

OpenAI Tools Mastery: A Comprehensive 3-Hour Hands-on Workshop – Lab Guide

Module Walkthroughs

a) ChatGPT

Introduction

ChatGPT is OpenAI's conversational interface to their language models. It allows for natural back-and-forth interactions and is the most accessible entry point to OpenAI's capabilities.

Practical Walkthrough

1. Basic Conversation

- Log in to ChatGPT at <https://chat.openai.com/>
- Start a new chat by clicking "New chat" in the sidebar
- Type a simple greeting like "Hello, can you help me learn about artificial intelligence?"
- Observe the response and try follow-up questions

2. Using System Instructions

- Click on the "Custom instructions" option in your settings
- Add details about your preferences, background, and goals
- For example: "I am an IT Architect focusing on hybrid-cloud infrastructure. When explaining technical concepts, please use enterprise IT examples."
- Test how this affects responses in your conversations

3. Advanced Prompting Exercise

- Try this prompt: "I need to explain cloud computing to my team. Can you create a 5-minute presentation outline with key points about hybrid cloud benefits?"
- Refine the output by asking for more specific enterprise examples
- Request formatting changes to make it more presentation-friendly

b) Custom GPTs

Introduction

Custom GPTs allow you to create specialized versions of ChatGPT tailored for specific purposes, without requiring coding knowledge. They can include specialized instructions, knowledge files, and capabilities.

Practical Walkthrough

1. Creating Your First Custom GPT

- Navigate to <https://chat.openai.com/create>⁸
- Click "Create a GPT" and then select "Create new GPT"
- Enter a name for your GPT (e.g., "Hybrid Cloud Advisor")
- In the conversation interface, describe what you want your GPT to do: "I want you to be an expert on hybrid cloud infrastructure that helps IT professionals plan migrations and optimizations"¹⁸
- Let the builder guide you through the process of refining instructions

2. Adding Knowledge to Your GPT

- In the builder interface, click on the "Knowledge" tab
- Upload a relevant document (workshop materials will include sample cloud architecture documents)
- Test your GPT's ability to reference this information by asking related questions

3. Configuring Capabilities

- In the "Configure" section, enable or disable web browsing, DALL-E image generation, and code interpreter as needed
- For our example, enable web browsing to allow for current information on cloud services
- Test these capabilities by asking your GPT to create a comparison chart of current hybrid cloud offerings⁴⁸

4. Publishing and Sharing

- Save your GPT by clicking "Save" in the builder
- Choose whether to make it private or share it with others
- If sharing, you can get a link or publish to the GPT Store¹⁸

c) Prompt Engineering

Introduction

Prompt engineering is the art and science of crafting inputs to get the best possible outputs from AI models. Mastering this skill is essential for effective use of any generative AI system.

Practical Walkthrough

1. Basic Prompt Structure

- Start a new chat in ChatGPT
- Compare these two prompts:
 - Basic: "Tell me about virtualization."
 - Improved: "Explain virtualization technology for enterprise data centers, including benefits, challenges, and trends in 2025. Format as a technical briefing with sections."
- Observe the difference in detail, structure, and usefulness of responses

2. Using the Chain-of-Thought Technique

- Try this prompt: "I need to determine the best storage solution for my company. We have 500TB of data with 30% annual growth. 40% is rarely accessed cold storage, 50% is regular operational data, and 10% requires high-performance access. Walk through your reasoning step by step to recommend a hybrid storage architecture."
- Notice how asking for step-by-step reasoning produces more thorough analysis

3. Role and Format Specification

- Use this template: "You are a {role}. Your task is to {task}. The output should be in {format}."
- Example: "You are a senior cloud architect. Your task is to create a disaster recovery plan for a hybrid cloud environment using both AWS and on-premises infrastructure. The output should be in the format of an executive summary followed by implementation phases with estimated timelines."
- Experiment with different roles and formats to see how they affect the output¹³

4. Iterative Refinement

- Take the output from a previous prompt and ask for specific improvements
- Example follow-up: "This is good, but can you add cost considerations for each phase of implementation and highlight the technical prerequisites for each step?"

d) Model Comparison

OpenAI offers various models with different capabilities, strengths, and pricing. Understanding these differences helps you choose the right tool for each task.

