$97 FULL SYSTEM

# The OpenClaw Complete System

Includes Quick Start + Advanced automations, all integrations, scaling strategies

For those ready to build serious AI automation

## 🎓 FOUNDATION COURSE (QUICK START INCLUDED)

Everything from the $39 Quick Start guide is included below. Start here if you're new to OpenClaw.

## Section 1: Installation & Basics

## 1.1 What You Need to Know

OpenClaw is software that lets you create AI agents - programs that do tasks automatically using artificial intelligence. Think of it like having a smart assistant that works 24/7.

### What is a Terminal?

A terminal is a text-based way to control your computer. Instead of clicking buttons, you type commands.

#### 1 Open Terminal on Mac

1. Press Command + Space (hold Command, tap Space)
2. Type: terminal
3. Press Enter
4. A black window opens - that's your terminal

#### 2 Open Terminal on Windows

1. Press the Windows key
2. Type: cmd
3. Press Enter
4. A black window opens - that's Command Prompt

### What is an API Key?

An API key is like a password that lets software access AI services. You need API keys from AI providers (like OpenAI, Claude) so OpenClaw can use their AI models.

> ⚠️ **SECURITY WARNING:** API keys are passwords. Never share them. Never post them online. If someone gets your key, they can use AI and charge your account.

## 1.2 Getting Your API Keys

### OpenAI API Key (GPT-4, GPT-3.5)

#### 1 Create Account

1. Go to: platform.openai.com
2. Click "Sign up"
3. Enter email and create password
4. Verify your email (check inbox)

#### 2 Add Payment

1. Click your profile (top right) → "Billing"
2. Click "Add payment method"

3. Enter credit card
4. Set usage limit to $20 (safety!)

### 3 Create API Key

1. Go to: platform.openai.com/api-keys
2. Click "Create new secret key"
3. Name it: OpenClaw
4. Click "Create secret key"
5. **COPY THE KEY NOW** (starts with sk-)
6. Paste it in a safe place (password manager)
7. Click "Done"

## Anthropic API Key (Claude)

### Get Claude API Key

1. Go to: console.anthropic.com
2. Sign up with email
3. Verify email
4. Go to "API Keys" in left menu
5. Click "Create Key"

6. Name it: OpenClaw
7. Copy the key (starts with sk-ant-)
8. Save securely

## 1.3 Installing Node.js

OpenClaw needs Node.js to run. Node.js is software that lets JavaScript programs run on your computer.

### 1 Check If Already Installed

1. Open your terminal (see Section 1.1)
2. Type exactly: node --version
3. Press Enter

If you see a version number like "v18.17.0" or higher, you have Node.js! Skip to Section 1.4.

If you see "command not found," continue below.

### Install on Mac

1. Go to: nodejs.org
2. Click the big green "LTS" button
3. Download the macOS Installer (.pkg)

4. Open Downloads folder
5. Double-click the downloaded file
6. Click "Continue" through all screens
7. Click "Install"
8. Enter your Mac password when asked
9. Wait for installation
10. Click "Close"

### Install on Windows

1. Go to: nodejs.org
2. Click the big green "LTS" button
3. Download the Windows Installer (.msi)
4. Open Downloads folder
5. Double-click the downloaded file
6. Click "Next" through all screens
7. Accept license agreement
8. Click "Install"
9. Click "Yes" if Windows asks for permission
10. Wait for installation
11. Click "Finish"

## 1.4 Installing OpenClaw

### 1 Install OpenClaw

1. Open your terminal
2. Type exactly: npm install -g openclaw
3. Press Enter
4. Wait for installation (may take 1-2 minutes)

### 2 Verify Installation

1. Type: openclaw --version
2. Press Enter

You should see a version number. OpenClaw is now installed!

## 1.5 Initial OpenClaw Setup

When you first run OpenClaw, it will walk you through setup. Here's what to expect:

### 1 Start OpenClaw

1. In terminal, type: openclaw
2. Press Enter

### 2 Select Your Skills

OpenClaw will show a list of skills. Use arrow keys to navigate, Space to select, Enter to confirm.

