppgaver er ikke parallelle.
- Cowork holder oversikten automatisk og krever alltid godkjenning for sletting.
- Agent Teams i CLI er eksperimentelt og for oppgaver der teammedlemmer trenger direkte koordinering..

Ressursstyring

Kapittel 11

Token Economy i autonome sesjoner

Hvor mange ganger har du startet en lang sesjon uten å tenke på hva den koster? Lange sesjoner med sub-agenter bruker mye av dagskvota. Det ser du ikke med mindre du ser etter. Sesjonen kjører, resultatet kommer, og etterpå oppdager du at det meste er brukt opp.

Bok 2 introduserte `/context` og `/compact`. De er fortsatt gyldige. Men autonome sesjoner med parallelle agenter er en annen situasjon enn en vanlig samtale.

Hvorfor autonome sesjoner er dyrere

En vanlig samtale sender meldingene dine og Claudes svar frem og tilbake. En autonom sesjon sender hele samtalehistorikken på nytt ved hvert trinn. Etter 50 meldinger leser Claude 50 meldinger for å svare på den 51. Det vokser ikke lineært.

Sub-agenter gjør det verre. Hvert kontekstvindu har overhead i tillegg til selve oppgaven. Et oppsett med tre parallelle agenter bruker ikke tre ganger mer tokens, det bruker omtrent syv ganger mer.

Pauser koster også. Lar du en sesjon ligge i mer enn en time og går tilbake, resender Claude hele samtalen som ny cache til full pris. Vil du ta en lang pause, komprimer konteksten og start en ny sesjon.

Følge med underveis

Cowork viser ikke tokenforbruk direkte. Du må selv avbryte ved naturlige bruddpunkter og stole på at Claude komprimerer automatisk.

I Code bruker du `/context` og `/stats`. Context viser hvor mye av kontekstvinduet som er brukt, Stats viser forbruk for sesjonen. Se på disse jevnlig under lange oppgaver. I CLI gir `/usage` samme oversikt.

Komprimere riktig

`/compact` med instruksjon bevarer det som faktisk betyr noe. `/compact` Bevar alle beslutninger tatt og filer endret er bedre enn å kjøre kommandoen uten argument og håpe at Claude prioriterer det samme som deg.

`/clear` brukes når du skifter til en helt annen oppgave. Gammel kontekst fra noe urelatert koster tokens på hvert trinn i det nye.

60-prosent-regelen

Bok 2 nevnte at kvaliteten daler når kontekstvinduet er over to tredjedeler fullt. I autonome sesjoner er det vanskeligere å oppdage fordi Claude ikke stopper og spør. Ser du vagere svar enn oppgaven tilsier, kortere sammendrag eller feil referanser til noe fra tidligere, er det et signal om å komprimere.

Modellvalg

Opus tømmer kvotaen raskere enn Sonnet på Pro og Max. For langvarige autonome sesjoner med mange iterasjoner er Sonnet bedre egnet for de fleste trinnene. Bytt til Opus fra oppgaven faktisk krever tyngre resonnering, og tilbake for resten.

Øvelse 11: Sjekk forbruket

1. Åpne Code etter en av øvelsene fra de forrige kapitlene.
2. Bruk **/context** og **/stats** og les av forbruket.
3. Kjør deretter **/compact** Bevar alle leveranser og beslutninger knyttet til Nordheim-prosjektet.
4. Start en ny sesjon og skriv: "Hva er status på Nordheim-prosjektet?" Stemmer svaret med det du vet? Klarte instruksjonen å bevare det som var viktig?

Husk dette



- Cowork viser ikke tokenforbruk.
- Bruk **/context** og **/stats** i Code eller **/usage** i CLI.
- Pauser på over en time resetter cacheprising.
- Komprimer med **/compact** ved naturlige bruddpunkter.
- Bytt sesjon med **/clear** ved oppgaveskifter.

Ressursstyring

Kapittel 12

Tillatelsesprofiler og trygge grenser

Hvor mange filer ligger i mappen du ga Claude tilgang til?

Ikke bare prosjektfilene. Gamle skattemeldinger, private dokumenter, ting du ikke tenkte på da du valgte mappen. Det er ikke et spørsmål om autonomi. Det er et spørsmål om tilgang.

Folk gir ikke Claude for mye autonomi. De gir Claude tilgang til for mye.

Tillatelsesprofiler i CC desktop og CLI

`/permissions` åpner et interaktivt grensesnitt som viser alle aktive regler og hvilken fil de kommer fra. Du kan legge til og fjerne regler uten å starte sesjonen på nytt.

