Difference between addEventListener and onclick in JavaScript

In JavaScript, both addEventListener and the onclick attribute are used to handle events,   
but they differ in how they are applied and their flexibility. Here's a breakdown of their differences:

# 1. addEventListener

The addEventListener method is a more flexible way to handle events in JavaScript. It allows adding multiple   
event listeners to the same element and doesn't overwrite existing event handlers. It's the modern standard   
for event handling.  
  
- Syntax: element.addEventListener(event, function, useCapture);  
- Example:   
 button.addEventListener('click', function() {  
 console.log('Button clicked!');  
 });  
- Multiple event handlers: You can attach multiple event handlers to the same event for a single element.  
- Event bubbling and capturing: You can specify whether the event should be captured in the capturing phase or bubbling phase.  
- Removal: You can remove specific event listeners using removeEventListener.

# 2. onclick

The onclick attribute is a simpler way to handle click events but is less flexible compared to addEventListener.   
It directly binds the event handler to the element, meaning it overwrites any previously assigned onclick handler.  
  
- Syntax: element.onclick = function;  
- Example:   
 button.onclick = function() {  
 console.log('Button clicked!');  
 };  
- Single event handler: Only one handler can be assigned to an event. Adding a new one overwrites the previous handler.  
- No capturing or bubbling control: onclick does not allow you to control event propagation phases.

# Conclusion

In general, addEventListener is preferred over onclick because of its flexibility, ability to handle multiple events,   
and support for capturing and bubbling. However, onclick is simpler and may be suitable for small, simple applications.