**For beginners, select:**

- ✓ filesystem (read/write files)
- ✓ web (access websites)
- ✓ memory (remember context)

### 3 Select Primary Model

Choose which AI to use by default:

- **GPT-4 (OpenAI):** Best quality, most expensive
- **GPT-3.5 (OpenAI):** Good quality, cheaper
- **Claude (Anthropic):** Great reasoning, mid-price

**Recommendation:** Start with GPT-3.5 (cheaper while learning)

# 1.6 Complete AI Model Guide (February 2026)

## OPENAI - Current Flagship Models

| Model | Best For | Context | Cost |
|---|---|---|---|
| **GPT-5.3-Codex** | Most capable agentic coding | 128K | $$$$ |
| **GPT-5.2** | Best overall, coding, agents | 128K | $$$ |
| **GPT-5.2 Pro** | Smarter, more precise responses | 128K | $$$$ |
| **GPT-5.2-Codex** | Long-horizon agentic coding | 128K | $$$ |
| **GPT-5.1** | Configurable reasoning effort | 128K | $$$ |
| **GPT-5** | Previous flagship | 128K | $$ |
| **GPT-5 Mini** | Fast, well-defined tasks | 128K | $ |
| **GPT-5 Nano** | Cheapest, fastest | 128K | $ |
| **GPT-4.1** | Smartest non-reasoning | 128K | $$ |
| **o3** | Complex reasoning tasks | 200K | $$$ |
| **o3 Deep Research** | Most powerful research | 200K | $$$$ |
| **o4 Mini** | Fast, cost-efficient reasoning | 200K | $$ |

## OPENAI - Specialized Models

| Model | Best For | Cost |
|---|---|---|

| | | |
|---|---|---|
| GPT Image 1.5 | State-of-the-art image generation | $$$ |
| Sora 2 | Video + audio generation | $$$$ |
| Sora 2 Pro | Most advanced video | $$$$$ |
| GPT Realtime 1.5 | Voice conversations | $$$ |
| GPT Audio 1.5 | Audio in/out | $$ |
| GPT-5.3-Codex | Ultimate coding model | $$$$ |

## ANTHROPIC CLAUDE - Current Generation

| Model | Best For | Context | Cost |
|---|---|---|---|
| **Claude 4 Opus** | Most capable, complex analysis | 200K | $$$$ |
| **Claude 4.6** | Enhanced reasoning, coding | 200K | $$$ |
| **Claude 4 Sonnet** | Best balance capability/speed | 200K | $$ |
| **Claude 4 Haiku** | Fast, cost-effective | 200K | $ |
| Claude 3 Opus | Long documents, complex instructions | 200K | $$$ |
| Claude 3 Sonnet | General tasks, good balance | 200K | $$ |
| Claude 3.5 Sonnet | Coding, analysis | 200K | $$ |

## GOOGLE GEMINI - Current Models

| Model | Best For | Context | Cost |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **Gemini 3.1 Pro** | Complex reasoning, coding | 1M | $$$ |
| **Gemini 3.1 Ultra** | Most capable Google model | 1M | $$$$ |
| **Gemini 3.1 Flash** | Fast, versatile tasks | 1M | $$ |
| **Gemini 3.1 Flash Image** | Image generation & editing | 64K | $$ |
| Gemini 2.5 Pro | Advanced reasoning | 1M | $$ |
| Gemini 2.5 Flash | Fast tasks | 1M | $ |

## MOONSHOT (KIMI)

| Model | Best For | Context | Cost |
|---|---|---|---|
| **Kimi K2.5** | Long context, Chinese/English | 256K | $$ |
| **Kimi K2.5 Pro** | Premium quality | 256K | $$$ |
| **Kimi K2** | General tasks | 200K | $$ |
| **Kimi K1.5** | Cost-effective | 128K | $ |

## MINIMAX

| Model | Best For | Context |
|---|---|---|
| **MiniMax-Text-01** | General text tasks | 200K |
| **MiniMax-VL-01** | Vision-language | 100K |
| **MiniMax-Video-01** | Video generation | N/A |