For prosjektspesifikk konfigurasjon, opprett `.claude/settings.json` i prosjektmappen. Reglene evalueres i fast rekkefølge: deny vinner alltid. En fornuftig grunnkonfigurasjon:

```
json
{
  "permissions": {
    "allowedTools": ["Read", "Write", "Edit"],
    "deny": ["Bash(rm *)", "Bash(sudo *)"]
  }
}
```

Dette lar Claude lese og redigere filer, men blokkerer sletting og kommandoer med forhøyede rettigheter. Vil du blokkere lesing av spesifikke filer, legger du til `"Read(.env*)"` i deny-listen.

For sesjonsbasert kontroll uten å redigere konfigurasjonsfiler:

```
claude --allowedTools "Read,Write,Edit" --disallowedTools "Bash(rm *)"
```

`--disallowedTools` fungerer i alle moduser, inkludert `bypassPermissions`. `--allowedTools` --fungerer ikke som forventet i `bypassPermissions`-modus. Hvis du bruker `bypass`, må du bruke `--disallowedTools` for å blokkere det du ikke vil at Claude skal gjøre.

Sandkassen i Cowork

Cowork opererer i en virtuell sandkasse. Claude ser ikke harddisken din. Den ser bare det du eksplisitt har lagt til i arbeidsområdet. Legger du til én mappe, er det den mappen Claude kan lese fra og skrive til.

Opprett en dedikert arbeidsmappe. Kall den `claude-workbench` eller noe tilsvarende. Kopier dit det du vil at Claude skal jobbe med. Gi aldri tilgang til Documents, Desktop eller hjemmemappen direkte.

Cowork har to godkjenningsmoduser: "Ask before acting" og "Act without asking", beskrevet i kap. 1. Uavhengig av modus krever sletting alltid eksplisitt godkjenning. Claude kan ikke slette en fil uten at du trykker Tillat i en dialog.

NB! Det Claude ikke kan se

Systempassord og nøkkelringer er utenfor rekkevidde. Claude kan ikke lese nettleserens lagrede passord eller macOS Keychain. Den kan ikke installere programvare på operativsystemet eller endre systeminnstillinger.

Øvelse 12: Konfigurer tillatelsesprofil for nordheim

1. Opprett `.claude/settings.json` i nordheim-mappen med innholdet over.
2. Åpne Code og kjør `/permissions`. Verifiser at reglene er aktive.
3. Gi Claude denne oppgaven: "Slett rapport.md og erstatt den med en ny versjon."
4. Ble slettingen blokkert? Ble erstatningen tillatt?

Husk dette



- Gi Claude tilgang til én avgrenset arbeidsmappe.
- Coworks sandkasse isolerer Claude til det du eksplisitt deler.
- Sletting krever alltid godkjenning.
- Deny-regler evalueres alltid først. Bruk `--disallowedTools` for eksklusjoner du er avhengig av.

Ressursstyring

Kapittel 13

Sikkerhet og risiko

To dager etter at Cowork ble lansert i januar 2026, demonstrerte sikkerhetsfirmaet PromptArmor at et Word-dokument med usynlig hvit tekst i ett punkts størrelse kunne lure Cowork til å laste opp filer fra maskinen din til en angreppers konto. Anthropic hadde kjent til sårbarheten siden oktober, nesten tre måneder tidligere.

Det er ikke unikt for Anthropic. Det er unikt for agenter. Et verktøy som kan handle, kan handle feil.

Risikoen endrer karakter

En chatbot som skriver noe galt kan rettes. En agent som gjør noe galt kan ha slettet en fil, sendt noe som ikke skulle sendes, eller lastet opp data ingen skulle se. Outputen er ikke lenger tekst. Det er handlinger.

Risikoen lever tre steder. Inputlaget er dokumentene og nettsidene Claude leser. Inneholder de skjulte instruksjoner, kan Claude bli lurt til å handle på dem uten at du ser det. Verktøylaget er alt Claude har tilgang til. Jo mer tilgang, jo større konsekvens når noe går galt. Handlingslaget er det som faktisk skjer mens du ikke ser på.

Plugins og marketplaces er en egen angrepsflate. En ondsinnet plugin kan endre tillatelser og omgå godkjenning gjennom såkalte hooks.

Prompt-injeksjon

Kap. 8 introduserte prompt-injeksjon i sammenheng med Computer Use. Her er det bredere bildet.

En nettside med skjult tekst kan gi Claude instruksjoner du ikke har bedt om. Et PDF-vedlegg kan inneholde det samme. Anthropic oppgir rundt én prosent angrepsrate på Claude i Chrome etter at mottiltak er innebygd. Ganget med mange nok sesjoner er det ikke en liten risiko.

Prompt-injeksjon er ikke begrenset til nettsider og dokumenter. Plugins og marketplaces introduserer en egen angrepsflate. En ondsinnet plugin kan bruke hooks til å endre tillatelser, omgå godkjenning og få Claude til å samle og sende data til eksterne tjenester

Tommelfingerregelen: Claude skal bare behandle filer og nettsider du stoler på. Ikke la Claude sammenfatte et vedlegg du ikke ville ha åpnet selv

Tre former for ukontrollert vekst

Det er tre kategorier som vokser stille i organisasjoner. Shadow AI er uautorisert bruk folk ikke forteller om. Agent sprawl er godkjente agenter ingen lenger har full oversikt over. Dark agent sprawl er agenter ingen vet om i det hele tatt.

