🏠 Script introduction

On this page

Script introduction

Script is an extension of the AITable, which supports you program online and interact with the datasheets through the built-in Script API. With this ability, you can quickly complete all kinds of repetitive work and greatly improve your work efficiency. For example, you can:

- Program to generate your own AITable widget that supports **find and replace** in less than 30 lines of code.
- Program a script to verify the data in the datasheet according to custom rules, such as verifying whether the email address is valid, and finding specified ID number or mobile phone number.
- Program a script to call any API quickly to support GET, POST and other request methods.
- ...

Script helps you **complete code writing** faster to complete all kinds of repetitive and complicated tasks in daily work and life, achieving **fast delivery**!

At the same time, it is also very friendly to developer starters. You just need to be familiar with **JavaScript** and have a **browser** to start!

Difference with widget

If experienced developers see the scope listed above, they may have some questions: It seems that the widget SDK can be done. Is there any difference between the two ways?

Scenes	Script	Widget
Enter a lookup value and replace it with the value I want.	**Experienced developers can complete it in less than 1 hour**, with a standardized UI interface.	Experienced developers will take about **1 day** from 0 to build their own development environment and think about UI layout and interaction.
Convert attachments to URLs or URL downloads to attachments	**Experienced developers can complete it in less than 2 hours**, with a standardized UI interface.	Experienced developers need about **0.5 days** from 0 to build their own development environment and think about UI layout and interaction.
Make a dynamic and interactive chart.	It cannot be achieved, only a declarative standardized UI interface.	It can be achieved, can be fully customized, and can be made very beautifully.
Embed third-party services into the AITable	It cannot be achieved, only a declarative standardized UI interface.	It can be achieved, can be fully customized, and can be made very beautifully.

In terms of scenarios, scripts can complete logical computing tasks and standardized UI rendering faster, while widgets are more suitable for complex and customized UIs and interactions.

What do I need to prepare before developing the script?

- Familiar with the JavaScript
- Prepare a Chromium browser, such as Google Chrome, Microsoft Edge

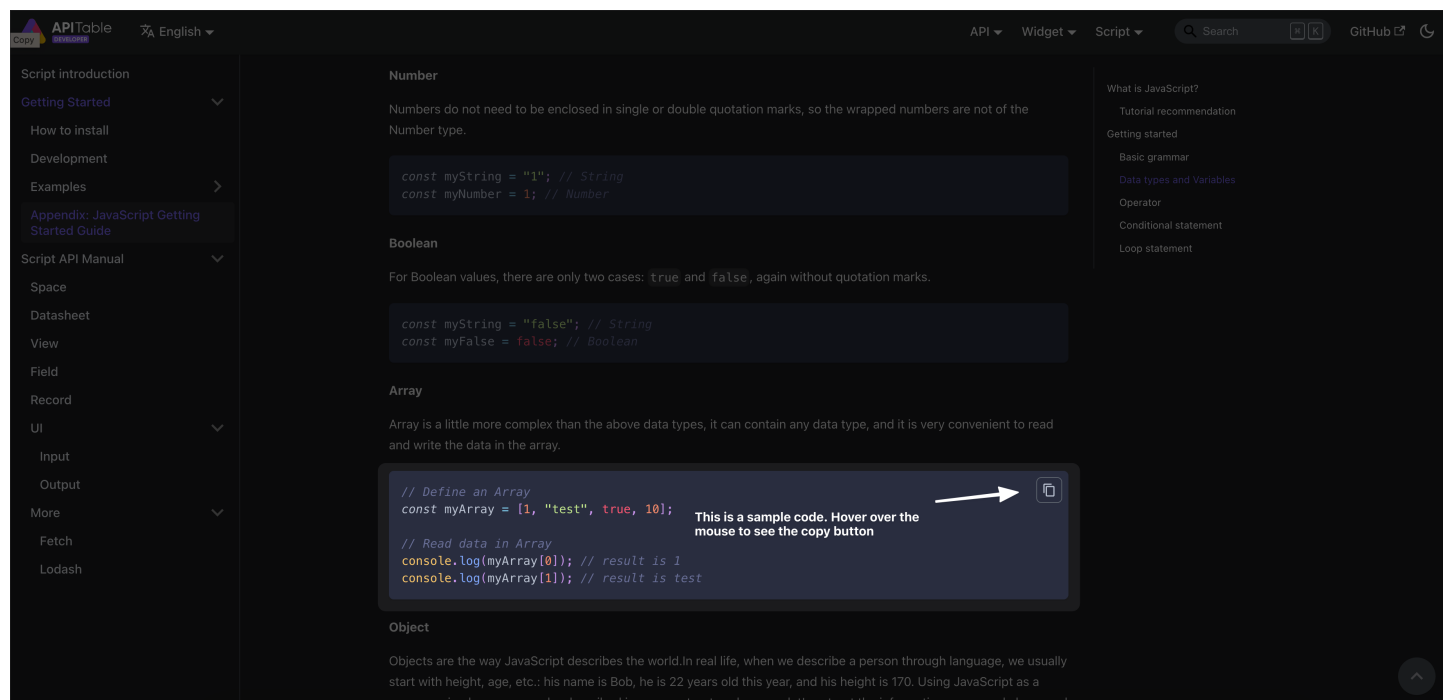# I don't know programming at all. How to get started?