## OPENROUTER - Top Models (Access to 200+)

| Model | Provider | Best For | Cost |
|---|---|---|---|
| openai/gpt-5.3-codex | OpenAI | Ultimate coding | $$$$ |
| openai/gpt-5.2 | OpenAI | Best overall | $$$ |
| anthropic/claude-4-opus | Anthropic | Complex tasks | $$$$ |
| anthropic/claude-4.6 | Anthropic | Enhanced coding | $$$ |
| google/gemini-3.1-pro | Google | Long context | $$$ |
| google/gemini-3.1-ultra | Google | Most capable | $$$$ |
| meta-llama/llama-4-maverick | Meta | Open weights | $$$ |
| meta-llama/llama-4-scout | Meta | Efficient | $$ |
| deepseek/deepseek-v4 | DeepSeek | Coding, reasoning | $$ |
| deepseek/deepseek-r1 | DeepSeek | Reasoning | $$ |
| qwen/qwen3.5-122b | Alibaba | Vision+text | $$ |
| qwen/qwen3.5-flash | Alibaba | Fast multimodal | $ |
| bytedance-seed/seed-2.0 | ByteDance | Multimodal | $ |
| mistral/mistral-large-3 | Mistral | European | $$ |
| cohere/command-r-plus | Cohere | Enterprise RAG | $$ |
| perplexity/sonar-reasoning | Perplexity | Web search | $$ |
| xai/grok-2 | XAI | Real-time data | $$ |

## 1.7 Your First Agent

Now let's create a simple agent that actually does something useful.

### 1 Create Agent File

1. Open your text editor (TextEdit on Mac, Notepad on Windows)
2. Copy the code below
3. Save as: my-first-agent.md
4. Save location: Your Documents folder

```
# My First Agent ## Task Say hello and tell me the current date ## Instructions 1. Greet the user by saying "Hello! I'm your OpenClaw agent." 2. Tell them today's date 3. Ask how you can help ## Run this openclaw run my-first-agent.md
```

## ✅ Foundation Complete!

You've installed OpenClaw and created your first agent. Continue below for advanced integrations and automations.

# Section 3: All Messaging Integrations

## 1.1 Telegram Bot Setup

Connect OpenClaw to Telegram to create bots that can receive commands and send messages.

### Create Telegram Bot

1. Open Telegram app
2. Search for: @BotFather
3. Click "Start"
4. Type: /newbot
5. Give your bot a name
6. Give it a username (must end in "bot")
7. BotFather will give you a token like: 123456789:ABCdefGHIjklMNOpqrSTUvwxyz
8. Save this token securely

```
# Telegram Agent Configuration # Save as: telegram-agent.md ## Task Monitor Telegram messages and respond ## Instructions 1. Connect to Telegram API using bot token 2. Listen for incoming messages 3. For each message: - If message contains "status": Report system status - If message contains "run [task]": Execute the task - Otherwise: Log message for review 4. Send confirmation
```

```
back to Telegram ## Configuration
TELEGRAM_BOT_TOKEN=your_token_here ALLOWED_CHAT_IDS=your_chat_id
```

## 1.2 Discord Bot Integration

### Create Discord Bot

1. Go to: discord.com/developers/applications
2. Click "New Application"
3. Name it: OpenClawBot
4. Go to "Bot" in left menu
5. Click "Add Bot"
6. Click "Copy Token" (save this!)
7. Enable "Message Content Intent"
8. Save changes

```
# Discord Agent Example ## Task Monitor Discord server and
automate responses ## Instructions 1. Connect to Discord using
bot token 2. Listen for messages in configured channels 3. When
keywords detected: - "!status": Post system status - "!help":
List available commands - "!report": Generate daily report 4. Log
all interactions ## Setup Commands - Invite bot to server with:
oauth link - Configure permissions: Send Messages, Read History
```

## 1.3 Slack Integration

### Create Slack App

1. Go to: api.slack.com/apps
2. Click "Create New App"
3. Choose "From scratch"
4. Name: OpenClaw Agent
5. Select your workspace
6. Go to "OAuth & Permissions"
7. Add Bot Token Scopes:
    - chat:write
    - channels:read
    - groups:read
    - im:read
8. Click "Install to Workspace"
9. Copy "Bot User OAuth Token"

## 1.4 Email Integration (IMAP)

```
# Email Monitoring Agent ## Task Monitor email inbox and process messages ## Configuration Required: - IMAP_SERVER=imap.gmail.com - IMAP_PORT=993 - EMAIL_USERNAME=your@email.com - EMAIL_PASSWORD=app-specific-password - CHECK_INTERVAL=300 (seconds) ## Instructions 1. Connect to email via IMAP 2. Every 5 minutes: a. Check for new unread emails b. For each email: - Extract sender, subject, body - Categorize: urgent, normal, spam
```

```
- Draft response if needed - Mark as read c. Send summary to
owner ## Security Note: Use app-specific passwords, not your main
password
```

