CURRICULUM PREVIEW

From Zero to Chatbot

AI Foundations for Middle School

Semester 1 | 17-Week Curriculum

What's Included in the Full Package:

- ✓ Complete 17-week curriculum guide with detailed lesson plans
- ✓ Supporting PowerPoint presentations
- ✓ Google Colab notebooks with hands-on coding activities
- ✓ History of AI companion guide with discussion questions
- ✓ Printable resources and worksheets
- ✓ Train the Trainer session for co-ops and schools

This Preview Includes:

- Curriculum overview & sample lesson plans (Weeks 1 & 9)
- Sample PowerPoint slides (Week 12 - Sentiment Analysis)
- Sample Google Colab notebook (Week 14 - Build a Chatbot)
- History of AI companion guide excerpt

Navigate AI LLC

navigateai.io | navigate.ai.llc@gmail.com

NAVIGATE AI

Middle School Curriculum

Grades 6–8 | Semester 1

17-Week Comprehensive AI Education Program

90-Minute Weekly Sessions | Detailed Lesson Plans

Navigate AI LLC
San Antonio, Texas
2026

Program Overview

Navigate AI's Middle School Semester 1 curriculum is an 17-week program designed to take students from AI curiosity to hands-on AI builders. Students progress through five units that build on each other, beginning with foundational AI concepts, advancing through data literacy and creative AI applications, and culminating in Python programming for AI and a capstone chatbot project.

By the end of this semester, every student will have built a working, customized AI chatbot in Python — a tangible portfolio piece that demonstrates real technical skill.

Class Format

Each weekly session is 90 minutes, structured to maximize engagement and hands-on learning time:

- **Opening Hook & Review** (~10 min) — Quick recap, warm-up question, or "what did you try since last week?"
- **Core Lesson** (~25 min) — New concept introduction, kept tight and interactive
- **Hands-On Activity** (~40 min) — The main build/explore/create block where students apply what they learned
- **Wrap-Up & Reflection** (~15 min) — Share work, discuss surprises, preview next week

Semester at a Glance

Unit	Title	Weeks	Duration
Unit 1	AI Foundations	Weeks 1–3	3 weeks
Unit 2	Data Literacy	Weeks 4–6	3 weeks
Unit 3	Creative AI & Hands-On Building	Weeks 7–8	2 weeks
Unit 4	Python for AI	Weeks 9–14	6 weeks
Unit 5	Capstone: Build Your Own AI Chatbot	Weeks 15–17	3 weeks

Package Bundles

Package	What's Included	Duration
AI Kickstart	Unit 1 only	3 weeks
AI Explorer	Units 1–3 (Foundations + Data + Creative AI)	8 weeks
Full Semester	All 5 units with Python and capstone	17 weeks
Python for AI Add-On	Unit 4 standalone for students with prior AI exposure	6 weeks

Detailed Lesson Plans

The following section provides detailed 90-minute lesson plans for each of the 17 weeks in the semester. Each lesson plan includes learning objectives, materials needed, a minute-by-minute class flow, and instructor notes.

Unit 1: AI Foundations

Weeks 1–3 | *The gateway to understanding artificial intelligence*

Week 1: What IS Artificial Intelligence?

Learning Objectives

- Students can define artificial intelligence and distinguish it from traditional software
- Students can identify at least five examples of AI in their daily lives
- Students can describe the basic timeline of AI development

Materials Needed

- Projector/screen with internet access
- Student devices (laptops, tablets, or Chromebooks)
- Help students pre-load tabs on their computers: Google Lens, Claude.ai, AIMusicGen.ai, Google ImageFX
- Printed or digital "AI or Not AI?" sorting cards
- Whiteboard/markers or digital whiteboard (optional)
- **Teacher-Prepared Resources (included with curriculum):**
- 📎 **Week 1 PowerPoint — NavigateAI_Week01_What_Is_AI.pptx**
- 📎 **Week 1 "AI or Not AI?" Sorting Cards — NavigateAI_Week01_AI_or_Not_AI_Cards.docx**
- 📎 **Week 1 AI Tool Observation Sheet — NavigateAI_Week01_AI_Tool_Observation_Sheet.docx**

Class Flow — 90 Minutes

0:00 – 0:10 — Opening Hook

- Icebreaker: "Raise your hand if you've used AI today." Most hands stay down.
- Reveal: Walk through their morning — phone alarm (smart assistant), social media feed (recommendation AI), autocorrect (NLP), navigation app (route optimization). "You've probably used AI 10+ times before breakfast."
- Pose the big question: "So what actually IS artificial intelligence?"

