Introduction to Claude Artifacts

Artifacts are substantial, self-contained pieces of content created by Claude, such as documents or code snippets, that are shared in a separate window for easy modification, reference, or reuse later. In this course, you'll learn how to use Anthropic AI's Artifacts feature to create documents, websites, and interactive components with GenAI. Discover the practical aspects of working with Claude, including how to create and publish Artifacts, use prompt engineering to create markdown documents, and generate SVG graphics and Mermaid.js diagrams by prompting and iterating with Claude. Finally, explore how to create HTML websites and interactive React components with Claude Artifacts. By the end of the course, you'll have a solid understanding of how to work with Claude Artifacts, and be able to use them to create a wide range of digital projects.

Table of Contents

1. Video: Course Overview (it_ccclauid_01_enus_01)

In this video, we will discover the key concepts covered in this course.

- *discover the key concepts covered in this course*

[Video description begins] *Topic title: Course Overview. Presented by: Joe Holmes.* [Video description ends]

Welcome to Introduction to Claude Artifacts. My name is Joe. I'm an ML/AI curriculum developer at Codeacademy, and I will be your instructor.

In this course, we will learn what Claude Artifacts are, explore how to prompt engineer Artifacts; provide a guide to the various types of Artifacts you can create; present some common use cases and examples, which will hopefully spark your own imagination and discover how to publish and remix both your own and other people's Artifacts.

Let's get started.

2. Video: What Are Claude Artifacts? (it_ccclauid_01_enus_02)

After completing this video, you will be able to identify what Claude Artifacts are and what you can accomplish with them.

- *identify what Claude Artifacts are and what you can accomplish with them*

[Video description begins] *Topic title: What Are Claude Artifacts? Presented by: Joe Holmes.* [Video description ends]

Let’s begin by diving into the Claude model that makes Artifacts possible. Claude 3.5 Sonnet, released on June 20th, 2024, is a powerful frontier model that outperforms OpenAI’s 4o model and Google's Gemini 1.5 Pro on a variety of leading evaluative benchmarks. Artifacts were released during Claude 3.5 Sonnet’s debut. And they can be thought of as representing a paradigm shift in the nature of human and AI model interaction.

Previous models can be thought of as a user-model conversation, where you chat back and forth with the large language model as a sort of one-on-one consultation. Artifacts represent something more like a collaboration, where you and the model pull up some content you’re working on, an Artifact, and think through how to design and improve upon it.

So what are Artifacts? They appear in the Claude window to the right of your chat, and they grant users the ability to write documents, design diagrams, code websites, and develop applets in real time. They can then be edited and improved upon, downloaded, and even published to the web.

As we’ll see later, you can even remix other people’s published Artifacts. So now, let’s fire up Claude and see one in action. I’ll open up the claude.ai chatbot window and simply enter in the text input the following prompt: [Video description begins] *The screen shows the Claude interface with the title at the top left corner. A conversation area displays a greeting and a prompt for assistance from the user. At the bottom, there is a text input field with options to initiate an action, to upload an attachment and an image. There are also icons for options such as sharing and settings in the top right corner.* [Video description ends] ‘Generate an artifact of a smiling cartoon character,’ and press Enter.

I’ll now watch Claude get to work. It first thinks, then writes the SVG code and then renders a smiling face to our browser. If we wanted, we could copy the code for the SVG or download the SVG to a file. [Video description begins] *The Claude screen is split into two sections with a conversation on the left and an image preview on the right. The conversation includes exchanges between the user and the Claude assistant about creating a cartoon character, with the assistant providing a description that matches the smiling cartoon face in the preview. At the bottom of the conversation pane are options to copy and retry, while at the bottom right corner of the image preview pane is an option to download the image and to copy.* [Video description ends] The types of Artifacts you can currently create are: Document Artifacts, SVG Artifacts like the one we just made, Mermaid.js diagrams for flowcharts, HTML websites, React.js

components for more complex interactive UIs, and Code files in a wide variety of programming languages.

3. Video: Document Artifacts (it_ccclauid_01_enus_03)

During this video, discover how to create well-formatted markdown documents via prompt engineering with Claude.