For en enkeltperson er dette abstrakt. For en lærer som bruker Cowork på et skolenettverk, eller en konsulent som behandler klientfiler, er det ikke det.

Hva du faktisk kan gjøre

Start med minste privilegium. Gi Claude tilgang til én avgrenset mappe med bare de filene den trenger for oppgaven. Ikke en hel prosjektmappe der det også ligger e-posteksport og sensitive notater.

Krev godkjenning for irreversible handlinger. Act without asking er nyttig for lesing og skrivning av utkast. Det er et dårlig valg for oppgaver der noe slettes, sendes eller publiseres.

Behandle plugins og MCP-servere som programvare. En plugin fra en ukjent kilde med bred systemtilgang er ikke en produktivitetsforbedring. Hold øye med handlingsloggen. Cowork-historikken viser alt Claude har gjort i en sesjon. Er det noe der du ikke kjenner igjen, er det verdt å undersøke.

Øvelse 13: Vurder din egen risikoprofil

1. Åpne Cowork og se på hvilke mapper du har gitt tilgang til. Er det noe der du ikke ville ha delt med en fremmed? Fjern det.
2. Se på hvilke plugins og MCP-servere som er installert. Kjenner du alle? Vet du hva de har tilgang til?
3. Les gjennom handlingsloggen fra en av de tidligere øvelsene. Stemmer alt med det du ba Claude om?

Husk dette



- Risikoen med agenter er handlinger, ikke tekst.
- Prompt-injeksjon har ingen komplett løsning.
- Gi Claude tilgang til minst mulig. Krev godkjenning for det som ikke kan angres.
- Les handlingsloggen etter lange sesjoner.

Avslutning

Tillit er ikke fravær av kontroll

Nordheim AS er ferdig. Rapport, presentasjon og oppfølgingsplan ligger i mappen. Claude produserte dem. Du leverte dem.

Vet du nok om hva som skjedde mellom instruksjonen og resultatet til å stå inne for det?

Autonomi betyr ikke at du gir fra deg ansvaret. Det betyr at du har tenkt gjennom hva Claude kan avgjøre på egenhånd og hva som alltid skal gjennom deg. Den grensen er ikke gitt av verktøyet. Den er noe du setter, justerer og setter på nytt etter hvert som du lærer hva Claude faktisk gjør når du ikke ser på.

Prompt-injeksjon utnytter ikke dumme modeller. Den utnytter uklare instruksjoner og for brede tilganger. Tokenkostnader eksploderer ikke fordi Claude er treg. De eksploderer fordi ingen satte grenser for hva som skulle skje mens ingen så på. Det er ikke tekniske problemer. Det er manglende tenkning i forkant.

Kontrollen du gir fra deg er ikke borte. Den er delegert. Det fungerer bare hvis du vet hva du delegerer.

Claude kjører en ny Nordheim-oppgave klokka 07:15. Rapporten er klar til frokostmøtet.

Appendiks A: Kildemateriale for øvelsene

Dokument 1: Møtereferat

Møtereferat Nordheim AS, allmannamøte Dato: 12. februar 2026 Sted: Møterom 2, Nordheim AS Til stede: Lena Dahl (daglig leder), Kristoffer Moen (prosjektleder), Silje Bakke (økonomi), Tariq Hussain (marked), Marte Vik (HR), Jonas Strand (IT)

Lena åpnet møtet med å oppsummere tilbakemeldingene fra medarbeiderundersøkelsen som ble gjennomført i januar. Tre funn gikk igjen: møter tar for lang tid, det er uklart hvem som har ansvar for hva etter møtene, og mange opplever at beslutninger ikke følges opp.

Kristoffer presenterte et forslag han hadde utarbeidet i forkant: en ny møtestruktur basert på en workshop han deltok på i Oslo i januar. Kjernen i forslaget er tre grep. Alle møter skal ha en fast agenda sendt ut senest dagen før. Hvert møte avsluttes med en oppsummering der ansvarlige og frister er navngitt. Oppfølgingspunkter loggføres i et felles dokument alle har tilgang til.

Diskusjon: Tariq mente forslaget ville fungere for planlagte møter, men at det ikke passet for de kortere uformelle avklaringene som skjer daglig. Marte støttet Kristoffers forslag og la til at HR allerede bruker en lignende struktur internt. Jonas påpekte at det felles dokumentet måtte ligge et sted alle faktisk sjekker, ikke i enda et nytt system.