Practical Walkthrough

1. Testing Different Models

- In ChatGPT, click the "GPT-4" or model selector dropdown at the top of the interface
- Note the available models (e.g., GPT-3.5, GPT-4, GPT-4o)
- Run the same complex prompt through different models
- Sample prompt: "Create a comprehensive analysis of microservices vs. monolithic architecture, including implementation costs, scaling considerations, and team structure implications."

2. Comparing Reasoning Abilities

- Test this logical reasoning problem across models:
- "A company is selecting a new cloud provider. Provider A costs \$10,000/month with 99.9% uptime. Provider B costs \$12,000/month with 99.95% uptime. Provider C costs \$15,000/month with 99.99% uptime. If each 0.01% of downtime costs the company approximately \$2,000 per month in lost revenue, which provider should they choose? Show your calculations."
- Compare the accuracy and detail of responses between models

3. Creative vs. Technical Tasks

- Try both creative and technical prompts:
 - Creative: "Write a short story about a CIO who discovers an AI living in their company's cloud infrastructure."
 - Technical: "Explain how containerization works at the operating system level and how it differs from traditional virtualization."
- Note which models excel at which types of tasks

4. Contextual Understanding and Memory

- Test how different models handle context from earlier in the conversation
- Start with: "My company uses VMware for virtualization."
- Several messages later, ask: "What monitoring tools would work well with the virtualization platform I mentioned earlier?"
- Compare how well different models remember and use this context

e) Code Generation

Introduction

OpenAI's models can generate, explain, and debug code across multiple programming languages, making them powerful tools for developers and those learning to code.

Practical Walkthrough

1. Basic Code Generation

- Ask for a simple script:
- "Create a Python script that reads a CSV file containing server information (hostname, IP, CPU cores, RAM) and outputs the total compute resources available in the environment."
- Run the generated code in your preferred environment or an online Python interpreter

2. Code Explanation

- Provide this sample code and ask for an explanation:

Javascript

```
async function fetchServerStatus(servers) {  
  const results = await Promise.all(  
    servers.map(async (server) => {  
      try {  
        const response = await fetch(`https://${server}/api/status`);  
        return { server, status: response.ok ? 'online' : 'error',  
data: await response.json() };  
      } catch (error) {  
        return { server, status: 'offline', error: error.message };  
      }  
    })  
  );  
  return results;  
}
```

- Ask follow-up questions about specific parts you don't understand

3. Code Debugging

- Present this buggy code:

Python

```
def calculate_storage_requirements(data_size_gb, growth_rate, years):  
    total_size = data_size_gb  
    for year in range(1, years):  
        total_size = total_size + (total_size * growth_rate)  
    return total_size  
  
print(calculate_storage_requirements(500, 0.3, 5))
```

- Ask the model to identify and fix the bugs (the range should start at 0 or end at years+1)

- Have it explain the consequences of the bug

4. API Integration Example

- Request code for integrating with OpenAI's API:
- "Write a Node.js script that uses the OpenAI API to summarize the content of IT incident reports. The script should take a text file as input and output a concise summary of the incident, key findings, and recommended actions."
- Have the model explain how to securely store and use the API key in the script

f) Document Conversion

Introduction

OpenAI's models excel at processing, transforming, and extracting information from documents, making them valuable tools for knowledge management and information extraction.

Practical Walkthrough

1. Text Summarization

- Copy and paste this sample technical document (provided in workshop materials) into ChatGPT
- Ask: "Summarize this technical document in 3 paragraphs, highlighting the key architectural components, benefits, and implementation considerations."
- Then try: "Now convert the same information into a 5-slide presentation outline with bullet points."