0:10 – 0:35 — Core Lesson: Defining AI & Brief History

- Open the Week 1 PowerPoint presentation. This provides a guided class discussion.
- Define AI: Software that can learn from data and make decisions or predictions, rather than just following fixed instructions
- Key distinction: Traditional software follows rules a programmer wrote. AI learns patterns from examples.
- Brief history timeline (keep it engaging, not a lecture): Alan Turing and the "can machines think?" question (1950), early chess computers (1990s), Siri and virtual assistants (2011), AlphaGo beats world champion (2016), ChatGPT launches (2022), today's AI explosion

  (Check the PowerPoint script for this slide for extra information.)
- Discussion checkpoint: "What surprised you? What do you think comes next?"

0:35 – 1:15 — Hands-On Activity: AI Tool Exploration & Sorting

- Use the instructions and questions in the PowerPoint slides to walk through these hand-on AI tool activities.
- Part 1 (15 min): Students rotate through 4 AI tool stations (or tabs on their device). At each station they interact with the tool and fill out a simple observation sheet: What does this tool do? What data does it need? How does it seem to "learn"?

- Part 2 (15 min): “AI or Not AI?” sorting activity. Students get a set of 15–20 cards describing different technologies and must sort them into “This is AI” vs. “This is NOT AI.” (Use week 1 sorting card resource.) Discuss the tricky ones as a group.
- Part 3 (10 min): Small group discussion - students share their most surprising discovery and brainstorm additional AI examples from their lives

1:15 – 1:30 — Wrap-Up & Reflection

- Class share-out: Each group names one AI they found most surprising
- Exit ticket: “In your own words, what makes something AI vs. regular software?”
- Preview Week 2: “Now that we know what AI is, next week we’re going to learn HOW it actually learns.”

Instructor Notes

- The “AI or Not AI” sorting activity generates great debate. Lean into the gray areas. Explain there is a lot that can be accomplished with coding. A calculator is not AI; a calculator app that learns your common calculations might be.
- Students often think “smart” = AI. Help them understand the distinction between programmed rules and learned behavior.

Week 9: Python Basics

Learning Objectives

- Students can explain why Python is the dominant language in AI/ML
- Students can write simple Python programs using variables, data types, print, and input
- Students can navigate Google Colab and run code cells

Materials Needed

- Student devices with Google accounts (for Google Colab)
- Projected Colab notebook for live coding
- Printed Python Quick Reference card or digitally available
- Week 9 Data Type Slide Deck
- Week 9 Word document of code snippets
- Pre-built starter notebook with guided exercises
- **Teacher-Prepared Resources (included with curriculum):**
- 📎 **Week 9 PowerPoint — NavigateAI_Week9_DataTypes.pptx**
- 📎 **Week 9 Code Snippets Reference — NavigateAI_Week9_CodeSnippets.docx**
- 📎 **Week 9 Google Colab Notebook — NavigateAI_Week9_CodeSnippets.ipynb**

- **Instructor Preparation: Getting Familiar with Google Colab and Python**
- This lesson uses Google Colab as the coding environment. Google Colab is a free, browser-based tool from Google that lets you write and run Python code without installing anything on your computer. If you've used Google Docs, Colab works the same way. It lives in your browser, saves to Google Drive, and can be shared with students via a link. All you need is a Google account.

- If you are not already familiar with Google Colab, watch this beginner walkthrough before class: **"Google Colab Tutorial for Beginners"** by Corey Schafer on YouTube (https://www.youtube.com/watch?v=inN8seMm7UI). This is a clear, concise walkthrough of the Colab interface that covers creating a notebook, running code cells, adding text cells, and navigating the menus — everything you need for this lesson. Alternatively, Colab has its own built-in tutorial: go to colab.research.google.com, click "File" → "Open Notebook," and select the "Welcome to Colaboratory" example notebook, which walks you through the interface interactively.

- If you are also new to Python itself, don't worry, you do not need to be an experienced programmer to teach this lesson. The code students will write is simple (print statements, variables, and input), and all of it is provided in the code snippets handout. To build your confidence, watch **"Python in 100 Seconds"** by Fireship on YouTube (https://www.youtube.com/watch?v=x7X9w_GIm1s). This is a fast, high-energy overview that explains what Python is, why it's popular, and what the syntax looks like all in under two minutes. If you want a slightly deeper introduction (still short), follow it with **"Python for Beginners — Learn Python in 1 Hour"** by Programming with Mosh (https://www.youtube.com/watch?v=kqtD5dpn9C8), which covers variables, data types, input/output, and conditionals, the exact topics in this lesson. You don't need to watch the full hour; the first 20 minutes cover everything students will encounter in Week 9.

- The key concepts you need to be comfortable with for this class: how to create and run a code cell in Colab (click the Play button or press Shift+Enter), what a variable is (a labeled container that stores data), what input() does (asks the user a question), what print() does

(displays output), and the difference between strings (text in quotes) and integers (numbers without quotes). The Data Types PowerPoint and code snippets handout cover all of this in student-friendly language, so you can teach from those materials directly.

