Steven Williams (AI and Education Specialist, Eden Centre & DSL) –
s.williams10@lse.ac.uk

Guide – Claude Artifacts for LSE Educators

Guide – Claude Artifacts for LSE Educators ***1***

What's new since the last version of this guide? **2**

Introduction **3**

What is a Claude Artifact? How is it different from a regular chat response? 3

What types of Artifacts can Claude create? 4

Why use Claude Artifacts? 5

How to get started creating Artifacts **6**

Creating Artifacts 6

Making changes to Artifacts 7

Testing Artifacts 7

Tips for creating effective Artifacts **8**

Give it a Starting Point 8

Consider Different Screen Sizes 8

Prioritise Accessibility 8

Provide design guidance 10

Adding images 10

Provide guidance for 'AI-powered' features 11

Make sure the Artifact makes sense by itself 11

Sharing Artifacts **11**

Sharing Artifacts internally 11

Collaborating on Artifacts 12

Sharing Artifacts externally 13

Artifact storage **14**

Important Artifact limitations to be aware of **19**

No secure private submission 19

No external API or database access 19

Artifact code can be inspected by any user 19

Examples of Artifacts for different purposes **20**

Ask students to create their own Artifacts **22**

Steven Williams (AI and Education Specialist, Eden Centre & DSL) – s.williams10@lse.ac.uk

What's new since the last version of this guide?

Quite a lot has changed since the last version of this guide. I've made a demo Artifact showing the most useful new feature for educators (storage) and drawing out some practical ways it could be used:

🛠️ Claude Artifact – Learning with Artifact storage (LSE Claude account required)

Here are the headline changes:

- **New Artifact types: Docs, Slides and Design** – you can now make editable documents, slide decks and visual designs such as posters and one-pagers, which you can edit directly as well as by asking Claude. See What types of Artifacts can Claude create?.
- **Storage** – Artifacts can now save data between sessions: personal storage (e.g. saving an individual student's progress) or shared storage (e.g. a class leaderboard, or shared quiz responses). See Artifact storage for examples and some important caveats.
- **Sharing with different access levels** – you can now share an Artifact with 'edit' access, so another person can collaborate on it with you. See Collaborating on Artifacts.
- **Hidden 'teacher view' content** – it's now possible to include content in an Artifact that is only visible to people with edit access, effectively creating a 'teacher view'. If you combine this with 'storage', you can do really useful things like store student submissions or chat logs from your Artifact, and generate cohort-level insights into what they understand well and what they are struggling with – these insights could be used to inform future teaching interventions or brief GTAs. This shouldn't be used for anything high-stakes like exams, but could work well for practice exercises.
- **Images** – you can now give Claude images (currently up to 20MB per image, or 2MB for SVG files – these limits may change) and ask it to include them in an Artifact. These could be free stock images with appropriate attribution, or images generated using the School's Copilot Pro licence. See Adding images.
- **More thorough testing** – Claude now has access to a cloud computer, so it can load an Artifact and click around in a browser to check it actually works, rather than just checking the code looks right. See Testing Artifacts.
- **Better visuals** – the latest models (Opus and Sonnet 5.5 onwards) have much stronger visual capabilities, producing better-quality animations, animated GIFs, interactive 3D models and even short MP4 videos. See Provide design guidance.
- **Sharing links now point to the latest version** – if you ask Claude to update an Artifact after sharing it, students will automatically see the new version without you needing to send a new link. See Sharing Artifacts.

Steven Williams (AI and Education Specialist, Eden Centre & DSL) – s.williams10@lse.ac.uk

There are also some smaller updates throughout the rest of the guide, but these are the changes most likely to affect how you use Artifacts if you read the previous version. For Anthropic's own up-to-date documentation, see What are artifacts and how do I use them? on the Claude Help Center.

Introduction

Welcome to this guide for educators at LSE about creating Claude Artifacts. You'll learn:

- What an Artifact is
- How to create and edit Artifacts
- How to share Artifacts
- Important limitations of Artifacts to be aware of

💡**Tip:** This content is also available as an interactive artifact with a GenAI Assistant feature!

Using the interactive version of this guide

The interactive version of this guide has the same content as this document, plus some features that also show what Artifacts can now do. Open it in Claude, signed in with your LSE account, to use them:

- **AI Assistant** – ask questions about anything in the guide. The Assistant answers only from the guide's content. The first time you use it, Claude asks you to allow it.
- **My prompts** – save any example prompt, edit it to fit your course, add prompts of your own, and download your collection as a Word document. Only you can see your saved prompts.
- **Bookmarks, progress, highlights and notes** – bookmark topics, see which ones you've read, highlight key points and keep private notes on each topic.
- **Prompt library** – share prompts that worked for you with colleagues, and mark the ones you find useful so the best rise to the top. Your name is shown with anything you share, and Claude checks every prompt for inappropriate content or personal information before it's added.
- **Questions the guide can't answer** – if the AI Assistant can't answer your question from the guide, the question is saved without your name, so it can be covered in the next version. Claude checks these questions for inappropriate content before they're saved.

What is a Claude Artifact? How is it different from a regular chat response?

A Claude Artifact is a standalone piece of content created by Claude that appears in a separate panel alongside the main chat. You can also share and access an Artifact via a standalone link. For more details, see Sharing Artifacts.