Beslutning: Nordheim AS innfører ny møtestruktur fra 1. mars 2026. Kristoffer har ansvar for å utarbeide en enkel mal for agenda og oppfølgingspunkter innen 20. februar. Jonas setter opp et delt dokument i det eksisterende systemet innen samme dato. Alle ledere er ansvarlige for at strukturen følges i møter de leder.

Neste statusgjennomgang: allmannamøte 9. april 2026.

Dokument 2: Statusrapport

Nordheim AS Ny møtestruktur: status etter fire uker Dato: 2. april 2026 Skrevet av: Kristoffer Moen

Det er nå fire uker siden ny møtestruktur ble innført. Denne rapporten oppsummerer hva som fungerer, hva som ikke fungerer og hva som må avklares før allmannamøtet 9. april.

Hva som fungerer

Agendaer sendes ut i forkant av de fleste planlagte møter. I en uformell runde blant deltakerne opplever de fleste at møtene er blitt kortere og mer fokuserte. Marte melder at HR-avdelingen har hatt god nytte av strukturen og at den er blitt en fast vane der.

Det delte oppfølgingsdokumentet brukes aktivt av prosjektavdelingen og HR. Jonas har fått positive tilbakemeldinger på at dokumentet ligger i det eksisterende systemet.

Hva som ikke fungerer

Markedsavdelingen har i liten grad tatt i bruk den nye strukturen. Tariq har ikke sendt ut agenda i forkant av noen av sine møter i perioden. Oppfølgingspunkter fra møter han leder er ikke loggført i det felles dokumentet.

To av de kortere uformelle avklaringsmøtene som holdes daglig er ikke inkludert i strukturen, noe som var forventet. Men grensen mellom hva som er et "møte" og hva som er en uformell prat er uklar for flere ansatte.

Uavklarte spørsmål

Skal strukturen gjelde alle møtetyper, eller bare planlagte møter over en viss lengde? Hvem har ansvar for å minne ledere på strukturen hvis den ikke følges? Er det behov for en kort oppfriskningsrunde for alle ansatte?

Kristoffer anbefaler at disse spørsmålene tas opp på allmannamøtet 9. april.

Dokument 3: E-posttråd

Emne: Møtestrukturen, fungerer den egentlig? Dato: 28. mars 2026

Fra: Tariq Hussain Til: Lena Dahl, Kristoffer Moen Dato: 28. mars, 09.14

Lena, Kristoffer,

Jeg vil ta opp noe før statusmøtet i april. Jeg er ikke overbevist om at den nye strukturen passer for alle typer møter. I marked jobber vi annerledes enn prosjekt og HR. Mye av det vi gjør er korte avklaringer og raske beslutninger. Å skrive agenda og loggføre oppfølgingspunkter for en ti minutters samtale føles som mer administrasjon, ikke mindre.

Jeg er ikke imot strukturen generelt, men jeg mener vi trenger et unntak for møter under 30 minutter eller møter med færre enn tre deltakere.

Tariq

Fra: Marte Vik Til: Lena Dahl, Kristoffer Moen, Tariq Hussain Dato: 28. mars, 10.47

Jeg forstår Tariqs poeng, men jeg er uenig i løsningen. Hvis vi lager unntak nå, fire uker inn, risikerer vi at strukturen gradvis visner. HR har brukt den på alle møter, også de korte, og det fungerer. Det handler mest om vane. De første ukene er tyngre, men det går over.

Det jeg derimot tror hadde hjulpet er en kortere mal for de raske møtene. Tre linjer, ikke en hel agenda. Kanskje det er der vi bør sette inn støtet.

Marte

Fra: Jonas Strand Til: Lena Dahl, Kristoffer Moen, Tariq Hussain, Marte Vik Dato: 28. mars, 13.22

Jeg er enig med Marte i at unntak er risikabelt, men jeg skjønner også Tariq. Det praktiske problemet er at folk ikke husker å åpne oppfølgingsdokumentet mellom møtene. Jeg har sett på om vi kan sette opp en automatisk påminnelse, men det krever at noen faktisk fyller inn noe å minne om.

Forslag: vi tester en forenklet variant for møter under 20 minutter de neste to ukene og evaluerer på allmannamøtet. Da har vi faktiske data å diskutere, ikke bare meninger.

Jonas

Fra: Lena Dahl Til: Kristoffer Moen, Tariq Hussain, Marte Vik, Jonas Strand Dato: 28. mars, 15.03

Takk for innspillene. Jonas sitt forslag er fornuftig. Kristoffer, kan du lage en forenklet mal innen 3. april? Vi tar en samlet vurdering på allmannamøtet.

Lena

Dokument 4: Notat med uløste spørsmål

Nordheim AS Forberedelse til allmannamøte 9. april 2026 Skrevet av: Kristoffer Moen Dato: 4. april 2026

Dette notatet oppsummerer hva som må besluttes på allmannamøtet 9. april. Det er ikke en statusrapport, men en liste over åpne spørsmål som trenger en avklaring fra ledelsen.