Class Flow — 90 Minutes

0:00 – 0:10 — Opening Hook: "Talk to Your Computer"

- **Instructor Note:** The following exercises include code snippets to be used in Google Colab Notebooks for demos and class activities. There is a detailed Week 10 word document with code snippets that are copy and paste friendly for Google Colab IDE or the instructor can choose to open the pre-loaded Google Colab notebook.

- Open by asking the class: "If you could type one sentence to your computer and have it actually do what you said, what would you tell it to do?" Let a few students share answers. They'll say things like "make me a game," "do my homework," "play music."

- Then say: "Here's the thing, you can. That's literally what programming is. You type instructions, and the computer follows them. The only catch is you have to speak its language."

- Pull up a Google Colab notebook on the projector (colab.research.google.com — free, runs in the browser, nothing to install). Type this live in front of them: (See Week 10 Word document and/or pre-built Google Colab notebook)


```python
name = input("What is your name? ")
print("Hey " + name + ", you just talked to a computer.")
```

- Run it. Type a student's name when prompted. The output appears instantly. The class sees that two lines of code created a real, working interaction.
- Then type this:

```python
import random
responses = ["Definitely yes", "Ask again later", "No way", "Signs point to yes", "Cannot predict now"]
question = input("Ask the Magic 8-Ball a question: ")
print(random.choice(responses))
```

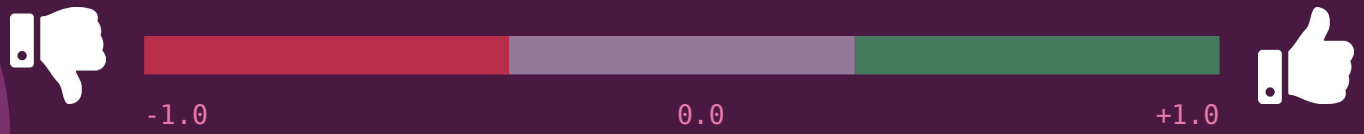
- Run it. Let a student ask a question. The class just watched you build a Magic 8-Ball in four lines of code. They'll want to try it immediately.
- Close with: "Everything you've learned in this class: AI tools, chatbots, prompt engineering, it's all built on code that looks exactly like this. Python is the language that powers almost all of AI, and it was specifically designed to be easy to read and write. You're going to learn it starting right now."
- Hand out the Google Colab setup instructions and move directly into the core lesson.

0:10 – 0:35 — Core Lesson: Google Colab Setup & First Code

LESSON 13

Sentiment Analysis

Teaching Python to Read Feelings



Navigate AI

What Is Sentiment Analysis?

Automatically determining whether text expresses a positive, negative, or neutral feeling.

Positive

`+0.5 to +1.0`

"I loved this!"

Neutral

`-0.1 to +0.1`

"It was okay."

Negative

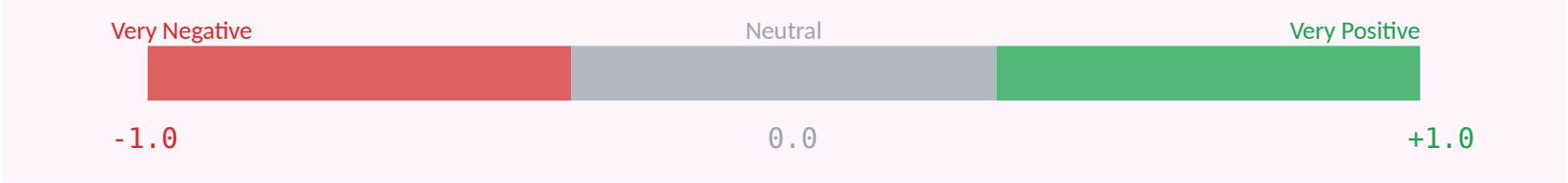
`-1.0 to -0.5`

"Terrible waste of time."

NLP (Natural Language Processing) — the branch of AI that helps computers understand human language. Sentiment analysis is one of its most common applications!

Your NLP Toolbox: TextBlob

TextBlob is a Python library that makes text analysis simple. It scores text on a polarity scale:




```
from textblob import TextBlob

# Analyze a sentence
text = TextBlob("This movie was absolutely amazing!")

print(text.sentiment.polarity)   # 0.85
print("Positive! 😊")              # We decide the label
```

3 Break the Model!

10 minutes — Find the AI's weaknesses

Try to find inputs that confuse the sentiment analyzer!

🎭	**Sarcasm**	*"Oh wonderful, another meeting."*
🔀	**Mixed Sentiment**	*"Food was great but the wait was awful."*
💬	**Slang / Informal**	*"that movie was lowkey fire ngl"*
🤔	**Context-Dependent**	*"This vacuum really sucks!"*
🔄	**Negation Tricks**	*"I wouldn't say it was bad."*

Document what works and what doesn't! This is real AI research.