Claude will sometimes create an Artifact automatically, when what it's making is substantial and self-contained (typically more than about 15 lines) and makes sense on its own without the conversation – for example a long piece of writing, a document or a small interactive tool. You can also directly instruct it to create an Artifact by asking for specific types of content. For more details, see Creating Artifacts.

By default, Artifacts are private, but you can create a sharing link that will work for anyone with an LSE Claude account (LSE staff and students can request a free account using the link on the Digital Skills Lab's Generative AI page, which also includes support emails). This means Artifacts are easier to share than Claude Projects:

- For Claude Projects, you need to directly add people's emails to the Project before the sharing link will work, and you can only add people once they have requested an LSE account. See How to share Projects in Claude.
- For Claude Artifacts, you can simply generate a link that will work for anyone with an LSE Claude account. See How to share Artifacts in Claude.

This means that in practice, Claude Projects can be impractical for large cohorts, and Claude Artifacts are a better alternative if you want to create and share learning resources with large numbers of students.

What types of Artifacts can Claude create?

Claude can create different types of Artifact:

- **Text documents** – longform writing, summaries, structured content
- **Mermaid diagrams** – flowcharts, concept maps, process diagrams
- **SVG images** – vector graphics, illustrations, diagrams
- **Data visualisations** – charts, graphs, interactive data displays
- **Interactive web-apps** – built with HTML or React (a coding library commonly used by app developers)
- **Animated GIFs** – short looping animations of a concept or process that you can drop into lecture slides
- **MP4 videos** – short animated videos, for example a pre-lecture introduction to key concepts (see Provide design guidance for how Claude makes these)
- **Interactive 3D models** – 3D visual explainers that students can rotate, zoom and step through at their own pace
- **Docs** – editable documents such as handouts, reading guides and reports, which you can edit directly and download as Word, PDF or Markdown
- **Slides** – 16:9 slide decks that you can present, edit directly and download (for example as PowerPoint or PDF)
- **Designs** – posters, one-pagers, infographics, mockups and other visuals laid out on a canvas, which you can edit directly and export

💡**Tip:** When you ask for a document, presentation or poster, Claude will usually make a Doc, Slides or Design Artifact automatically. If you need a particular file instead – for example a Word document or a PowerPoint file – name the file type in your request.

💡**Tip:** Anthropic also offers two other types, **Motion** (animated explainers exported as MP4 videos) and **Dashboards**, but these need an LSE administrator to turn them on, so they aren't currently available. Claude can still make MP4 videos with code, as described above.

Example Prompts:

"Create a mermaid diagram Artifact showing the research methodology workflow."

"Build an interactive data visualisation Artifact for the CSV file I've uploaded."

"Make a text document Artifact summarising the key points from our discussion."

"Create an interactive tutorial Artifact explaining supply and demand curves."

"Create a looping animated GIF showing how compound interest grows over time that I can add to my lecture slides."

"Using the lecture slides I've uploaded, create a short animated MP4 video introducing the key concepts for students to watch before the lecture."

"Create an interactive 3D model Artifact of a regression plane with two explanatory variables that students can rotate and explore at their own pace."

"Create a slide deck Artifact introducing Porter's five forces for my first-year strategy lecture."

"Design a one-page A4 poster Artifact summarising the key referencing rules for our noticeboard."

"Create a Doc Artifact with a reading guide for this week's seminar, based on the article I've uploaded."

Why use Claude Artifacts?

There are several reasons why you may want to use Claude Artifacts:

- **Visualisation of complex concepts** – Artifacts can make abstract ideas more concrete
- **Active learning** – Interactive Artifacts can give students retrieval practice, help activate their existing knowledge or prime them for further learning by helping them to identify gaps in their existing understanding.
- **Personalised productivity tools** – You (or your students) can create custom tools tailored to your course or individual requirements

- **Discoverability** – Interactive formats can help students explore information at their own pace
- **Skill development** – Setting students the task of creating Artifacts could help them develop AI literacy and skills like design thinking. For more, see Ask students to create their own Artifacts.

How to get started creating Artifacts

Creating Artifacts

You can tell Claude directly to make an Artifact – for example "Create an interactive quiz Artifact about Sweller's Cognitive Load Theory" or "Make an Artifact that explains the components of Willingham's simple model of the mind in a visual way". Claude can create different types of Artifact: they can be text documents, mermaid diagrams, SVG images, data visualisations, interactive web-apps, animated GIFs, MP4 videos, interactive 3D models, editable Docs, slide decks (Slides) or visual designs (Design).

💡**Tip:** Artifacts can use Claude within them, so you can make things like chatbots, dynamic feedback systems, adaptive tools, and content analysis features that respond dynamically to user input. New Artifacts ask your permission the first time they want to use Claude. The **AI-powered Artifacts** setting in **Settings > Capabilities** only applies to legacy Artifacts (those made in a chat before 16 September 2026). Anyone using your Artifact signs in with their own LSE Claude account, so their usage counts against their own limits rather than yours.

💡**Tip:** In Spring 2026 Anthropic added a feature now called **Inline visualizations** (originally called 'Inline Artifacts'). This means that, for simple things, Claude may show visuals in the body of the chat, rather than in a standalone panel. If this happens and you want a standalone Artifact, just tell Claude to 'change it to a standalone artifact'; you can also directly specify whether you want an 'inline' or 'standalone' Artifact in your initial instructions. Alternatively, you can turn off Inline visualizations in **Settings > Capabilities > Visuals > Artifacts**.

