

کلیدی داریم که با فشردن شدن تابع `sortList` را فرا می خواند.

عملکرد تابع `sortList` :

دو متغیر `switching` و `shouldSwitch` را داریم. که `switching` برای این است که ببینیم آیا هنوز لیست نامرتب است یا خیر. و `shouldSwitch` برای این است که آیتم های نامرتب را پیدا کرده و مرتب کنیم.

بدین منظور ابتدا `switching = true` است که حلقه انجام شود. پس از شروع حلقه فرض را مرتب بودن لیست میگیریم مگر اینکه خلاف آن در ادامه ثابت شود.

المان های `li` را انتخاب می کنیم و روی آرایه این المان ها یک حلقه به اندازه تعداد المان های `li` که همان طول آرایه است می نویسم. در حلقه چک می شود که محتوی اولین تگ از تگ بعدی بزرگتر است یا خیر. همچنین با استفاده از متد `toLowerCase()` همه ی حروف را کوچک می کنیم که اشتباهی رخ ندهد و شرایط یکسان باشد.. اگر این شرط برقرار باشد متغیر `shouldSwitching = true` می شود و از حلقه خارج می شویم و چون `i` با `var` تعریف شده است و بلاک اسکوپ نیست در خارج از حلقه به عدد آن دسترسی داریم. سپس با استفاده از `parentNode` به المان پدر المانی که باید تعویض شود می رویم و المان بعد آن را `b[i+1]` را با متد `insertBefore` به قبل آن منتقل می کنیم و سپس `switching` که شرط اجرای دوباره این فرایند است را `true` میکنیم که دوباره المان ها چک شوند.به این شکل ادامه پیدا می کند تا جایی که دیگر `shouldSwitch` صحیح نمی شود و در ادامه آن `switching` هم صحیح نمی شود و فرایند خاتمه می یابد.