1. Property: **position**  
   - sets where an element is positioned in document  
   - position: static/relative/absolute/fixed  
   - any position property value (except static) provides additional properties like –   
    (a) top  
    (b) bottom  
    (c) left  
    (d) right  
    (e) z-index  
    we can make use of above properties to position element accurately.
2. **position: static**  
   - an HTML element has position property set as ‘static’ by default  
   - we cannot apply properties – top, right, left, bottom, z-index on a static position element.
3. **position: relative**  
   - with this property we can position the element w.r.t its current position in HTML document  
   - all properties will work - top, right, left, bottom, z-index  
   - example - **top:20px will push the HTML element, 20px downwards** w.r.t its current position. Same for others, **left:20px moves the element towards the right** and vice-versa.
4. **position: absolute**  
   - with this property we can position an HTML element w.r.t its closest parent ancestor (parent, parent’s parent and so on) that has a position property of anything ( except static which by default everyone has)  
   - if no parent ancestor has position property other than static, then our HTML element (that has position: absolute) will be positioned w.r.t HTML document! (ref. file=8position-b.html in code)  
    - above two properties make this property extremely powerful  
   - when this property is applied to an HTML element, it gets removed from the regular document flow. Hence, subsequent elements occupy its space.
5. **position: fixed**  
   - with this property we can position an HTML element w.r.t HTML document  
   - you may think it is similar a special use case of absolute, but the answer is wrong.  
   -