As mentioned above, script is easier to get started than widget. It is recommended that beginners or programming beginners can learn according to the following paths:

1. Installation to learn how to install the script widget and its corresponding functional partition.
2. Practice will teach you how to use JavaScript combined with the Script API to make a search and replacement widget with less than 30 lines of code.
3. If you are not familiar with JavaScript, it is recommended to familiarize yourself with the basic JavaScript syntax through the Appendix: JavaScript Getting Started Guide
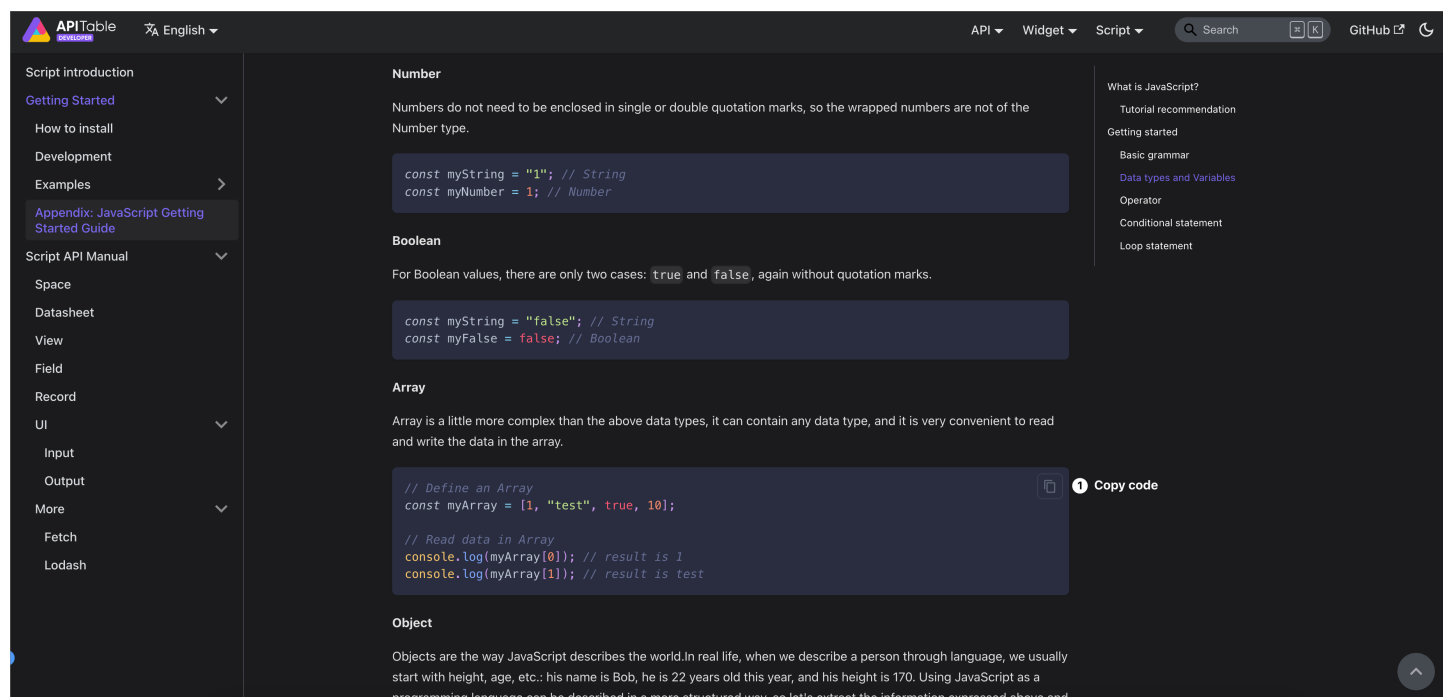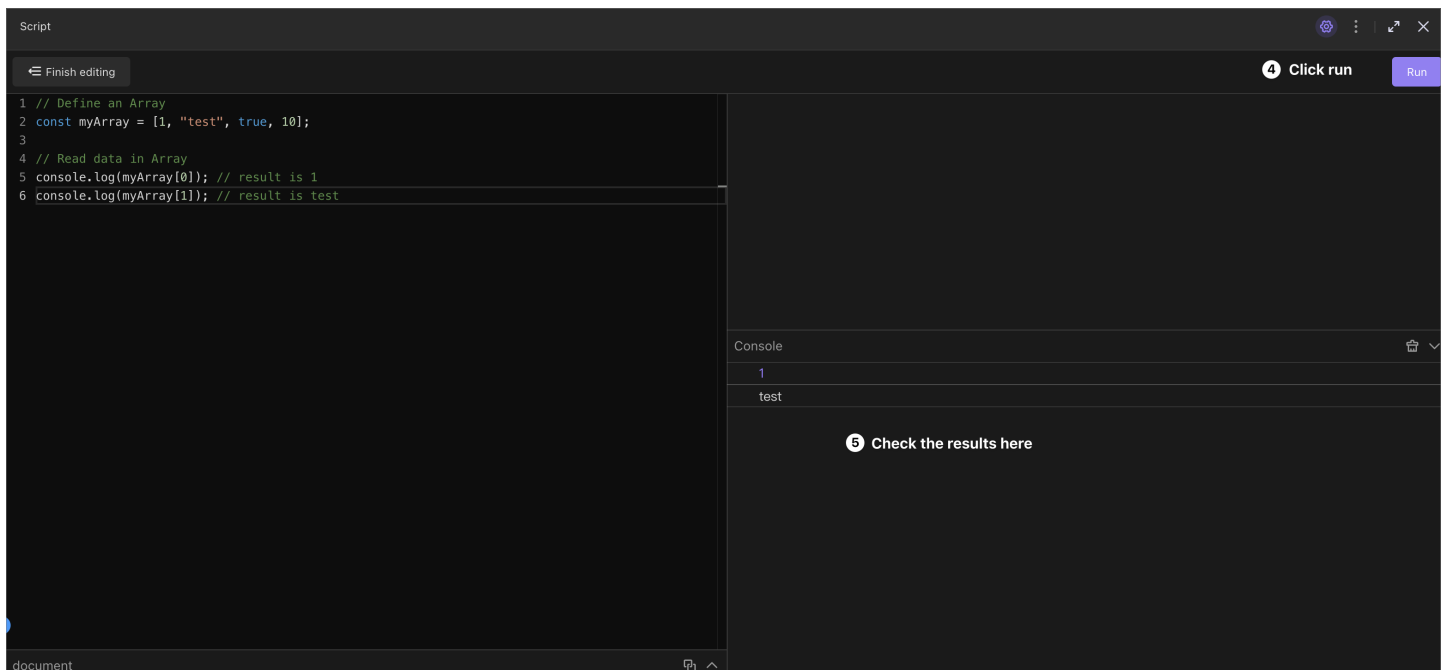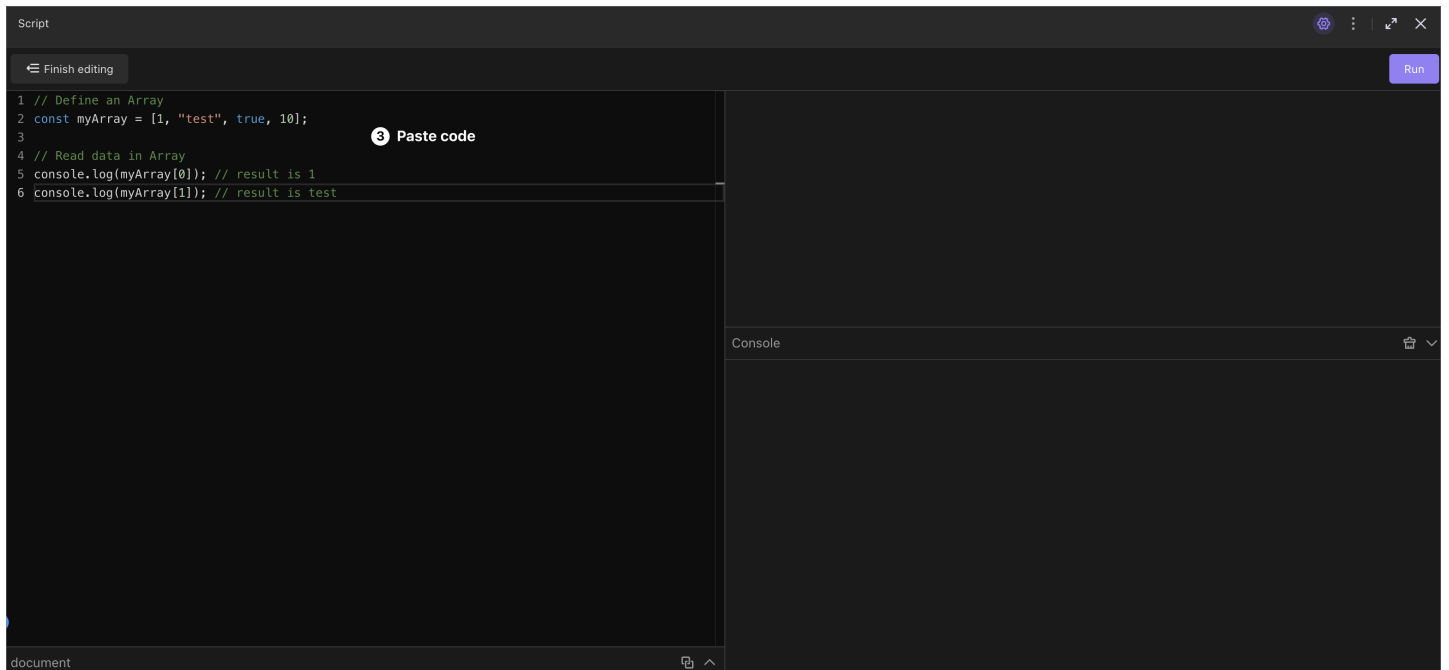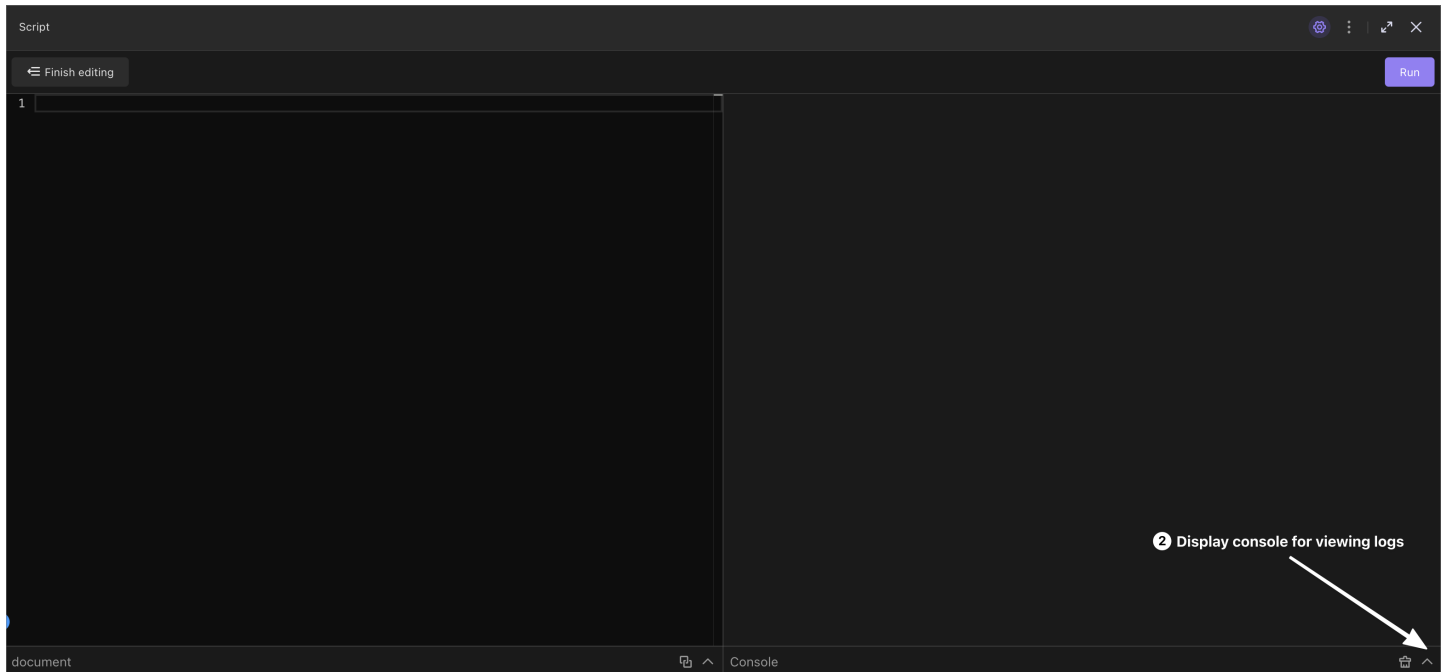
How to run sample code