💡**Tip:** Everything you make is saved to the **Artifacts** tab in the Claude sidebar, so you can find your Artifacts again from any conversation.

Example Prompts:

"Create a mermaid diagram Artifact showing the research methodology workflow."

"Build an interactive data visualisation Artifact showing global temperature trends."

"It should be an AI-powered standalone artifact that uses Claude to generate any dynamic feedback for the user rather than simply displaying pre-written responses."

Making changes to Artifacts

It's unlikely that Claude will create a perfect Artifact after one instruction. After it's created an Artifact, test that it works, and consider if you want Claude to make any bug fixes, change any content, or add any features.

How you make changes depends on the type of Artifact. For Docs, Slides and Designs, you can edit directly – type into a doc, edit a slide, or move elements on a design canvas – as well as asking Claude to make changes. In a Doc, you can also highlight some text and choose **Edit with Claude** to change just that part.

For other Artifacts, such as interactive web-apps, you can't make changes directly within Claude – you need to describe to Claude what changes you'd like it to make. For instance, once Claude has created an Artifact, in the same chat give it a follow-up instruction like "Add a home button on each screen". You can also click the **Code** button symbol above the Artifact, select a section of the code, click **Improve** in the pop-up, then type a comment about a specific change you want to make. It's normally best to give it one follow-up instruction at a time.

💡**Tip:** To adapt an Artifact for another module or cohort without changing the original, click the Artifact's name at the top left and choose **Duplicate** to make your own copy.

💡 **Tip:** Every time Claude makes a significant change to an Artifact, it will create a new version. Previously, you had to create a new sharing link for each version, but now the default sharing link will always point to the latest version. For further information, see the Sharing Artifacts section of this guide.

Another way to make changes to Artifacts is to download the Artifact and make manual edits to the file in a program like Visual Studio Code (if it's code), Word (if it's text), or PowerPoint (if it's slides). You can then host or use the file as you wish outside of Claude, or upload the updated file to your chat in Claude and tell it to edit its version of the Artifact to implement your changes.

Testing Artifacts

Artifacts make it easy to create good looking web-apps, but just because it looks good doesn't mean it works! After Claude creates the Artifact, you should always test it thoroughly. You may find things don't work the way you expect, or you may think of additional features you want it to add. Give it follow-up instructions telling Claude what to change. This iterative approach often produces better results than trying to specify everything in a single prompt.

Claude can now do more of this checking itself. It has access to a cloud computer, so it can load the Artifact in a browser behind the scenes and click around to check that it displays correctly and that buttons, navigation and interactive features actually work – rather than just checking that the code looks right. It won't always do this unless you

ask, and it can't fully test features that rely on Claude's own platform, such as AI-powered features or storage, so check those using the shared link. You should still test the whole Artifact yourself too, particularly on a mobile device and with the kinds of input your students are likely to give.

Example Prompts:

"The text overflows the container box when viewed on a mobile. Fix it."

"Before you finish, load the Artifact in a browser on your cloud computer, click through every screen, button and interactive element, and fix anything that doesn't work or display correctly."

💡**Tip:** If an Artifact shows an error message, look for the **Try fixing with Claude** button next to it. This copies the error details into a new message so Claude can diagnose and fix the problem – though some errors need more troubleshooting.

Tips for creating effective Artifacts

Give it a Starting Point

If you simply tell Claude to "Make an Artifact that does X", it will generate all the content itself. You can often achieve better, more context-appropriate results if you provide a starting point, such as uploaded documents, worksheets, academic papers, or web content.

Example Prompts:

"Base the content of the artifact on the uploaded documents."

"Transform the worksheet I've uploaded into an interactive artifact."

"Create a learning game to help people understand my research – I've uploaded my article."

Consider Different Screen Sizes

Claude will often generate Artifacts that work well on large screens, but they may have issues when viewed on small mobile screens. Tell Claude to make its Artifacts responsive – though it is still a good idea to test on different screen sizes yourself.

Example Prompt:

"Ensure the artifact is mobile-first and responsive."

Prioritise Accessibility

By default, Claude may generate Artifacts that are not well set up for people who use assistive technologies like screen readers. It can be helpful to instruct it directly to prioritise accessibility.

Example Prompt:

The Artifact must meet accessibility standards (WCAG 2.2 AA minimum), for example:

* Sufficient colour contrast (at least 4.5:1 for normal text, 3:1 for large text/UI components).

* Full keyboard operability – no functionality that relies solely on mouse/hover or drag gestures.

* Visible focus indicators on all interactive elements.

* Semantic HTML (proper headings, landmarks, labelled form fields) rather than divs styled to look like controls.

* Text alternatives for any non-text content (icons, images, charts).

* Target size of at least 24×24 CSS pixels for touch/click targets.

* No content that flashes more than three times per second.

* Don't rely on colour alone to convey meaning (e.g. correct/incorrect feedback).

Be clear about which elements you're able to inspect yourself via reviewing the code, and what I need to test for myself by interacting with the Artifact.

However, even when you instruct Claude to make an Artifact accessible, you should be aware that making websites and apps fully accessible can be challenging and it's difficult to know how disabled students will experience the Artifact without lots of testing. For this reason, it's important to offer students an alternative format for the content of the Artifact. A simple way to do this is to ask Claude to produce a Word document version of the Artifact's content, with a clear heading structure, and / or a prompt that students could paste into a normal Claude chat to replicate any interactive features of the Artifact. This gives students who use screen readers – or anyone who finds the interactive Artifact format harder to navigate – an accessible alternative that covers the same material.