Spørsmål 1: Scope Skal den nye møtestrukturen gjelde alle møtetyper, eller innfører vi et skille mellom planlagte møter og korte avklaringer? Jonas og Marte har testet en forenklet mal for møter under 20 minutter siden 31. mars. Tilbakemeldingene så langt er blandede.

Spørsmål 2: Ansvar Hvem minner ledere på strukturen hvis den ikke følges? Per i dag er det ingen formell mekanisme. Kristoffer har gjort det uformelt, men det er ikke en rolle han er tildelt.

Spørsmål 3: Opplæring Er det behov for en kort oppfriskningsrunde for alle ansatte? Markedsavdelingen har i liten grad brukt strukturen. En felles gjennomgang på 20 minutter kunne avklart forventningene.

Spørsmål 4: Evaluering Når evaluerer vi om strukturen fungerer godt nok til å beholdes? Ingen dato er satt. Kristoffer foreslår en formell evaluering i juni, to måneder etter innføringen.

Hva møtet må lande på En beslutning på hvert av de fire punktene over. Kristoffer sender ut agenda innen 7. april.

Appendiks B: Hurtigreferanse

COWORK

- Act without asking — Godkjenningsmodusen som lar Claude jobbe uten å stoppe og spørre. Ligger under samtaleboksen ved siden av filer og modellvelger.
- Adaptive Thinking — Toggle i modellvelgeren som lar Claude vurdere innsatsbehovet selv. Gjelder alle modeller.
- Dispatch — Kobler mobilappen til Cowork via QR-kode. Finn den i venstremenyen.
- Live Artifacts — Dashboards som henter ferske data ved åpning. Finn dem i venstremenyen.
- Scheduled Tasks — Planlagte oppgaver som kjører mens maskinen er åpen. Start med /schedule eller klikk "Scheduled" i venstremenyen.
- Projects — Dedikerte arbeidsrom med innebygd hukommelse. Klikk Projects i venstremenyen, deretter +.
- Global instructions — Faste instruksjoner som gjelder alle Cowork-sesjoner. Finn dem under Settings > Cowork > Global instructions.

CODE

- Mode-velger — Velg mellom Plan og Accept Edits under samtaleboksen.
- Context — Viser hvor mye av kontekstvinduet som er brukt. Finn den i sidefeltet.
- Stats — Viser tokenforbruk for sesjonen. Finn den i sidefeltet.
- Scheduled Tasks — Planlagte lokale oppgaver. Klikk Schedule i sidefeltet, deretter "New task" og "New local task".
- Routines — Skybaserte oppgaver som kjører uten åpen maskin. Klikk "New task" og velg "New remote task", eller gå til `claude.ai/code/routines`.

TERMINAL

- /effort — Åpner interaktiv slider for innsatsnivå. Alternativt /effort low, /effort medium, /effort high, /effort xhigh eller /max.
- --effort — Setter innsatsnivå ved oppstart. Verdier: low, medium, high, xhigh, max.
- --continue / -c — Gjenopptar siste sesjon i gjeldende mappe.
- --resume / -r — Åpner velger med alle tidligere sesjoner.
- /recap — Gir situasjonsoppdatering inne i en aktiv sesjon.
- /usage — Viser tokenforbruk for aktiv sesjon.
- /compact — Komprimerer samtalehistorikken. Bruk med instruksjon: /compact Bevar alle beslutninger tatt.
- /clear — Sletter konteksten og starter frisk.
- /agents — Åpner agentpanelet med Running og Library-faner.
- /permissions — Åpner interaktivt grensesnitt for å se og administrere tillatelsesregler.
- /schedule — Oppretter en planlagt Routine direkte fra CL.
- /loop — Repeterer en oppgave med angitt intervall mens sesjonen er åpen. Eksempel: /loop 5m sjekk status.
- /remote-control — Starter Remote Control-sesjon og gir deg URL og QR-kode.
- --channels plugin:telegram@claude-plugins-official — Starter Channels-integrasjon med Telegram. Krever Bun og Claude Code v2.1.80 eller nyere.
- --dangerously-skip-permissions — Kjører uten tillatelsessjekker. Kun i isolerte miljøer.
- --allowedTools — Begrenser hvilke verktøy Claude kan bruke i sesjonen.
- --disallowedTools — Blokkerer spesifikke verktøy. Fungerer i alle moduser inkludert bypassPermissions.
- CLAUDE_CODE_EXPERIMENTAL_AGENT_TEAMS=1 — Aktiverer Agent Teams i settings.json. Krever Claude Code v2.1.32 eller nyere.

OPPSETT

- .claude/settings.json — Prosjektspesifikk konfigurasjon for tillatelser, modell og verktøy.
- ~/.claude/settings.json — Global konfigurasjon som gjelder alle prosjekter.
- .claude/agents/ — Mappe for prosjektspesifikke sub-agenter definert som Markdown-filer.
- ~/.claude/agents/ — Mappe for globale sub-agenter tilgjengelig på tvers av alle prosjekter.
- CLAUDE.md — Prosjektinstruksjoner Claude laster automatisk ved sesjonstart.