2. Format Conversion

- Provide an unstructured document (workshop materials) and ask:
- "Convert this unstructured information into a structured JSON format with categories for hardware requirements, software dependencies, network configuration, and security considerations."
- Review the structured output and ask for refinements

3. Information Extraction

- Use this prompt with the sample document:
- "Extract all mentions of technical specifications, costs, and timeline estimates from this document. Create a table with columns for Item, Specification/Value, and Page Reference."
- Check the accuracy of the extraction against the original document

4. Document Analysis

- Ask the model to perform deeper analysis:
- "Analyze this infrastructure proposal document for potential risks and omissions. What important considerations might be missing? What assumptions should be validated? What alternatives should be considered?"
- Use the analysis to understand how AI can help with critical review of documents

g) Vision Model

Introduction

OpenAI's vision capabilities allow models to process and understand images, creating powerful multimodal interactions that combine visual and textual information.

Practical Walkthrough

1. Image Analysis

- Use the provided sample diagram of a network architecture (workshop materials)
- Upload the image to ChatGPT and ask:
- "Analyze this network diagram. Identify the key components, potential security issues, and suggest improvements to the architecture."
- Note how the model interprets visual information and provides contextual insights

2. Technical Troubleshooting

- Upload a screenshot of an error message (workshop materials)
- Ask: "What might be causing this error in my virtualization environment? What troubleshooting steps would you recommend?"
- Follow up with specific questions about the error details

3. Visual Documentation

- Show an image of a server rack configuration
- Request: "Create documentation for this server rack setup, including a description of each component visible, their likely functions, and best practices for maintenance."
- Use this to see how vision models can help with technical documentation

4. Image-Enhanced Planning

- Upload a floor plan or data center layout
- Ask: "Help me plan cooling optimization for this data center layout. Identify hot spots, suggest improvements for airflow, and recommend optimal placement for additional cooling units."
- Observe how the model combines visual understanding with technical knowledge

h) Realtime Voice

Introduction

OpenAI's Realtime API enables live voice interactions with AI models, allowing for natural, conversational experiences and applications ranging from voice assistants to transcription services.

Practical Walkthrough

1. Setting Up Voice Interaction

- In ChatGPT with a Plus subscription, locate the microphone icon at the bottom of the chat interface
- Click it to activate voice mode
- Give permission for browser microphone access when prompted
- Try a simple voice query: "What are the primary considerations when designing a hybrid cloud architecture?"
- Notice how the model transcribes your speech and responds both textually and verbally¹⁹

2. Voice-Guided Troubleshooting

- Initiate a more complex voice interaction:
- "I'm trying to diagnose an issue with our containerized application that's running slow. Can you walk me through some diagnostic steps?"
- Continue the conversation entirely by voice, asking follow-up questions
- Note how the conversational flow differs from text-based interaction

3. Multi-turn Voice Conversation

- Try this sequence by voice:
- "I need to explain edge computing to my team."
- After the response: "Can you make that more technical?"
- Then: "Now create a comparison table between edge, fog, and cloud computing."
- Observe how context is maintained across voice interactions

4. Advanced Voice Features (If Time Permits)

- Experiment with different speaking paces and styles
- Try interrupting the model's response with a follow-up question
- Test how well it handles technical terms and acronyms in speech¹⁷¹⁹

i) API

Introduction

The OpenAI API allows developers to integrate AI capabilities directly into applications, websites, and backend services, providing programmatic access to the same models that power ChatGPT.

Practical Walkthrough

1. Setting Up Your First API Request

- Open a code editor and create a new file named openai-test.js
- Install the OpenAI SDK: `npm install openai`
- Create a basic script:

Javascript

```
import { OpenAI } from 'openai';

const openai = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY
});

async function main() {
  const completion = await openai.chat.completions.create({
    model: "gpt-4o",
    messages: [
      { role: "system", content: "You are a helpful IT infrastructure assistant." },
      { role: "user", content: "What are the key considerations for a successful hybrid cloud migration?" }
    ],
  });

  console.log(completion.choices[0].message.content);
}

main();
```

- Set your API key as an environment variable: `export OPENAI_API_KEY=your-api-key`
- Run the script: `node openai-test.js`

2. Working with Different Models

- Modify your script to compare responses from different models:

Javascript

```
async function compareModels() {  
  const models = ["gpt-3.5-turbo", "gpt-4o"];  
  
  const prompt = "Explain the concept of infrastructure as code and  
its benefits.";  
  
  for (const model of models) {  
    console.log(`\n--- Results from ${model} ---`);  
  
    const completion = await openai.chat.completions.create({  
      model: model,  
      messages: [{ role: "user", content: prompt }],  
    });  
  
    console.log(completion.choices[0].message.content);  
  }  
}
```

3. Function Calling via API

- Create a script with function calling capabilities:

Javascript

```
async function functionCallingExample() {
  const completion = await openai.chat.completions.create({
    model: "gpt-4o",
    messages: [
      { role: "user", content: "What's the storage capacity needed
for 500 VMs with an average of 250GB each, including 20% growth over
3 years?" }
    ],
    functions: [
      {
        name: "calculate_storage",
        description: "Calculate storage requirements with growth over
time",
        parameters: {
          type: "object",
          properties: {
            initial_capacity_gb: {
              type: "number",
              description: "Initial storage capacity in GB"
            },
            growth_rate: {
              type: "number",
              description: "Annual growth rate as a decimal"
            },
            years: {
              type: "number",
              description: "Number of years to project"
            }
          },
          required: ["initial_capacity_gb", "growth_rate", "years"]
        }
      }
    ],
    function_call: "auto",
  });

  console.log(JSON.stringify(completion.choices[0].message, null,
2));
}
```

4. Implementing a Simple Chatbot

- Create a basic chatbot application:

Javascript

```
import { OpenAI } from 'openai';
import readline from 'readline';

const openai = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY
});

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

const messages = [
  { role: "system", content: "You are a helpful IT infrastructure advisor specializing in hybrid cloud environments." }
];

function chat() {
  rl.question('You: ', async (input) => {
    if (input.toLowerCase() === 'exit') {
      rl.close();
      return;
    }

    messages.push({ role: "user", content: input });

    const completion = await openai.chat.completions.create({
```

```
    model: "gpt-4o",
    messages: messages,
  });

  const response = completion.choices[0].message.content;
  messages.push({ role: "assistant", content: response });

  console.log(`\nAI: ${response}\n`);
  chat();
});
}

console.log("IT Infrastructure Advisor Bot (type 'exit' to quit)\n");
chat();
```

- Run the chatbot and test it with infrastructure-related questions

Summary and Lessons Learned

Throughout this workshop, you've gained hands-on experience with the full spectrum of OpenAI's generative AI tools. Here are the key takeaways:

1. **Accessible AI:** You've seen how interfaces like ChatGPT make powerful AI capabilities accessible without requiring deep technical knowledge.
2. **Customization Power:** Through Custom GPTs and prompt engineering, you've learned how to tailor AI to specific use cases and domains.
3. **Multimodal Interactions:** The workshop demonstrated how combining text, vision, and voice creates richer and more natural AI experiences.
4. **Development Flexibility:** The API walkthrough showed how developers can integrate these capabilities into custom applications and workflows.
5. **Practical Applications:** Each module focused on real-world scenarios relevant to enterprise IT and infrastructure, demonstrating immediate practical value.
6. **Optimization Techniques:** You've learned how to select the right models, craft effective prompts, and structure interactions for optimal results.
7. **Rapid Prototyping:** The hands-on exercises illustrated how quickly new AI-powered tools and workflows can be created and tested.

The foundation you've built today will serve you well as generative AI continues to transform how we work with technology. The skills are transferable across platforms and will remain valuable as the AI landscape evolves.

Citations:

1. <https://www.freecodecamp.org/news/learn-generative-ai-in/>
2. <https://www.roitraining.com/courses/course-listing/course-937-hands-on-generative-ai-tools-and-techniques>
3. <https://www.linkedin.com/learning/hands-on-openai-api-building-a-real-world-solution>
4. <https://www.linkedin.com/pulse/lets-make-custom-gpt-10x-speed-video-shows-every-step-andy-crestodina-seovc>
5. <https://stories.autonationdrive.com/romantic/how-to-access-your-open-ai-login-a-complete-guide-to-seamless-authentication.html>
6. <https://help.openai.com/en/articles/7967234-enabling-multi-factor-authentication-mfa-with-openai>
7. <https://platform.openai.com/docs/guides/production-best-practices>
8. <https://www.youtube.com/watch?v=0Q1AQXpdGg>
9. <https://www.datacamp.com/tutorial/realtime-api-openai>