Example Prompt:

I want to provide users with some alternate formats in case they have issues using the Artifact. Please create:

* a Word document with the content

* a prompt that can be pasted in normal Claude chat that would reproduce the interactive elements of the Artifact via a chat-based interaction.

💡**Tip:** Anthropic has a help article on using Artifacts with a keyboard or screen reader, which is worth sharing with students who use assistive technology.

Steven Williams (AI and Education Specialist, Eden Centre & DSL) – s.williams10@lse.ac.uk

Provide design guidance

Claude can work with visual references to create Artifacts that match your preferred aesthetic. Upload screenshots of designs you like, or create a mock-up in a tool like Adobe Express (or draw one on paper!). You can now also add your own images (see the section below), or for simple visual elements, tell Claude to use Lucide icons or emojis to break up text.

Claude's ability to work with visual information has also improved a lot with the latest models (Opus and Sonnet 5.5 onwards). It can now make much better-quality animations – and even short videos. Claude doesn't have an image or video generation model: to create a video, it draws each scene with code, loads the scenes into a browser on its cloud computer, takes screenshots, and joins them together into a video.

To take advantage of the different types of content Claude can make, try asking it to make:

- An Artifact with a **3D visual explainer** for [concept] that students can step through at their own pace
- An **animated MP4 video** that students could watch before a lecture to briefly introduce the key concepts – upload your lecture slides as a starting point
- A **looping animated GIF** for [concept] that you can add to your lecture slides

Example Prompts:

"Use Lucide icons for the navigation buttons."

"Add a short animation showing how the demand curve shifts when consumer income increases."

Adding images

You can now include images in Artifacts. Upload the images to your chat (currently up to 20MB per image, or 2MB for SVG files – these limits are subject to change) and tell Claude where you want them to appear. Make sure you have the right to use any images you add – for example, free stock images (with attribution where the licence requires it), your own photos or diagrams, or images generated using the School's Copilot Pro licence.

💡 **Tip:** Ask Claude to give every image descriptive alt text (a short text description read out by screen readers) – see Prioritise Accessibility.

Example Prompt:

"Add the images I've uploaded to the relevant sections of the Artifact. Include a short attribution caption under each stock image, and give every image descriptive alt text."

Provide guidance for 'AI-powered' features

If you tell Claude to include an 'AI-powered' feature in your Artifact, it will write the prompt for this feature itself. You can check the prompt it writes in the Code for the Artifact. To improve accuracy and appropriacy, it's often best to tell Claude what the prompt should include. For example, if you are making an informal quiz that has 'AI-powered' feedback, you can tell Claude that the prompt must instruct the LLM to compare the user's input to a model answer you've written, or to only provide feedback on certain aspects.

Example Prompts:

"For the AI-powered feedback, it must compare the user's answer to the following model answer to ensure accurate, up-to-date and context-specific feedback."

"The AI-powered feedback must only comment on the structure of the user's argument, not on content."

"For the AI-powered 'Assistant' feature, the LLM should be passed the entire text content of the Artifact and must be constrained to only answer using the Artifact content rather than its general training data."

Make sure the Artifact makes sense by itself

Because of the way large language models work, when Claude creates Artifacts, it is influenced by the whole history of the chat in which you created the Artifact. This means the Artifact often doesn't make sense if the person hasn't read that chat. You can normally avoid this by instructing Claude to make sure the Artifact makes sense on its own terms.

Example prompt:

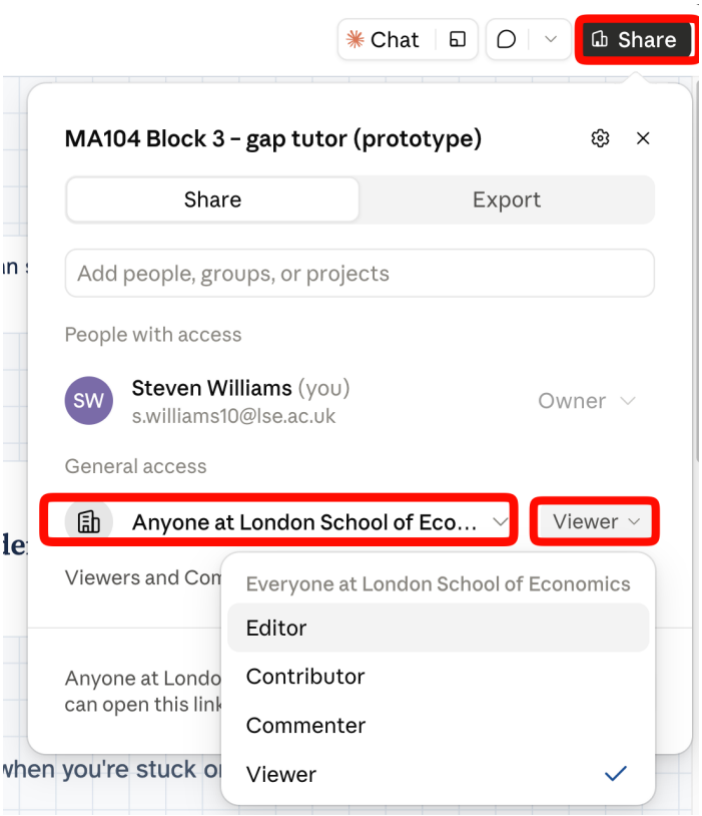
Make the Artifact understandable on its own: write it as a standalone document; include whatever context (names, dates, decisions, terms) the reader needs to understand it without having read the chat that produced it. Explain non-obvious concepts or terminology by glossing inline. Include a brief introduction explaining what the Artifact is and how to use it, so that someone given just the link, with no further context, would know what they're supposed to do. Take a step back and think: will someone who hasn't read this chat understand the Artifact?

