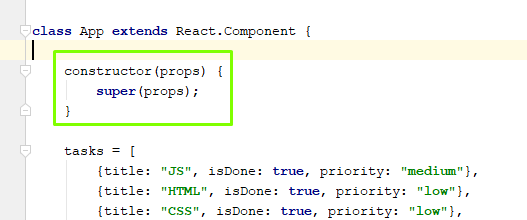
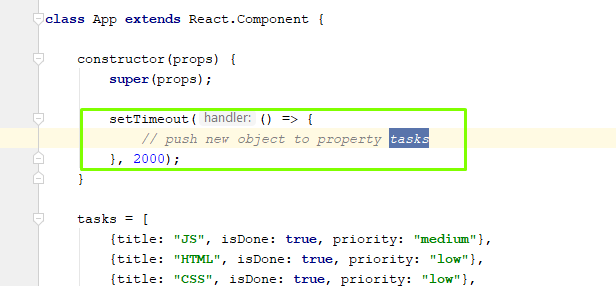
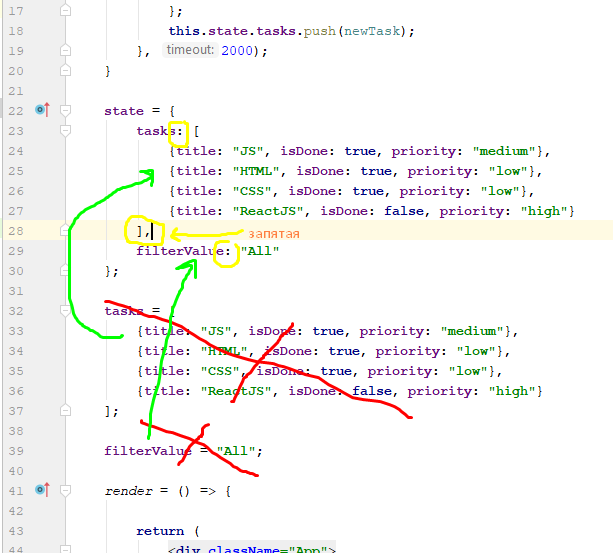
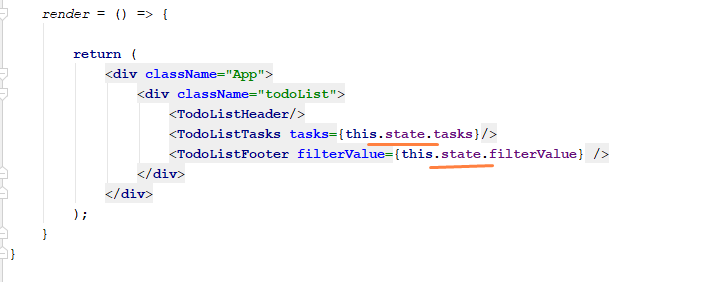
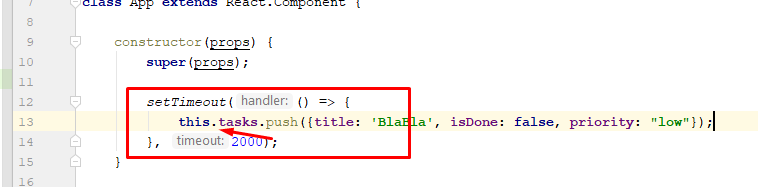
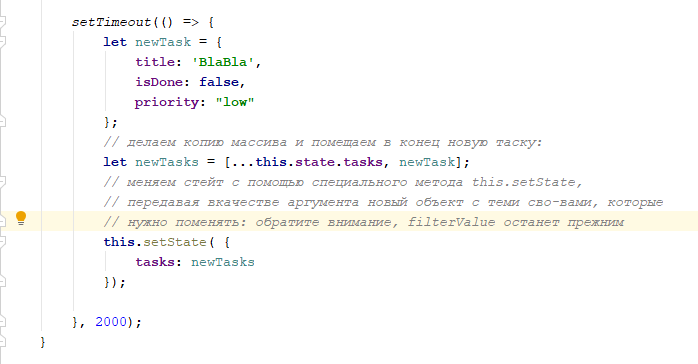
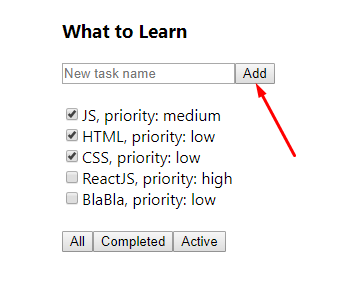
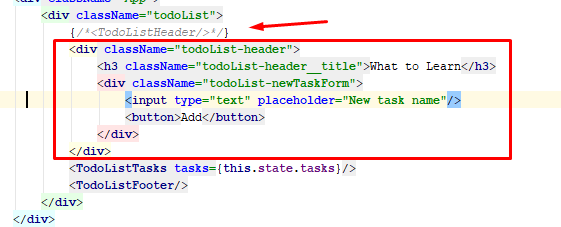
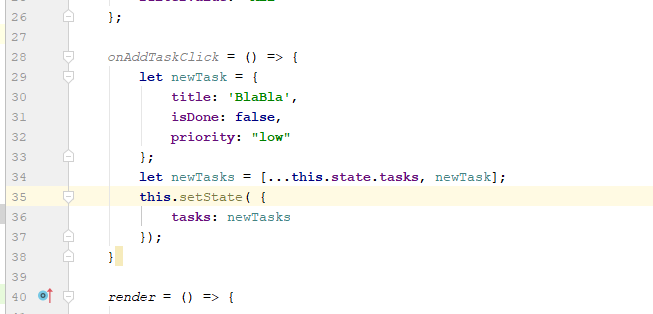
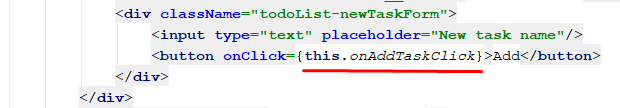
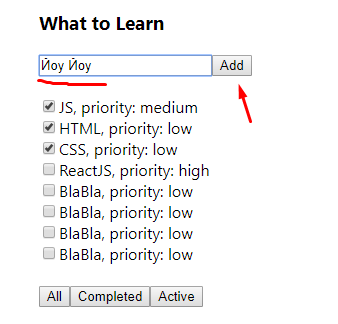
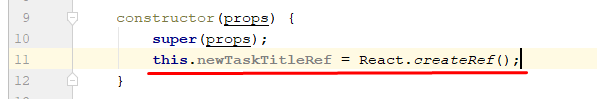
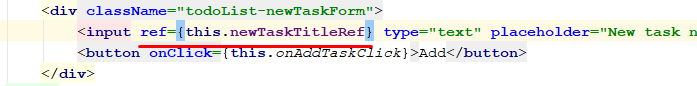
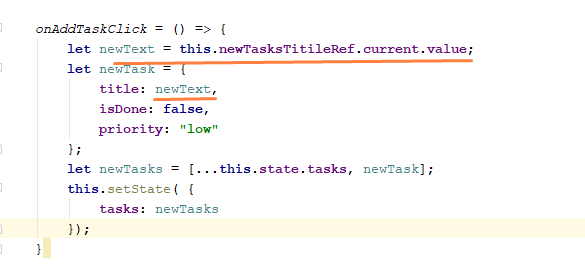
1. добавьте в class конструктор  
     