Appendiks C: Sjekkliste før du skruer på Act without asking

Bruk denne før du gir Claude en autonom oppgave med Act without asking aktivert.

Tilgang

- Claude har bare tilgang til mappen eller filene denne oppgaven faktisk trenger.
- Ingen sensitive filer ligger i samme mappe, skattemeldinger, passord, klientdata eller lignende.
- Du har ikke gitt tilgang til Documents, Desktop eller hjemmemappen direkte.

Oppgaven

- Instruksjonen er presis nok til at Claude vet hva den skal gjøre uten å gjette.
- Du vet hva et godt resultat ser ut som og kan vurdere det når Claude er ferdig.
- Oppgaven inneholder ingen handlinger som er vanskelige å angre: sletting, sending, publisering.

Omgivelsene

- Filene Claude skal behandle kommer fra kilder du stoler på.
- Du har ikke bedt Claude åpne vedlegg eller nettsider du ikke kjenner.
- Maskinen er i et nettverksmiljø du kontrollerer.

Underveis

- Du er tilgjengelig til å avbryte hvis noe ser galt ut.
- Du vet hvor du finner handlingsloggen og kan lese den etterpå.
- Hvis oppgaven skal kjøre uovervåket, har du satt opp tillatelsesprofil i settings.json som begrenser hva Claude kan gjøre.

Etter

- Resultatet stemmer med det du ba om.
- Handlingsloggen inneholder ingen handlinger du ikke kjenner igjen.
- Ingen filer er endret eller slettet som ikke skulle være det.

Appendiks D: Ordliste

Act without asking — Godkjenningsmodusen i Cowork som lar Claude utføre handlinger uten å stoppe og be om tillatelse for hvert steg. Bruker en klassifikator som skiller trygge handlinger fra handlinger som krever godkjenning.

Adaptive Thinking — Funksjon i Cowork som lar Claude selv vurdere om en oppgave krever utvidet resonnering. Aktiveres som en toggle i modellvelgeren.

Agent sprawl — Ukontrollert vekst i antall agenter i en organisasjon. Godkjente agenter ingen lenger har full oversikt over hva gjør.

Agent Teams — Eksperimentell funksjon i CLI der flere Code-instanser koordinerer som et team. Én instans er teamleder, resten er teammedlemmer som kan kommunisere direkte med hverandre.

Auto mode — Tillatelsesmodus i Code (aktiveres i innstillingene) som bruker en klassifikator til å avgjøre hvilke handlinger som er trygge å kjøre uten godkjenning. Tilsvarende konseptuelt Act without asking i Cowork.

Channels — Research preview-funksjon i CLI som kobler en kjørende Code-sesjon til Telegram, Discord eller iMessage via en MCP-plugin.

Computer Use — Funksjon som lar Claude se skjermen din og styre mus og tastatur for å navigere i apper og nettlese. Siste utvei i prioriteringskjeden etter direkte koblinger og nettlese navigering.

Dark agent sprawl — Agenter ingen i organisasjonen vet om. Enten satt opp utenfor godkjente systemer eller plantet med ondsinnede hensikter.

Dispatch — Funksjon i Cowork som kobler Claude-appen på telefonen til Cowork på maskinen via en persistent tråd. Gjør det mulig å starte og følge opp oppgaver fra mobilen.

Live Artifacts — Dashboards i Cowork som henter ferske data fra koblede tjenester hver gang du åpner dem. Ikke et engangsoutput, men et persistent arbeidsverktøy.

Permission relay — Funksjon i Channels som videresender godkjenningsforespørsler til telefonen i stedet for å stoppe sesjonen.

Prompt-injeksjon — Angrep der skjulte instruksjoner i dokumenter, nettsider eller e-poster forsøker å kapre Claude og få den til å utføre handlinger du ikke har bedt om.

Remote Control — Funksjon i CLI som gir deg et fullstendig claude.ai-grensesnitt inn i en kjørende Code-sesjon via URL eller QR-kode. Startkommando: `/remote-control`.

Routines — Planlagte oppgaver som kjører på Anthropic's skyinfrastruktur uavhengig av om maskinen er på. Tilgjengelig i Code og via `claude.ai/code/routines`. Krever Pro eller Max.

Scheduled Tasks — Planlagte oppgaver som kjører lokalt mens maskinen er åpen. Tilgjengelig i Cowork og Code.

Session Memory — Automatisk bakgrunnssystem i Code og CLI som lagrer strukturerte sammendrag fra sesjoner til disk og laster relevante minner ved sesjonstart.

Shadow AI — Uautorisert bruk av KI-verktøy av ansatte uten organisasjonens kjennskap eller godkjenning.