Sharing Artifacts

Sharing Artifacts internally

To the top-right of the Artifact there is a Share button. Click this to view the Sharing menu. You can add people directly using their email address to give individuals access (they'll receive an email notification), or create a link that will work for anyone with an

LSE Claude account. For the latter option to work, you need to make sure the chat where you created the Artifact isn't in a Project – if it is, first click the arrow next to the chat name, select 'Remove from Project', and then create the sharing link.



Every time Claude makes a significant change to an Artifact, it will create a new version. By default, the sharing link will now point to the latest version (previously this wasn't the case). This means you only need to create one sharing link, and you can make changes or fixes to the Artifact in the future without needing to re-share it.

💡 **Tip:** For an LSE-specific video about sharing Claude Artifacts, see Sharing Claude Artifacts on the Digital Skills Lab Learning Hub.

💡**Tip:** Sharing settings can only be changed in Claude on the web or Claude Desktop, not in the mobile apps. For full details of who can open a shared Artifact and what they see, see Anthropic's guide to sharing artifacts.

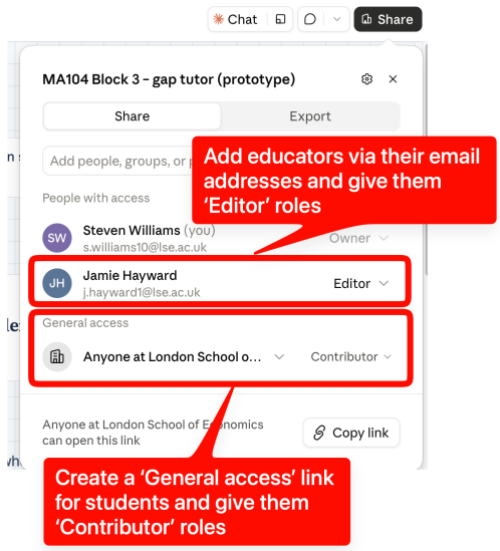
Collaborating on Artifacts

When you share an Artifact, you can now choose different levels of access. Giving someone 'edit' access means they can collaborate with you on the Artifact – useful if you're developing a resource with a colleague.

If someone has shared an Artifact with you with edit access, open it and select the **Chat** button to start editing. Claude will copy the Artifact into a new conversation, and once it

has made the changes you ask for, it will ask for your permission to publish them back to the original link.

Edit access also makes it possible to include content in an Artifact that is only visible to people with edit access – effectively letting you create a 'teacher view', for example with model answers or facilitation notes. If your Artifact includes Claude chats, you could also have a feature that stores the chats in the teacher view and summarises what students are struggling with, so that GTAs can focus on those areas in remediation. Let students know you're doing this before they use the chats, and remind them not to include anything personal. For this to work, give the educators 'Editor' access, and students 'Contributor' access.



💡 **Tip:** A 'teacher view' shouldn't be used for anything high-stakes like exams or assessed work, but it can work well for practice exercises.

Example Prompts:

"Add a 'teacher view' that is only visible to people with edit access, showing the model answer for each question and some suggested discussion points. Tell me how to share this with students and educators with the correct roles."

"Save each student's chat with the tutor so only people with edit access can see it, and add a teacher view that summarises the topics students are finding hardest, for our GTAs to focus on in seminars."

Sharing Artifacts externally

The reason that Claude Artifacts can only be shared internally is a security measure: depending on the Artifact, it could leak information from your chat, or be vulnerable to cyber attacks. If you understand the risks and are prepared to take responsibility for hosting and maintaining the website or app yourself (including any data it might collect),

you can ask Claude to reformat your Artifact into a single file that you could host externally, for example via GitHub or Netlify. Note that it's not possible to maintain any 'AI-powered' functionality when doing this, as it would require you to set up and pay for your own API key – Claude sorts this out for you behind the scenes when it hosts Artifacts for you internally, and the cost is included in LSE's licence.

Example Prompt:

"I want to be able to export the final artifact as a single HTML file and host it externally: Use vanilla HTML, CSS and Javascript and don't use external libraries so that I don't need to maintain dependencies. Make sure it's secure against cross-site scripting and other potential cyber attacks. Make sure it doesn't ask for or store any user data so I don't have to worry about retention policies. Walk me through how to host it on GitHub as a GitHub Pages site and make sure I understand any potential risks."