If you are going to learn how to use scripts according to the above documents, you will find that each chapter contains a large amount of sample code, which can help you learn faster.

First move the mouse into the sample code area to copy it, as shown in the following figure:

The way you run the sample code can refer to the following process:

Script   ⚙ ⋮ ⤢ ✕

[◁ Finish editing]   [Run]

```
1
```

document   Console   ② **Display console for viewing logs**
                                                                        🗑 ⌄

Script   ⚙ ⋮ ⤢ ✕

[◁ Finish editing]   [Run]

```
1  // Define an Array
2  const myArray = [1, "test", true, 10];        ③ Paste code
3
4  // Read data in Array
5  console.log(myArray[0]); // result is 1
6  console.log(myArray[1]); // result is test
```

document   Console   🗑 ⌄

Script   ⚙ ⋮ ⤢ ✕

[◁ Finish editing]   ④ **Click run**   [Run]

```
1  // Define an Array
2  const myArray = [1, "test", true, 10];
3
4  // Read data in Array
5  console.log(myArray[0]); // result is 1
6  console.log(myArray[1]); // result is test
```

                                                  Console   🗑 ⌄
                                                    1
                                                    test
                                                          ⑤ **Check the results here**

document

Developer Center

[Development guide](#)

[API Reference](#)

Social

[Twitter](#) ↗

More

[Homepage](#) ↗

[Help Center](#) ↗

[GitHub](#) ↗