1. Компонентный подход к вёрстке помогает сделать её управляемой.
2. Он решает проблему копипаста, т.е. можно использовать один код в разных местах не переписывая его.
3. Решает проблему зависимости компонентов друг от друга (например, при изменении стилей у конкретных элементов или добавлении новых элементов, другие элементы затронуты не будут, и переписывать код не придётся)
4. Делает код понятным и доступным для других разработчиков, если проект будет масштабироваться, либо для поддержки через некоторое время, даже если это будет тот же самый разработчик.
5. Блок — это независимый компонент страницы, который может содержать внутри как элементы, так и другие блоки. Элементы зависимы от блока и не могут существовать вне блока, а также не могут содержать внутри себя другие элементы.
6. Модификаторы используются для описания внешнего вида какого-либо компонента. Это нужно в случае, если есть несколько экземпляров одного компонента, и нам нужно управлять характеристиками только одного из них, а не всеми сразу. Также позволяет избежать увеличение специфичности селекторов. Например, у нас есть 2 блока меню, реализованных при помощи списка. Нам необходимо поменять цвет фона на зелёный у одного из них. При помощи модификаторов мы можем сделать это только для одного из них, а не у обоих.
7. Миксы используются для того, чтобы не копипастить имеющийся код при создании новых компонентов, а переиспользовать уже имеющиеся компоненты и их свойства для создания новых. Также миксы позволяют оставить универсальным блок, который мы миксуем, т.е. не придётся переопределять его стили, увеличивая специфичность, что позволит остаться изначальному блоку универсальным. Например, в header проекта имеется блок поиска “header\_\_search”. В теле проекта нужно сделать форму для заполнения с возможностью поиска “form-search”. Замиксуем новый блок поиска и уже имеющейся блок поиска из header “form-search header\_\_search”. Это позволит нам не увеличивая специфичность применить поведение имеющегося блока для нового, при этом оставив блок “header\_\_search” универсальным для всего проекта.
8. Универсальный селектор применяет стили ко всем компонентам проекта, что делает вёрстку не управляемой. Например, стили могут примениться к компонентам из другого проекта или из библиотеки, хотя нам это не нужно. Также вёрстка по БЭМ предполагает оптимизацию кода, а стили, написанные для универсального селектора, часто переопределяются в других компонентах, что делает написание стилей для универсального селектора бесполезным.