Artifact storage

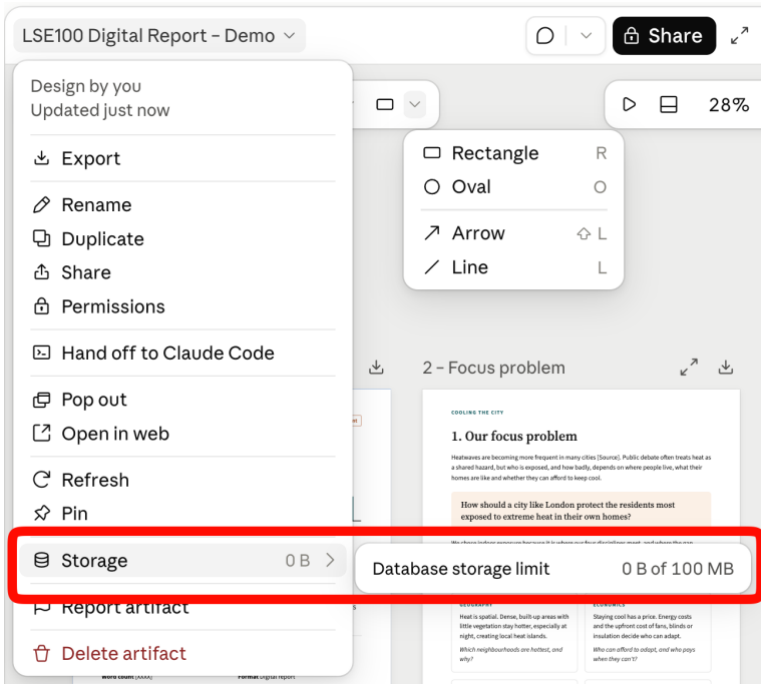
One shortcoming of Claude Artifacts has been that they previously had no 'storage' – there was no way for an Artifact to save user details or progress between sessions. Anthropic added this feature in recent months, and it opens up some interesting new possibilities.

There are now two kinds of storage available:

- **Personal storage** – data only visible to the individual user, for example allowing students to save their progress through a learning resource
- **Shared storage** – data visible to everyone using the Artifact, for example allowing a class to submit scores to a leaderboard, or contribute answers to discussion prompts that can be voted on or discussed further in a seminar

You can see how much storage an Artifact is using by clicking the Artifact's name at the top left and choosing **Storage**. At the time of writing, newer Artifacts have a database storage limit of 100MB each. Anthropic's help article gives a limit of 20MB of text only, which is likely to apply to legacy Artifacts (made before 16 September 2026). Images and other files are stored separately (see Adding images). Personal and shared storage are kept separate, and the first time someone uses an Artifact with shared storage, Claude shows them a confirmation explaining that their data will be visible to others.

Steven Williams (AI and Education Specialist, Eden Centre & DSL) – s.williams10@lse.ac.uk



💡**Tip:** For legacy Artifacts (made in a chat before 16 September 2026), storage only works once the Artifact is published, and **unpublishing a legacy Artifact permanently deletes all its stored data**. Newer Artifacts can store data without being published first.

Examples of how Artifact storage can be used

Here's an example of an Artifact that uses personal storage to allow a student to save their chats and feedback from a role-play and download them to submit to a teacher (via email or Moodle – Artifacts aren't a secure way for students to submit work – see No secure private submission):

Link: ⚒️ Role-play artifact with storage of chat history and feedback (LSE Claude Account required)

Here are two more examples of what's now possible with storage:

Link: ⚒️ Twelve months of generative AI – a quiz (LSE Claude Account required)

Students answer ten questions in their own words, and Claude marks each answer against a model answer and gives instant feedback. Before an answer is saved, Claude checks it for offensive or inappropriate content. Each student sees only their own answers, while a teacher view for people with Editor access shows everyone's answers and Claude's analysis of what the group understands well and what it's struggling with – useful for planning your next session or briefing GTAs.

Link: ⚒️ Maths game with shared leaderboard (LSE Claude Account required)

Steven Williams (AI and Education Specialist, Eden Centre & DSL) – s.williams10@lse.ac.uk

A lighter example showing shared storage used for a live class leaderboard – students' scores persist and update in real time across sessions.

💡 **Tip:** When creating Artifacts with 'shared storage' (anyone with access to the Artifact can see what is submitted), there are content moderation considerations – the example Artifacts above approach this in different ways. The generative AI quiz uses Claude to check each answer for offensive or inappropriate content before it's saved; the maths game asks users to select names from a pre-set list rather than entering free text. If your Artifact lets students type anything in – answers, comments or names – you can tell Claude to scan each entry before it's saved or shared, and flag anything inappropriate so the student can edit it or you can review it. No automatic check is perfect, though, so keep an eye on what comes in. It's also worth reminding students to check whether an Artifact uses shared storage before they enter anything sensitive.

💡**Tip:** Another option is to have the Artifact show people's names from their LSE Claude accounts, rather than letting them type a name in. This avoids inappropriate or fake names, but it means everyone's real name is visible to others, which some students may find off-putting on a leaderboard. You could instead show randomised names to students and real names only in a 'teacher view' (see Collaborating on Artifacts).

Example Prompts:

"On the leaderboard, show each player a randomised nickname, but show their real name from their Claude account in the teacher view."

"Before saving anything a student types in, use Claude to check it for offensive or inappropriate content. If an entry is flagged, don't save it – ask the student to edit it and try again."

Ready-made prompts

The generative AI quiz above includes two ready-made prompts for making similar Artifacts for your own course. Copy one into a new Claude chat and send it as it is. Claude will ask you about your course and ask you to upload your teaching materials before it builds anything.

Make a quiz like this for your course

Check the model answers and marking guides carefully before you share the quiz – the feedback is only as good as they are.

Prompt:
I'd like you to build me a published Claude Artifact: a short retrieval quiz for my students, with free-text answers that you mark against model answers, and a teacher view for me.