### 1.5 SMS Integration (Twilio)

#### Setup Twilio

1. Go to: twilio.com
2. Create account
3. Get a phone number
4. Copy Account SID and Auth Token

# Section 4: Advanced AI Models Deep Dive

### 4.1 Model Comparison Matrix

| Model | Provider | Strengths | Cost/Million Tokens | Best For |
|---|---|---|---|---|
| GPT-4 | OpenAI | Highest quality, reasoning, coding | $30 input / $60 output | Complex analysis, final drafts |

| | | | | |
|---|---|---|---|---|
| GPT-4 Turbo | OpenAI | 128k context, faster, cheaper | $10 input / $30 output | Long documents, most tasks |
| GPT-3.5 Turbo | OpenAI | Fast, cheap, good enough | $0.50 input / $1.50 output | Drafting, sorting, simple tasks |
| Claude 3 Opus | Anthropic | Best reasoning, very careful | $15 input / $75 output | Analysis, sensitive content |
| Claude 3 Sonnet | Anthropic | Good balance of speed/quality | $3 input / $15 output | General purpose workhorse |
| Claude 3 Haiku | Anthropic | Fastest, cheapest | $0.25 input / $1.25 output | Quick tasks, high volume |
| Gemini 1.5 Pro | Google | 1M token context, multilingual | $3.50 input / $10.50 output | Long docs, non-English |
| Gemini 1.5 Flash | Google | Fast, cheap, large context | $0.35 input / $1.05 output | High volume, speed needed |

## 4.2 Model Selection Strategy

> 💡 **The Tier Strategy:**
>
> - **Tier 1 (Cheap):** GPT-3.5 or Claude Haiku for drafting, sorting, simple tasks
> - **Tier 2 (Mid):** GPT-4 Turbo or Claude Sonnet for most production work

- **Tier 3 (Premium):** GPT-4 or Claude Opus only for final review, complex analysis

This approach saves 60-80% on costs while maintaining quality.

# Section 5: Coolest Use Cases

## 5.1 Autonomous Research Agent

```
# Research Agent - Competitive Intelligence ## Task Weekly competitive research report ## Instructions Every Monday at 9 AM: 1. Visit competitor websites (list of 10) 2. Check for: - New blog posts - Pricing changes - New features - Job postings (growth indicators) - Social media updates 3. Search news for: - Company name mentions - Industry trends - Funding announcements 4. Analyze findings: - What changed since last week? - What are they focusing on? - What threats/opportunities emerge? 5. Generate report: - Executive summary - Detailed findings - Recommended actions - Sources and links 6. Send via email to team
```

## 5.2 Customer Support Automation

```
# Advanced Customer Support Agent ## Task Full support automation with escalation ## Instructions LEVEL 1 - Auto-Response: - Password reset requests → Send reset link - Common questions → Send knowledge base link - Status inquiries → Check system, reply LEVEL 2 - Draft Response: - Technical issues → Draft troubleshooting steps - Billing questions → Draft with account lookup - Feature requests → Log + draft thank you LEVEL 3 - Human Required: - Angry customers (sentiment analysis) - Complex
```

```
technical issues - Custom pricing requests - Refund demands
ESCALATION RULES: - Reply time > 2 hours → Escalate - Sentiment
score < -0.5 → Escalate - Keywords: "cancel", "refund",
"lawsuit", "manager" → Escalate INTEGRATION: - Connect to
Zendesk/Freshdesk - Update ticket status - Add internal notes -
Route to correct team
```

## 5.3 Content Creation Pipeline

```
# Content Factory Agent ## Task End-to-end content creation ##
Instructions WEEKLY SCHEDULE: Monday - Research: - Scan industry
news - Check trending topics - Review competitor content -
Identify content gaps Tuesday - Plan: - Generate 10 article ideas
- Score each by: * Search volume potential * Competition level *
Business relevance - Select top 3 Wednesday - Create: - Draft
long-form articles (GPT-4) - Create social media versions (GPT-
3.5) - Generate images (via DALL-E API) - Make video scripts
Thursday - Optimize: - SEO optimization - Keyword insertion -
Meta descriptions - Internal linking Friday - Schedule: - Upload
to CMS - Schedule social posts - Set email campaign - Prepare
analytics tracking OUTPUTS: - 3 blog posts - 15 social media
posts - 1 email newsletter - 2 video scripts - Weekly performance
report
```