- *create well-formatted markdown documents via prompt engineering with Claude*

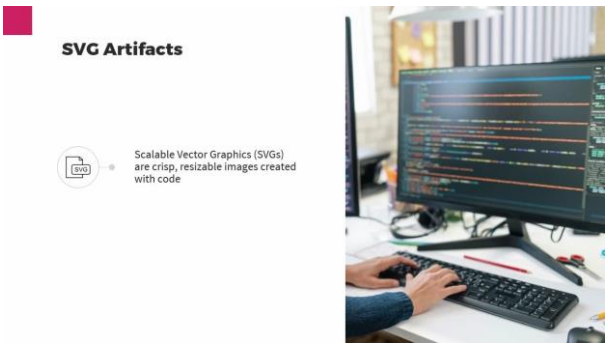
[Video description begins] *Topic title: Document Artifacts. Presented by: Joe Holmes.* [Video description ends]

We'll begin creating Artifacts with a simple example: creating text Artifacts with Claude. Claude's document Artifacts are written in Markdown, which is an easy-to-read and easy-to-learn text formatting language. Common use cases for Claude document Artifacts include: brainstorming and writing down ideas from a chat session with Claude; creating documentation for a coding feature or new process you've developed at work; and drafting formal documents and official communications using specified styles and consulting external knowledge sources.

Now, let's try out creating document Artifacts by opening claude.ai, beginning a new chat, and entering the following prompt: "I want to create a style guide for new hires at my artisanal dog treat company that specifies our tone, voice, and grammatical rules in communications. Ask me five questions about our preferred communication methods, then create a markdown Artifact that lays down our basic style guidelines in clear, readable prose." Now I'll push Enter and Claude will invent for me five questions about a preferred communication style, and I can answer them, and it will use those answers to create the Artifact.

So I'm going to answer these questions and then we'll pick right back up. All right, I've answered the five questions, and now, with this information, Claude should be able to create a helpful Document Artifact. So, I'll press Enter, and Claude will get to work drafting the document. On the right-hand side of the screen, Claude is outputting a series of Markdown headers that contain our thoughts on contractions, emojis, industry jargon, Oxford commas, and so on. With this completed Artifact, we can download it by clicking the Download to File button in the bottom right, or copy the markdown directly to our clipboard and use it at our job.

4. Video: SVG and Diagram Artifacts (it_ccclauid_01_enus_04)



In this video, you will learn how to generate SVG graphics and Mermaid.js diagrams by prompting and iterating with Claude.

- *generate SVG graphics and Mermaid.js diagrams by prompting and iterating with Claude*

[Video description begins] *Topic title: SVG and Diagram Artifacts. Presented by: Joe Holmes.* [Video description ends]

Now, let's explore how Claude Artifacts can help us create visual content using SVGs and Mermaid.js diagrams. But before we get started, let's explain what each type of Artifact actually is. SVG is short for Scalable Vector Graphics, are great ways to quickly obtain crisp, resizable images.

SVGs are written in HTML code and are lightweight and performant way to share graphics on the web. Use cases include icon sets, resizable web art, and general graphic design. Mermaid.js is a library and markup language used to make diagrams. It's simple to learn and simple to write.

Use cases include making flowcharts and system design diagrams, mind mapping out complex concepts, and developing infographics to help communicate your ideas. [Video description begins] *The screen shows a slide titled Mermaid.js Diagrams. On the left, there is a brief description of Mermaid.js as a diagramming language and its use cases, which include flow charts, diagrams, and infographics. On the right side, there is a flowchart example with a start point, a decision-making step, two possible processes, and an end point.* [Video description ends]

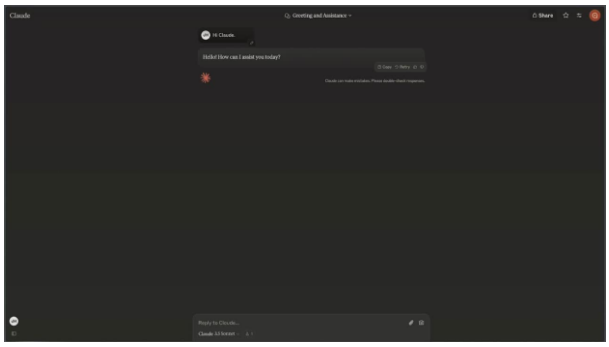
Now, let's hop into Claude and see how these tools can be used to create helpful Artifacts. All right, I have once again got claude.ai's chat window open, and I am going to create an SVG Artifact now. I'll do so by pasting the following prompt into the text input: 'I'm creating a website for a renewable energy company. Create an SVG Artifact of an icon for the website. The icon should depict a solar panel absorbing sunlight. It should be one color, line art with no fills, and should be simple.' I'll now press Enter and Claude will get to work.