Before you start

- First, ask me a few questions, one or two at a time, so you have what you need: the course name and code, who the students are and roughly how many, the topic the quiz should cover, when students will do it (for example after a lecture or before a seminar), how many questions I'd like (suggest 5–6), and whether a question should connect this topic to an earlier one.
- Ask me to upload the teaching material the quiz should be based on, such as slides, a reading or lecture notes. Don't write any questions until I've shared it.

Content
- Write questions that need short free-text answers, not multiple choice. Base them only on the material I upload.
- Start with one or two questions most students should get right, then build up the difficulty. If I've named an earlier topic, include one question that connects to it, and include none that rely on topics we haven't covered yet.
- For each question write a model answer and a marking guide saying exactly what earns 0, 1 or 2 marks. Show me these before you build anything so I can check them.

How it should work
- Before students check an answer, ask how confident they are: Guessing, Unsure, Fairly sure or Certain.
- When they check, Claude marks the answer against the model answer and marking guide only, and gives 2–3 sentences of feedback: what they got right, what they missed, and – if they missed something – a prompt to explain the idea in their own words. If their confidence and mark don't match, point this out kindly.
- Save each student's submissions so they can come back later. Students see only their own submissions.
- Before saving an answer, check it for offensive or inappropriate content. Don't mark or save anything flagged; ask the student to edit it instead.
- Give Editors (me and my teaching team) a teacher view with every submission, the average mark and average confidence for each question, and a button that asks Claude for an anonymised summary of what the group understood, where it struggled, common misconceptions, and what I could do in the next session.
- Use British English and make it easy to read on a phone.

When it's published, tell me how to share it so students are Contributors and colleagues are Editors.

Make a role-play for your course

The same features work for activities with no single correct answer. Claude plays the other side of a conversation – for example a client, a stakeholder or a sceptical examiner – and gives feedback against your rubric, and each student's full conversation is saved for the teaching team. Tell students their conversations are saved and who can read them, keep personal information out of the scenarios, and seek ethics approval

before using the logs for research. Check the character brief and rubric before you share it with students.

Prompt:
I'd like you to build me a published Claude Artifact: a role-play in which my students practise a skill with you playing the other side, with feedback against my rubric and a teacher view for me.

Before you start
- First, ask me a few questions, one or two at a time, so you have what you need: the course name and code, who the students are and roughly how many, the skill they should practise (for example interviewing a client, negotiating with a stakeholder or defending an argument to a sceptical examiner), the character you should play, and roughly how long it should take (suggest 10–15 minutes).
- Ask me for my rubric, or help me write one with 3–5 things to look for (for example: asks open questions, tests assumptions, uses evidence from the case, considers counter-arguments).
- Ask me to upload the material the scenario should be based on, such as a case study, reading or brief. Don't write the scenario until I've shared it.

The scenario
- Play the character I describe, based only on the material I upload. Keep the character consistent, realistic and challenging but fair, and don't break character unless the student asks for help.
- Write a short brief for students explaining their role, what they're trying to achieve and roughly how long it should take.
- Don't introduce information or concepts that aren't in the uploaded material or that we haven't covered yet.
- Show me the character brief and the student brief before you build anything so I can check them.

Feedback
- When the student ends the role-play, first ask them to rate how well they think they did on each rubric point and what they'd do differently. Then give feedback against each point, quoting short moments from their conversation as evidence, and suggest one thing to try next time.
- If they want another go, let them repeat the role-play with a slightly more challenging version of the character.

Saving and the teacher view
- Save each student's full conversation, self-assessment and feedback so they can come back later. Students see only their own.
- Give Editors (me and my teaching team) a teacher view showing who has completed the role-play, each full transcript, and a button that asks Claude for an anonymised summary across the cohort: approaches most students took, points where many got stuck, the rubric points the group found hardest, arguments almost nobody considered, and short excerpts showing strong practice I could discuss in class.
- At the start, tell students that their conversations are saved and that the teaching team can read them.
- Use British English and make it easy to use on a phone.

When it's published, tell me how to share it so students are Contributors and colleagues are Editors.

Important Artifact limitations to be aware of

No secure private submission

By default, Artifacts aren't set up for private submission. Shared storage is visible to everyone using the Artifact, and personal storage is only visible to the individual student – so not to you either.

However, you can ask Claude to set up an Artifact so that students' responses are saved where only people with edit access can see them, and shown to you in a 'teacher view' (see Collaborating on Artifacts). This works well for low-stakes activities like exit tickets or practice answers. Because Artifact features are still changing, don't rely on this for assessed work or anything personal or sensitive. Don't try to make submissions private by adding a password, either – anyone can get around this by viewing the Artifact's code.

For anything students need to submit formally, it's best to build in a feature that lets them download their work as a file – for example a Word document, PDF or plain text file. Students can then send this to you by email or submit it via Moodle.

Example Prompts:

"Add a button that lets students download their conversation with the chatbot role-play from the Artifact as a Word document, so they can submit it to me via Moodle."

"Save each student's exit ticket answer so that only people with edit access can see the responses, and show them in a teacher view."

No external API or database access

At LSE, Artifacts can't connect to external APIs, apps or databases. All data and functionality must be self-contained within the Artifact itself.

Example Prompt:

"Summarise these ten PDFs and create a searchable database within the artifact code."