Sub-agent — En Claude-instans med eget kontekstvindu og avgrenset oppgave, spawnet av en hovedagent for å håndtere en deloppgave parallelt eller isolert.

Appendiks E: Brukseksempler i Cowork

Generelt kunnskapsarbeid

Morgenbrief

Bygg et Live Artifact kalt "Morgenbrief" som jeg åpner hver hverdag. Hent fra: - Google Kalender (dagens møter med deltakere og eventuelle agendadokumenter) - Gmail (uleste e-poster siste 18 timer, sortert i "svar i dag" og "til info") - Slack (omtaler og direktemeldinger siden sist) Layout: - Øverst: dato, totalt antall møtetimer og antall uleste meldinger per kilde - "Må besvares i dag": de 5 viktigste sakene på tvers av e-post og Slack, med én setnings sammendrag - Møteoversikt: hvert møte med ett forberedelsesnotat hentet fra lenket dokument eller tråd - "Lavere prioritet": alt annet, sammenfoldet som standard Design: rent, enkeltkolonne, lesbart på telefon. Kun rødt for haster. Lagre til Live Artifacts-fanen.

Mapperydding

Rydd mappen /Users/[brukernavn]/Temp. Følg denne strukturen:

1. Les alle filer i mappen og én mappenivå ned.
2. Finn duplikater. Behold nyeste versjon, flytt resten til "_duplikater".
3. Grupper i undermapper: Dokumenter, Bilder, Regneark, Installasjonsfiler og Annet.
4. Inne i hver undermappe, sorter etter år-måned.
5. Jobbrelevante filer fra siste 14 dager legges i "_nylig-jobb" øverst.
6. Filer eldre enn 12 måneder som ikke er åpnet på 6 måneder, legg i "_arkivkandidater". Ikke slett noe.

Vis meg planen før du flytter noe. Gi meg til slutt et sammendrag: antall filer, duplikater fjernet og de 10 eldste filene jeg bør vurdere å slette.

Forskningssammendrag

Jeg skriver et sammendrag om [tema]. Kildematerialet ligger i: /Users/[brukernavn]/Dokumenter/research/[mappenavn]

Les alt. Produser et dokument med:

1. Sammendrag (maks 150 ord): de 3 viktigste funnene
2. Hovedtemaer: 4-6 temaer som går igjen, med 2-3 setningers forklaring og hvilke kilder som støtter dem
3. Motsetninger: steder der kildene sier ulike ting, beskriv begge sider
4. Tall og data: kulepunkter med statistikk verdt å sitere, med kildehenvisning
5. Åpne spørsmål: 3-5 ting kildematerialet ikke svarer på
6. Kildeliste: filnavn, dokumenttittel og dato

Format: DOCX i samme mappe som kildematerialet. Faglig og presis tone.

Kvitteringsbehandling

Mappen /Users/[brukernavn]/Dokumenter/utgifter/[kvartal] inneholder kvitteringer som JPG, PNG, PDF og skjermbilder.

For hver fil, hent ut:

- Leverandør - Dato (ÅÅÅÅ-MM-DD)
- Totalbeløp med valuta
- Mva hvis oppgitt separat
- Betalingsmåte hvis synlig
- Kategori: Reise, Mat, Programvare, Kontorrekvisita, Maskinvare, Markedsføring, Profesjonelle tjenester eller Annet
- En setnings beskrivelse
- Originalt filnavn

Output: XLSX lagret til /Users/[brukernavn]/Dokumenter/utgifter/[kvartal]-sammendrag.xlsx

- En rad per kvittering, sortert etter dato
- "Sammendrag"-ark foran med totaler per kategori og per måned
- "Flagget"-ark for kvitteringer der beløp ikke kunne leses

Bruk Excel-formler for totaler. Kjør i "Spør før du handler"-modus.

Undervisning og skole

Ukentlig plan fra læreplanen

Lag en ukentlig undervisningsplan for [fag] på [trinn] for uke [nummer].

Klassen er her i progresjonen: [beskriv kort].

Kompetansemål i fokus: [lim inn fra læreplanen].

Tilgjengelig tid: [antall] timer fordelt på [antall] økter.

Særlige hensyn: [for eksempel elever med lesevansker, ekskursjon torsdag].

Planen skal inneholde:

- Læringsmål per økt - Forslag til aktiviteter og arbeidsformer
- Vurderingsmoment for uka
- Differensieringsforslag for elever som trenger mer støtte og elever som trenger mer utfordring

Lagre som ukesplan-uke[nummer].docx i [mappenavn].

Fra foreldremøtereferat til oppfølgingsplan

Les referatet i filen [filnavn].

Lag en oppfølgingsplan med:

- Alle saker som ble tatt opp
- Hvem som har ansvar for oppfølging - Foreslått frist for hver sak
- Saker som trenger avklaring før de kan følges opp

Lagre som oppfølgingsplan-[dato].docx.

