**Should use:**

1. need a flexible way to define new operations over the existing elements.
2. centralize behavior: Instead of scattering logic across multiple classes, you can put all related logic in one visitor.
3. separate logic from the objects

**Should not use:**

1. need to add new element types often.
2. The operations are closely tied to the element (therefore, visitors might lack the necessary access to the private fields and methods of the elements that they’re supposed to work with)