## 5.4 Lead Generation & Outreach

```
# Lead Generation Agent ## Task Find and qualify leads
automatically ## Instructions DAILY WORKFLOW: 1. Find Leads: -
Search LinkedIn for target titles - Check Crunchbase for funding
news - Monitor job postings (hiring = budget) - Track website
traffic (SimilarWeb) 2. Qualify: Score each lead 1-100 based on:
- Company size (1-25 pts) - Recent funding (1-25 pts) - Tech
stack match (1-25 pts) - Engagement signals (1-25 pts) 3. Enrich:
- Find email (Hunter.io, Apollo) - Find phone (Lusha) - Research
background - Check mutual connections 4. Personalize: - Draft
```

```
personalized email - Reference specific company news - Mention
relevant case study - Custom subject line 5. Outreach: - Send
email (via your email) - Connect on LinkedIn - Add to CRM - Set
follow-up reminder FILTERS: - Skip if: personal email, <10
employees, competitor - Prioritize if: recent funding, hiring,
tech match GOAL: 50 qualified leads/week, 20% response rate
```

### 5.5 Financial Monitoring & Alerts

```
# Financial Guardian Agent ## Task Monitor finances and alert on
anomalies ## Instructions DAILY CHECKS: 1. Bank Account
Monitoring: - Check balances - Flag unusual transactions - Watch
for duplicate charges - Monitor cash flow trends 2. Invoice
Tracking: - Check unpaid invoices - Send reminders (day 7, 14,
30) - Flag overdue payments - Calculate expected revenue 3.
Expense Analysis: - Categorize new expenses - Flag unusual
amounts - Check for subscriptions to cancel - Compare to budget
4. Alerts Send If: - Balance < $10,000 - Expense > $1,000 -
Invoice 30+ days overdue - Unusual transaction pattern -
Duplicate charge detected WEEKLY REPORT: - Cash position -
Outstanding invoices - Upcoming bills - Spending vs budget -
Recommendations INTEGRATION: - Plaid API (bank connections) -
Stripe API (payments) - QuickBooks/Xero (accounting) -
Email/Slack (alerts)
```

## Section 6: Advanced Agent Recipes

### 6.1 Multi-Step Research Agent

```
# Deep Research Agent ## Task Comprehensive topic research ##
Instructions INPUT: Research topic/query STEP 1 - Broad Search: -
Search web for top 20 resources - Identify key subtopics - Find
```

```
authoritative sources STEP 2 - Deep Dive: - Read top 10 articles
thoroughly - Extract key findings - Note contradictions -
Identify gaps STEP 3 - Synthesis: - Create outline of findings -
Organize by theme - Rate source credibility - Note confidence
levels STEP 4 - Analysis: - Identify trends - Note outliers -
Assess consensus vs controversy - Draw preliminary conclusions
STEP 5 - Output: - Executive summary - Detailed findings - Source
list with credibility scores - Gaps requiring more research -
Recommended actions QUALITY CHECKS: - Verify source credibility -
Cross-check facts - Flag unverified claims - Note uncertainty
levels OUTPUT FORMAT: Markdown report with citations
```

## 6.2 Meeting Assistant Agent

```
# Meeting Intelligence Agent ## Task Pre-meeting prep and post-
meeting follow-up ## PRE-MEETING (15 min before): 1. Research
Attendees: - LinkedIn profiles - Recent company news - Past
interactions - Shared connections 2. Meeting Prep: - Review
agenda - Prepare relevant case studies - Draft talking points -
Note questions to ask 3. Send Brief: - Attendee summaries -
Suggested talking points - Relevant materials - Pre-meeting
confidence boost ## DURING MEETING (if recording): 1. Real-time
Notes: - Key points discussed - Decisions made - Action items
mentioned - Questions raised 2. Live Assistance: - Pull up
relevant data when mentioned - Suggest follow-up questions - Flag
commitments made ## POST-MEETING (immediately after): 1. Process
Notes: - Organize by topic - Extract action items - Identify
decisions - Note open questions 2. Send Follow-ups: - Thank you
email - Meeting summary - Action item list with owners - Next
steps 3. Update Systems: - Add notes to CRM - Schedule follow-ups
- Create calendar reminders - Update deal stage INTEGRATION: -
Calendar (Google/Outlook) - Zoom/Meet (recordings) - CRM
(Salesforce/HubSpot) - Email
```