10. <https://github.com/saragibby/gen-ai-workshop>
11. <https://www.youtube.com/watch?v=oNA6pTNfHCY>
12. <https://sds2024.ch/a-hands-on-exploration-of-openai-api/>
13. <https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/>
14. <https://www.youtube.com/watch?v=3m4RY1FbwRU>
15. <https://help.openai.com/en/articles/6384897-dall-e-2-sign-in-and-sign-up-help>
16. <https://digitalstrategies.tuck.dartmouth.edu/programs/workshop-create-your-own-gpts/>
17. <https://www.youtube.com/watch?v=OVguB1h-eTs>
18. <https://openai.com/index/introducing-gpts/>
19. <https://platform.openai.com/docs/guides/realtime>
20. <https://learn.microsoft.com/en-us/azure/ai-services/openai/realtime-audio-quickstart>
21. <https://www.youtube.com/watch?v=mEsleV16qdo>
22. <https://otl.uoguelph.ca/teaching-assessment-resources/teaching-context-ai/teaching-generative-ai-workshops-and-events>
23. <https://github.com/microsoft/OpenAIWorkshop>
24. <https://help.openai.com/en/articles/10032626-prompt-engineering-best-practices-for-chatgpt>
25. <https://www.content.shi.com/cms-content/accelerator/media/training/course-ttai2305-outline.pdf>
26. <https://www.youtube.com/watch?v=kJfEhSS1xhU>
27. <https://www.datacamp.com/tutorial/azure-openai>
28. <https://www.youtube.com/watch?v=0Q1AQAxpdGg>
29. <https://openai.com/index/generative-models/>
30. <https://cognitiveclass.ai/learn-ai-hands-on-generative-ai>
31. <https://www.datacamp.com/tutorial/guide-to-openai-api-on-tutorial-best-practices>
32. <https://platform.openai.com/docs/guides/prompt-engineering>
33. <https://platform.openai.com/docs>
34. <https://help.openai.com/en/articles/6613629-why-can-t-i-log-in-to-openai-platform>
35. <https://help.openai.com/en/>
36. <https://platform.openai.com/docs/guides>

37. <https://help.openai.com/en/collections/3943089-account-login-and-billing>
 38. <https://help.openai.com/en/collections/6864559-login-issues>
 39. <https://platform.openai.com>
 40. <https://platform.openai.com/login>
 41. <https://help.openai.com/en/collections/3675947-account-management>
 42. <https://platform.openai.com/docs/assistants>
 43. <https://addepto.com/blog/what-is-an-openai-api-and-how-to-use-it/>
 44. <https://platform.openai.com/login?launch>
 45. <https://zapier.com/blog/custom-chatgpt/>
 46. https://www.youtube.com/watch?v=_yKfpW3B8tY
 47. <https://platform.openai.com/docs/tutorials>
 48. https://cookbook.openai.com/examples/agents_sdk/app_assistant_voice_agents
 49. <https://platform.openai.com/docs/api-reference>
 50. <https://www.youtube.com/watch?v=zJSYMWlCcPY>
 51. <https://help.openai.com/en/articles/4936824-can-i-change-my-authentication-method>
 52. <https://mitsloanedtech.mit.edu/ai/tools/writing/custom-gpts-at-mit-sloan-a-comprehensive-guide/>
 53. <https://www.youtube.com/watch?v=EWdCMPnm8uY>
 54. <https://www.coursera.org/learn/openai-custom-gpts>
 55. https://www.reddit.com/r/ChatGPTPro/comments/187knbp/seen_any_interesting_custom_gpts_also_tutorial_on/
 56. <https://www.youtube.com/watch?v=84JaqOq-7ws>
 57. https://www.youtube.com/watch?v=AZ0WzizQ_QA
 58. <https://learn.microsoft.com/en-us/azure/ai-services/openai/how-to/realtime-audio>
-