SAMPLE: Google Colab Notebook

Week 14: APIs & Your First AI Chatbot

The Restaurant Analogy

You don't walk into the kitchen and cook your own food. You tell the waiter what you want, the kitchen makes it, and the waiter brings it back.

```
YOU → WAITER → KITCHEN → WAITER → YOU

YOUR CODE → API REQUEST → AI MODEL → API RESPONSE → YOUR CODE
```

Your First API Call

This is the moment of truth — your code is about to talk to an AI model running on Google's servers, and it will talk back.

```
# Your first API call! Send a message to the AI.

response = client.models.generate_content(
    model=MODEL,
    contents='What is the coolest animal in the world and why?'
)

print('AI says:')
print(response.text)
```

System Prompts — Give the AI a Personality!

A **system prompt** is like giving the waiter special instructions before they take your order. Without it, the AI is generic. With one, you can make it talk like a pirate, act as a math tutor, or only speak in haikus!

```
# Demo: Same question, DIFFERENT system prompts

question = 'What is 2 + 2?'

# --- Pirate Mode ---
print('PIRATE MODE')
answer = chat(question, system_prompt='You are a friendly pirate.
              Answer everything in pirate speak.')
print(answer)

# --- Math Tutor Mode ---
print('MATH TUTOR MODE')
answer = chat(question, system_prompt='You are a strict but kind
              math teacher. Always explain your reasoning.')
print(answer)
```

Same question. Same AI model. Totally different answers.

That's the power of the system prompt. This is one of the most important concepts in working with AI: **how you prompt the model changes everything.**

NAVIGATE AI

History of AI

Companion Guide for Middle School Semester 1 Curriculum

17 Stories That Shaped Artificial Intelligence

From Turing's Enigma Machines to the ChatGPT Revolution

Navigate AI LLC
San Antonio, Texas
2026

How to Use This Guide

This companion guide provides one historical AI story for each of the 17 weeks in the Navigate AI Middle School Semester 1 curriculum. Each story is carefully matched to the lesson's topic so that instructors can weave real-world history directly into classroom teaching.

Each entry includes three components:

The History is a 3-paragraph narrative written at a middle school reading level. These are designed to be engaging and story-driven, not dry encyclopedia entries. Instructors can read them aloud, assign them as reading, or summarize the key points in their own words.

The Instructor Hook is a short paragraph explaining exactly where in the lesson to introduce the story and how to connect it to what students are learning. This is the "bridge" between history and hands-on activity.

The Discussion Question is an open-ended question that connects the historical story to the lesson's activity. These work well as opening hooks, exit ticket prompts, or small group discussion starters.

The stories span over 180 years of history, from Herman Hollerith's 1890 Census punch cards to the deep learning revolution of 2012, and are chosen to show students that AI didn't appear overnight. It's the product of centuries of human curiosity, failure, persistence, and breakthrough.

A full bibliography of sources is provided at the end of this guide, organized by week, for instructors who wish to explore any topic in greater depth.

WEEK 1 | Unit 1: AI Foundations

Lesson: **What IS Artificial Intelligence?**

The Turing Test & the Birth of a Dream (1950)

In 1950, British mathematician Alan Turing published a paper called "Computing Machinery and Intelligence" that asked one of the most daring questions in the history of science: "Can machines think?" To answer it, he proposed what we now call the Turing Test. If a human judge has a conversation with both a machine and a person (without seeing either) and cannot reliably tell which is which, the machine should be considered intelligent.

Turing wasn't writing in a vacuum. During World War II, he had been the lead architect behind the effort to crack the German Enigma code at Bletchley Park. The Enigma machine was a typewriter-like device that scrambled military messages using a system so complex that it produced 150 quintillion possible settings. The German military believed it was unbreakable. Turing and his team built a series of electromechanical machines called "Bombes" that could test thousands of Enigma settings per hour, searching for patterns in intercepted messages. Their work is estimated to have shortened the war by two years and saved millions of lives.

After the war, Turing turned his attention from code-breaking to code-thinking. His 1950 paper planted the seed that grew into the entire field of artificial intelligence. Six years later, in the summer of 1956, a group of researchers gathered at Dartmouth College and officially coined the term "artificial intelligence." This launched a field that would take decades to fulfill even a fraction of Turing's vision.

INSTRUCTOR HOOK — *How to tie this into the lesson*

When students learn the definition of AI and see the timeline of its development, share Turing's story. The man who helped defeat the Enigma didn't stop at winning a war. He asked whether machines themselves could one day think. That question is the reason the field your students are studying exists.

DISCUSSION QUESTION — *For students*

Turing proposed his test in 1950. Over 70 years later, do you think today's chatbots would pass it? What would you ask an AI to figure out if it was really 'thinking' or just predicting words?