On the right-hand side of the screen, Claude is writing the SVG code and then rendering the solar panel icon, which is a simple rectangular grid representing the solar panel and an array of sunlight lines representing the sunlight that it's absorbing. Looks pretty good. I can iterate on this icon by replying to Claude and saying, for instance, 'Now add color.' Claude will receive my reply and rewrite the SVG to be colored.

Once again, the right-hand side of the screen is filling out with code, and the rendering now includes yellow rays of sun instead. All right, I have once again opened the chat window, and to create a Mermaid.js diagram, I will insert the following prompt: 'Create a Mermaid diagram of a typical sales funnel of a B2B enterprise software-as-a-service company.' I'll now press Enter, and Claude will get to work writing the Mermaid code.

On the right-hand side of the screen, Claude is writing the simple Mermaid markup, and then renders a nice flowchart to the Artifact window. In the bottom right of the screen, there are controls to zoom in and zoom out, as well as to see a full-screen version of the flowchart.

5. Video: HTML Artifacts (it_ccclauid_01_enus_05)



Find out how to create HTML websites with Claude Artifacts - both simple presentation websites and more complex sites that import JS libraries.

- *create HTML websites with Claude Artifacts - both simple presentation websites and more complex sites that import JS libraries*

[Video description begins] *Topic title: HTML Artifacts. Presented by: Joe Holmes.* [Video description ends]

In this lesson, we'll explore Claude's ability to generate HTML Artifacts. HTML is the basic markup language of the Internet, and Claude can write it like a pro. In addition to prompting and designing a simple website, we'll also experiment with Claude's ability to use JavaScript libraries to grant it more advanced functionality.

Let's head over to Claude and give it a try. For our first exploration of HTML Artifacts, we will make a HTML website out of an existing resume PDF. To do so, we'll begin by attaching the existing PDF by going to the right-hand corner of the text input and clicking on the Upload docs or images button. In this case, the sample resume is from the Yale School of Management's career development site, but you could use whatever you wanted.

I'll upload the PDF and enter the following prompt: 'I'd like you to convert the attached resume PDF into an attractive personal website showcasing my accomplishments. Use tasteful but impressive graphic design, color, typography, and visual hierarchy. I don't know how to code, so explain to me what the HTML and CSS are doing in simple terms.' Now I'll press Enter, and let Claude get to work. It soon opens up an Artifact window and begins writing the HTML, beginning with the CSS and in a style tag and proceeding to sections for education, work experience, additional information, and so on.

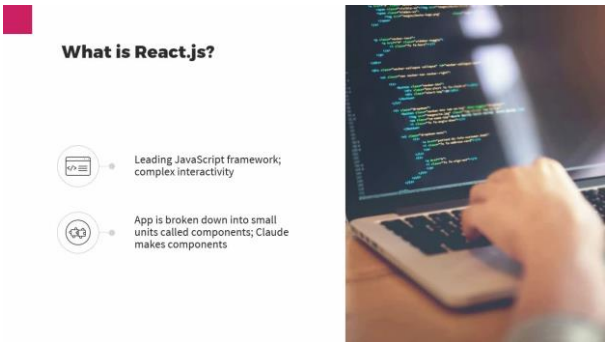
When it's done, the website is rendered in the Artifact window and we can scroll through it and take a look at what it says. Notice also that on the left-hand side of the chat window, Claude obeys my request and explains the structure of the HTML and the CSS styling in a way that's approachable to a beginner programmer. If you wanted to use this HTML, you could go down to the bottom right of the Artifact window and either copy the contents or download the HTML as a file.

From there, you could serve it on a web server if you wanted a nice resume website. Next, let's try out using some JavaScript in our HTML Artifacts. This time, we'll instruct Claude to create an HTML Artifact that uses JavaScript and animations. I will simply add to the text input at the bottom of the screen: 'Generate an HTML Artifact that simulates the solar system. Animate all eight planets plus Pluto orbiting the Sun. Show their orbit paths as thin white circles.' I'll now press Enter, and Claude will get to work.