Ressursinnsamling til et kompetansemål

Jeg skal undervise om [tema/kompetansemål] for [trinn].

Søk opp og sammenstill relevante ressurser. Jeg trenger:

- 3-5 fagartikler eller forskningsbaserte tekster egnet for lærerbakgrunn
- 3-5 elevrettede ressurser tilpasset [trinn]: nettsider, videoer, interaktivt innhold
- 2-3 eksempler på undervisningsopplegg andre lærere har delt
- Eventuelle relevante ressurser fra Udir, NDLA eller Utdanningsnytt

For hver ressurs, oppgi:

- Tittel og lenke - En setnings beskrivelse av hva den inneholder
- Hvorfor den er relevant for dette kompetansemålet
- Nivå: egnet for lærer, elev eller begge

Lagre oversikten som ressurser-[tema]-[dato].docx i [mappenavn].

Skriving og publisering

Fra rånotater til kronikk

Mappen [mappenavn] inneholder rånotater og kildelenker om [tema].

Les alt. Skriv et utkast til en kronikk på 600-800 ord:

- Åpner med en konkret scene eller observasjon, ikke en generell påstand
- Har ett tydelig argument som holder hele teksten
- Bruker minst ett konkret eksempel eller datapunkt fra kildematerialet
- Unngår akademisk språk og honnørord
- Avslutter uten å oppsummere

Tone: saklig og direkte. Tenk aviskronikk, ikke debattinnlegg. Lagre som kronikk-utkast-[dato].docx.

Ukentlig nyhetsbrev fra egne notater

Mappen [mappenavn] inneholder notater fra denne uka som tekst- og Markdown-filer.

Les alle filene. Skriv et nyhetsbrev på 400-600 ord:

- En åpningssetning som setter tonen
- 3-5 ting jeg har tenkt på, lest eller oppdaget, med en kort forklaring av hvorfor det betyr noe
- En avsluttende tanke eller spørsmål

Tone: som en e-post til en kollega du liker og respekterer. Lagre som nyhetsbrev-[dato].md.

Appendiks F: Kilder og ressurser

Offisiell dokumentasjon

- Claude Code-dokumentasjon — code.claude.com/docs
- Cowork-produktside — claude.com/product/cowork
- Claude desktop app — support.claude.com
- Sub-agents — code.claude.com/docs/en/sub-agents
- Agent Teams — code.claude.com/docs/en/agent-teams
- Tillatelseskonfigurasjon — code.claude.com/docs/en/permissions
- Sikkerhet — code.claude.com/docs/en/security
- Kostnadsstyring — code.claude.com/docs/en/costs
- Auto mode — anthropic.com/engineering/claude-code-auto-mode

Bruerveiledninger og guider

- Introduksjon til Cowork (offisielt kurs) — anthropic.skilljar.com/introduction-to-claude-cowork
- Introduksjon til sub-agents (offisielt kurs) — anthropic.skilljar.com/introduction-to-subagents
- Claude best practices 2026 — the-ai-corner.com
- Cowork starter guide med 30 eksempler — claudiaplusai.substack.com
- Lær 80 % av Cowork på 20 minutter — jeffsu.org
- Claude Code for produktledere — ccforpms.com/cowork
- Slik maksimerer du Cowork — towardsdatascience.com
- Multi-agent orkestrering i Claude Code — shipyard.build/blog/claude-code-multi-agent

Sikkerhet og risiko

- Sikker implementering av Cowork (PromptArmor) — promptarmor.com/resources/implement-claude-cowork-securely
- CISO-guide til Claude i bedriften — weekendbyte.com
- Sikkerhetspraktiker-guide til Cowork — harmonic.security
- Agent Risk, Security, and AI Sprawl in 2026 (Everyday AI, ep. 767) — youeverydayai.com
- Er Cowork trygt? — wonderingaboutai.substack.com

Tekniske referanser

- Claude Code changelog — claude.st/blog/guide/changelog
- Kostnader og tokenforbruk i 2026 — finout.io/blog/claude-code-pricing-2026
- Kontekstbuffer og komprimering — claude.st/blog/guide/mechanics/context-buffer-management
- Token-optimering — buildtolaunch.substack.com
- Channels-integrasjon — anthropic.com/engineering/claude-code-channels (se offisiell dokumentasjon)

Blogginlegg fra Anthropic

- Sub-agents i Claude Code — claude.com/blog/subagents-in-claude-code
- Når bruke multi-agent-systemer — claude.com/blog/building-multi-agent-systems-when-and-how-to-use-them

© 2026 KENNETH BAREKSTEN

Denne boka er lisensiert under Creative Commons (CC BY 4.0). Det betyr at du kan dele, bruke og tilpasse innholdet, også kommersielt, så lenge du krediterer forfatteren.

Kontakt: kenneth@laererliv.no

Nettside: laererliv.no

