# Copilot

Understanding the change detection mechanism of the Angular framework involves several key concepts:

1. **Change Detection Implementation**: Angular can detect when component data changes, and then automatically re-render the view to reflect that change[**1**](https://blog.angular-university.io/how-does-angular-2-change-detection-really-work/). This is achieved by patching several low-level browser APIs, such as addEventListener, at startup time[**1**](https://blog.angular-university.io/how-does-angular-2-change-detection-really-work/).
2. **Change Detectors**: Upon application startup, a change detector is created for each UI component and arranged in a tree[**3**](https://dev.to/chintanonweb/the-magic-of-angular-change-detection-a-developers-handbook-42ak). In each change detection cycle, Angular uses a recursive approach with a depth-first search algorithm to identify and re-render the updated component’s view[**3**](https://dev.to/chintanonweb/the-magic-of-angular-change-detection-a-developers-handbook-42ak).
3. **Default Change Detection**: The default change detection mechanism works by adding more functionality to any event handler. Not only is the registered callback called, but Angular is also given a chance to run change detection and update the UI[**1**](https://blog.angular-university.io/how-does-angular-2-change-detection-really-work/).
4. **OnPush Change Detection**: This is another change detection strategy that Angular provides. Understanding when to use this strategy is crucial for building efficient and responsive Angular applications[**5**](https://oncehub.com/blog/angular-change-detection).
5. **Optimization**: Change detection is highly optimized and performant, but it can still cause slowdowns if the application runs it too frequently[**2**](https://bing.com/search?q=understand+change+detection+mechanism+Angular+framework). Therefore, understanding how to control and optimize the change detection mechanism by skipping parts of your application and running change detection only when necessary is important[**2**](https://bing.com/search?q=understand+change+detection+mechanism+Angular+framework).

For a deeper understanding, you may want to read more about these topics from the provided references[**1**](https://blog.angular-university.io/how-does-angular-2-change-detection-really-work/)[**2**](https://bing.com/search?q=understand+change+detection+mechanism+Angular+framework)[**3**](https://dev.to/chintanonweb/the-magic-of-angular-change-detection-a-developers-handbook-42ak)[**5**](https://oncehub.com/blog/angular-change-detection). Additionally, you can also experiment with coding examples and try to modify them to see how change detection reacts to different scenarios. This hands-on approach often provides a more intuitive understanding of how things work.

Source(s)

1. [Angular Change Detection - How Does It Really Work?](https://blog.angular-university.io/how-does-angular-2-change-detection-really-work/)

2. [Angular Change Detection - OnceHub](https://bing.com/search?q=understand+change+detection+mechanism+Angular+framework)

3. [The Magic of Angular Change Detection: A Developer's Handbook](https://dev.to/chintanonweb/the-magic-of-angular-change-detection-a-developers-handbook-42ak)

4. [Angular change detection and runtime optimization](https://angular.io/guide/change-detection)

5. [Angular Change Detection - OnceHub](https://oncehub.com/blog/angular-change-detection)