# Section 7: Scaling Strategies

## 7.1 Multi-Agent Architecture

**The Specialist Pattern:** Instead of one agent doing everything, create specialized agents:

- **Research Agent:** Finds and summarizes information
- **Writing Agent:** Crafts content based on research
- **Review Agent:** Edits and improves content
- **Distribution Agent:** Publishes and promotes

Each agent is simpler, more reliable, and easier to debug.

## 7.2 Cost Optimization

```
# Cost Optimization Strategies 1. MODEL CASCADING: Try cheapest model first: - Haiku/GPT-3.5 → Draft content - If quality insufficient → Sonnet/GPT-4 Turbo - Only for final polish → Opus/GPT-4 Savings: 60-80% 2. BATCHING: Instead of 10 separate API calls: - Send all 10 items in one prompt - Process as batch - One response Savings: 30-50% (fewer API round trips) 3. CACHING: Store responses for common queries: - FAQ answers - Template content - Standard analyses Savings: 20-40% for repetitive tasks 4. CONTEXT OPTIMIZATION: - Remove unnecessary context - Use summaries instead of full docs - Clear conversation history periodically Savings: 10-30% (fewer tokens) 5. MONITORING: Track costs per agent: - Set daily/weekly budgets - Alert on anomalies - Optimize expensive agents - Kill runaway processes MONTHLY COST
```

```
TARGETS: - Small business: $50-100 - Medium business: $200-500 - Large business: $500-2000
```

### 7.3 Error Handling & Reliability

```
# Error Handling Best Practices RETRY LOGIC: - If API fails → Retry 3 times with backoff - If still fails → Log error, notify human - Never fail silently TIMEOUTS: - Set maximum execution time - Kill hung processes - Alert on timeouts FALLBACKS: - If GPT-4 fails → Try GPT-3.5 - If Claude fails → Try GPT-4 - If all fail → Human escalation VALIDATION: - Check output quality - Validate data formats - Verify links work - Confirm actions completed LOGGING: - Log all actions - Track success/failure rates - Monitor performance - Debug when issues occur ALERT CONDITIONS: - Error rate > 5% - Response time > 30 seconds - Cost spike > 200% of normal - Output quality drops
```

## Section 8: ROI Tracking & Measurement

### 8.1 What to Measure

| Metric | How to Track | Why It Matters |
|---|---|---|
| Hours Saved | Before/After time tracking | Direct productivity gain |
| Cost Per Task | API costs ÷ tasks completed | Efficiency metric |
| Error Rate | Failed tasks ÷ total tasks | Quality indicator |

| | | |
|---|---|---|
| Human Escalation Rate | Tasks sent to human ÷ total | Autonomy level |
| Response Time | Average time to complete | Speed vs human baseline |
| Customer Satisfaction | Surveys, feedback scores | Quality from user perspective |

## 8.2 Building Your ROI Dashboard

```
# ROI Dashboard Agent ## Task Weekly ROI report ## Instructions Every Monday morning: 1. Collect Metrics: - API costs (from all providers) - Tasks completed - Human hours saved - Error rates - Customer feedback scores 2. Calculate ROI: Cost = API spend + setup time Value = Hours saved × hourly rate ROI = (Value - Cost) ÷ Cost × 100% 3. Generate Report: - Executive summary - Detailed metrics - Week-over-week trends - Cost per task breakdown - Agent-by-agent performance - Recommendations 4. Send to: - Business owner - Team leads - Yourself VISUAL DASHBOARD: - Create simple HTML dashboard - Update in real-time - Show trends over time - Highlight anomalies TOOLS: - Google Sheets (simple) - Notion database (medium) - Custom web dashboard (advanced)
```

## Build Something Amazing

You now have the knowledge to build serious automation.

Start with one use case. Get it working. Then scale.

*— Charlie*