Artifact code can be inspected by any user

Due to the way that large language models work, when Claude creates Artifacts it is influenced by the whole history of the chat in which you created the Artifact. This means that it often includes as comments in the code things from the chat that you don't want the end-users (students) to see. You can look at the code yourself, or tell Claude that the Artifact will be shared with students, and you want it to remove any comments that may include private information (or just tell it to remove all comments):

Steven Williams (AI and Education Specialist, Eden Centre & DSL) – s.williams10@lse.ac.uk

Example Prompt:

"This Artifact will be shared with students. Please review the code and remove any comments or content that reference our conversation, personal information, or anything else that shouldn't be visible to them."

Examples of Artifacts for different purposes

Here are some examples of how you could use Artifacts.

Using Artifacts to explain complex concepts

Description: You could use Artifacts to create visual, interactive explanations of complex topics. The links below are to two Artifacts that explain concepts related to working effectively with LLMs (this was split into two as the initial Artifact became too big – a workaround for Artifact size limitations). These have interactive activities with preset answers, as well as an LLM-powered 'Check your understanding' feature that allows the user to answer a free-text question and get dynamically generated feedback. The feedback compares the user's answer to a model answer to ensure the feedback is accurate.

Link: Claude Artifact – LLM Explainer Part 1

Link: Claude Artifact – LLM Explainer Part 2

Using Artifacts to create learning games

Description: You can use Artifacts to make interactive learning games, like this two-player warm-up game in which students take turns to try and explain what different terms mean, or this 'Case Study Simulator' (created by the Eden Centre's Darren Moon) in which students can apply theoretical knowledge to practical scenarios.

Link: Claude Artifact – AI Terms Flashcard Game

Link: Claude Artifact – Case Study Simulator

Using Artifacts to check understanding

Description: Polling tools like Mentimeter are great for quickly checking users' understanding or opinions and displaying the results for everyone to see. However, they are often limited to checking receptive knowledge with multiple-choice questions. As an alternative, you could create an AI-powered Claude Artifact that lets students actively check their productive understanding. They can type answers and get feedback (and you can constrain the feedback to model answers or guidance you provide).

Link: Claude Artifact – Check your understanding

Using Artifacts to make information more discoverable

Steven Williams (AI and Education Specialist, Eden Centre & DSL) – s.williams10@lse.ac.uk

Description: You can give Claude a text document, or a series of text documents and ask Claude to transform them into a visual knowledge base Artifact with AI-features so the users can ask questions about the content. Rather than storing the original documents, the simplest approach is to instruct Claude to extract the key points from them and build these directly into the Artifact's code. This keeps the Artifact self-contained, and means any AI-powered features (like an Assistant that answers questions) can be given the full content every time. As with any Artifact, bear in mind that users can view the code, so don't include anything that shouldn't be visible to students.

Link: Claude Artifact – Claude: Tips for Educators

Link: Claude Artifact – Using Artifacts effectively

Link: Claude Artifact – AI Terms Dictionary

Using Artifacts to create productivity tools

Description: Rather than relying on general productivity tools, you could create Artifacts that meet specific needs for you or your team. For example, this 'Rubric Generator' (created by the Eden Centre's Darren Moon):

Link: Claude Artifact – Demo rubric generator

Use Artifacts to visualise data

Description: You can ask Claude to help you visualise specific CSVs, or tell it the structure of the CSVs you want to work with (the headers, what type of data is in each column) and make an Artifact that allows the user to upload a CSV to be visualised. The example below was created using mock data about the time taken to mark different types of assessment.

Link: Claude Artifact – Data visualiser

Using Artifacts to build interactive role-play simulations

Description: You can turn a written role-play or case study into a self-contained interactive Artifact, where each character is voiced by Claude so no two runs are the same. The example below, 'Project Helios', casts the user as a change-management consultant who interviews six stakeholders, presents to a Programme Board that pushes back, and gets light qualitative feedback – with each attempt saved as a downloadable transcript (making use of the new storage feature).

Link: Claude Artifact – Project Helios change management simulation

Ask students to create their own Artifacts

The examples above are some ideas about ways educators can use Artifacts. But students could also benefit from experimenting with making Artifacts – it's a good way to learn skills like design thinking, and understanding the process behind the creation of the apps they use every day. Here are some ideas for things you could ask students to do:

- Students create interactive explainers of concepts they've learned
- Students build data visualisations to explain their own research
- Students design educational games for their peers
- Students create tools relevant to their field of study (e.g., calculators, simulators)

You could potentially use Artifacts as part of assessments, and ask students to:

- Share their chat history showing their iterative process
- Write a reflection on what they learned while creating the Artifact
- Present their Artifact to the class with explanation of design choices
- Critique each other's Artifacts using specific criteria

Submitting Artifacts for assessment

If you ask students to make and submit Artifacts, ask them to submit a PDF version alongside the Artifact link. This gives you a back-up if the link stops working, and a snapshot of the Artifact as it was when they submitted it, which can help show whether they made changes afterwards.

Moodle can't accept HTML files, but Claude now lets you export Artifacts as PDFs. However, the export uses your browser's print dialogue, so it only captures what's on screen – not anything hidden in drop-down boxes, tabs or collapsed sections. Suggest students include something like this in their prompt when creating the Artifact:

Example Prompt:

"I will submit this report as both a web page and a PDF. Please add a print stylesheet so that when printed, all tabs, accordions and hidden sections are shown in full and in order, navigation and buttons are hidden, each major section starts on a new page, backgrounds are white, and charts show their key values as text rather than relying on hover."