This time, in addition to the HTML elements, Claude will write JavaScript, a programming language built for the Internet that can help us accomplish more advanced techniques in our websites. When it finishes, on the right-hand side of the screen, there is a solar system simulation. The smaller planets closer to the Sun have faster orbits, and the ones further out, including Pluto, have slower orbits.

The planets are colored differently, and the Sun is even glowing. And we can iterate on this simulation too. For instance, I might say as a follow-up, 'Put a starry background shimmering behind the planets.' And now Claude will rewrite the Artifact and include a simple animation in the background to represent the twinkling stars of the Milky Way galaxy. We'll once again watch Claude rewrite the code, and then when it nears completion, it will render that simulation to the Artifact window once again. And there are our shimmering stars. Pretty neat right?

6. Video: React.js Artifacts (it_ccclauid_01_enus_06)



In this video, discover how to generate interactive React components with the full functionality of modern web front-ends using Claude.

- *generate interactive React components with the full functionality of modern web front-ends using Claude*

[Video description begins] *Topic title: React.js Artifacts. Presented by: Joe Holmes.* [Video description ends]

Among the most powerful features of Claude Artifacts is the capacity to make custom React.js components. But first, what is React.js? React.js, or just React, is a framework used by web developers to create complex software applications using JavaScript.

While HTML and Vanilla JavaScript are sufficient for simple exercises, for more dynamic software, we need a framework to use JavaScript to its full potential. In React, parts of the app are broken down into files known as components. Claude Artifacts can create a single React component and render it in the Artifact window. You can interact with it inside the Claude application, or if you have web development experience, you can copy and paste the code into a React code base.

So, what are the main differences between HTML Artifacts and React.js Artifacts? HTML Artifacts have a lower learning curve, meaning, you can troubleshoot and manually handle them without knowing much about code, and they can run on their own as a single file in the browser. There's a lower ceiling for what you can accomplish in a single HTML file compared to a React component, but you can still do quite a lot.

React, on the other hand, comes with a more challenging learning curve as the leading web development framework. Components are designed to run inside React codebases, meaning, you'll need to manually integrate them into a React project if you want to use it off of the Claude platform. But it does come with a high ceiling for what you can accomplish. One final thing to mention about React Artifacts is that they come pre-installed with a few libraries that are great for quickly creating user interfaces.

These include: Shadcn-ui, an extremely popular UI component library that can quickly get you designing professional-looking interfaces; TailwindCSS, a leading CSS framework; Recharts.js, an interactive data visualization library; and Lucid Icons, a versatile open-source icon set. Now, let's try making some React Artifacts and seeing what they can do.

Now, we can give React Artifacts a try by entering the following prompt into the text input: 'Generate a React Artifact for a recipe planner. The planner should allow the user to select a recipe, render that recipe, and display a grocery list with checkboxes for all of the recipe's ingredients. There should also be a number input that can set the desired servings of the recipe, and the quantities of ingredients should change appropriately with the serving count. Use shadcn-ui and Tailwind to create a sleek, attractively styled user interface using color and design.'

I'll now press Enter, and we can let Claude get to work on our React component. If you take a look at the generated code on the right-hand side of the screen, you'll see that the syntax is different from our HTML Artifacts, but remains relatively readable even for outsiders to the web development world. It is designing our recipe template using Shadcn-ui as well as Tailwind for styling, and then it renders our app to the Artifact window. Now, if I select different recipes, I can see different ingredients and instructions rendered to the Artifact window.

Likewise, if you note the serving sizes for each of our ingredients, as I increase the serving count, they reactively change in accordance with the number of servings that I want to prepare. I can also check list off different ingredient items as I purchased them at the grocery store. This reactive user interface design is a crucial feature of React.js, and in the next example, we'll see a more complex demonstration of how it works. This time, we'll experiment with visualizing some data using React Artifacts and CSV data from the Internet.

So, I have a CSV file that I've downloaded from Our World in Data, and it's on the annual global corporate investment in artificial intelligence by investment type. And with the CSV, I will click on the Upload docs or images button in the text input form at the bottom of the screen. I'll select the CSV and I'll input the following prompt: 'Analyze this CSV of AI investment and generate a React Artifact that visualizes it in a variety of ways.

Offer a dropdown at the top of the component that allows me to select from among a few different Recharts chart types. Then render the chart type I choose. Feel free to select a limited sample of data in the CSV for length purposes. Use shadcn-ui and Tailwind to give the applet a sleek, professional look.'

I'll now press Enter, and let Claude get to work. This time, the code generation may take a little while, so, I'm going to fast-forward to when the Artifact has rendered. All right, as we can see, we now have obtained an AI investment visualization applet. And all of this was done in React and written via Claude Artifacts. Each of the types of investment— mergerAcquisition, minorityStake, privateInvestment, and publicOffering— are captured in the data. Here we see a Bar Chart, but we could also select a Line Chart or an Area Chart or a Pie Chart. Pretty cool.

7. Video: Publishing and Remixing Artifacts (it_ccclauid_01_enus_07)

Learn how to publish Artifacts and remix other people's Artifacts using Claude.

- *publish Artifacts and remix other people's Artifacts using Claude*

[Video description begins] *Topic title: Publishing and Remixing Artifacts. Presented by: Joe Holmes.* [Video description ends]

Now that we've covered the basic ways to create Artifacts, let's see what we can do with completed Artifacts created both by ourselves and other people. When you've completed an Artifact you're happy with, you can publish it to share it with the wider world.

You do so by pressing the 'Publish' button on the bottom right of the Artifact screen. This generates a public URL to share the Artifact with others. Be sure, however, to only publicly publish Artifacts that you're okay with the whole world seeing. In addition to publishing your own Artifacts, you can remix other people's Artifacts.

You do so by selecting the 'Remix Artifact' button on the bottom right of the screen of a published Artifact. Doing so will send you to an active Claude chat, which will take the Artifact and propose alternative remixing ideas, allowing you to provide your own unique spin on the Artifact. Let's give them both a try. Let's revisit our HTML website Artifact and see how we can publish this resume website as a live link on the Internet.

It's important to note that publishing Artifacts is only available on personal Claude plans— that is, Calude Free and Cloud Pro plans— but is not available on Claude for Work plans provided by your employer. This is for security reasons. Now, to publish this Artifact on our Cloud Pro plan, we'll go to the bottom right-hand corner of the screen and click the Publish button.

This will pull up a modal dialog box, giving us the option to both publish the Artifact and copy the link to the published Artifact to our clipboard. [Video description begins] *The screen displays a small pop-up window with the heading Publish Artifact. There's a cautionary note reminding the user to ensure no personal or confidential information is included. A selection option is present, labeled Unpublished, signifying that currently only the user has access. There is also a mention of a usage policy against harmful content. A button at the bottom offers the option to Publish & Copy Link.* [Video description ends] Now, we can open a new tab and visit a live link to our Artifacts website hosted on the Claude platform. Next, we'll find out how to remix Artifacts. Revisiting our solar simulation Artifact, let's assume we've published it and are now ready to remix it from the live URL to the Artifact.

We can go to the bottom right-hand corner of the screen and click Remix Artifact. It's important to note that unlike the Publish Artifact option, the Remix Artifact option is available across all Claude memberships. Now clicking the button, I'll pull up the window to the Claude chat and Claude will get to work doing two things: first, it will reproduce the Artifact how we discovered it in the published version; second, it will suggest three ways we could remix the Artifact.

In our case, I already know the remix I want to do of the Artifact, and so I'll paste it here: 'Remix this artifact to generate an alien solar system full of extravagantly colored, imaginative looking planets.' I'll

press Enter, and let Claude get to work on our remix. And just like that, Claude creates an imaginative, creative world out of an alien solar system, using our existing Artifact and remixing it.

8. Video: Course Summary (it_ccclauid_01_enus_08)

In this video, we will summarize the key concepts covered in this course.

- *summarize the key concepts covered in this course*

[Video description begins] *Topic title: Course Summary. Presented by: Joe Holmes.* [Video description ends]

In this course, we have offered an overview of Claude Artifacts; given a walkthrough of Document, SVG, Mermaid.js, HTML, and React.js Artifacts; and offered a short guide to publishing as well as remixing Artifacts.

Course File-based Resources

- Document Artifacts

  Topic Asset

- SVG and Diagram Artifacts

  Topic Asset

- HTML Artifacts

  Topic Asset

- React.js Artifacts

  Topic Asset

- Publishing and Remixing Artifacts

  Topic Asset

© 2024 Skillsoft Ireland Limited - All rights reserved.