     
     
   Конструктор - это метод, который вызывается при конструировании объекта.  
   Так как у нас есть наследование (**extends**) мы обязаны в своём конструкторе вызвать конструктор родителя и передать в него то, что пришло в нас (**props**)  
     
   !!! Конструктор вызывается **один раз.**
2. Давайте в конструкторе запустим **setTimeout** и через 2 секунды добавим в массив ещё один новый объект.  
     
     
     
   Объект на самом деле меняется… но визуально ничего не произошло ((   
   То есть **React** не перерисовал наш компонент с новыми данными (то есть не вызвал метод **render**), потому что он НЕ СЛЕДИТ за ВСЕМ объектов и за придуманными нами св-вами **tasks** и **filterValue**
3. Но следит за **специальным** его свойством, которое называется **state.   
   state** должен быть объектом. Поэтому перепишем наши 2 сво-ва, упоковав эти 2 сво-ва в объект, у которого внутри будет сидеть массив **tasks** и **filterValue**:  
     
     
     
   Не забудьте подправить путь к массиву и к **filterValue** вот здесь и здесь:  
     
     
     
     
   Результат: всё отображается, но новая **таска по-прежнему не добавляется**.
4. Свойство **state -** оно особое, но чтобы React **отреагировал** на его изменение, его нужно менять особым встроенным методом **this.setState  
     
   **  
   Результат: через 2 секунды  **this.state** поменяется и **React** отреагирует на это перерисовкой компоненты
5. Давайте добавлять таску по нажатию на кнопку:  
     
     
   Для упрощения задачи, давайте временно закомментируем отрисовку компоненты **<TodoListHeader />,** а её внутренности вытянем наружу (это временно, так как задача всё раздробить на компоненты отдельные - задача номер 1) :  
     
     
     
   Зачем это нужно? Мы же напротив “упаковывали” всё в отдельные компоненты? Мы вернули всё назад временно! Чтобы упростить работу с обработкой клика и изменения стейта. Потом мы всё вернём как было.
6. Повесим обработчик на событие **onClick:  
     
   **Кликните на кнопку и проверьте, что отобразиться **alert**
7. Перенесите код из **setTimeout** по изменению стейта (добавлению таски) в стрелочную функцию внутри onClick (вместе alert-а)  
     
   Если всё хорошо, то при каждом нажатии на кнопку будет добавляться новая таска.
8. Чтобы не засорять **JSX** кодом JS, давайте создадим в классе метод **onAddTaskClick:  
     
   **И на кнопку в качестве обработчика события **onClick** повесим этот метод:  
     
   То есть мы просто зарефакторили (улучшили код). Поведение должно остаться тем же!!!
9. Давайте добавлять таску с названием, напечатанным в поле ввода:  
     
     
     
   Нужно в методе **onAddTaskClick** как-то взять и прочитать значение, введённое в **input.** Для этого нужно создать специальную свойство-ссылку в конструкторе. Эта ссылка умеет ссылаться на элементы в разметке:  
     
     
     
   Чтобы сослаться на КОНКРЕТНЫЙ элемент, нужно в JSX-разметке связать элемент **input** c созданным свойством-рефом **this.newTaskTitleRef:  
     
   **И теперь в методе добавления новой таски мы можем обратиться к этому элементу и прочитать его значение:  
     
   
10. Самостоятельное задание: попробуйте занулить текст в инпуте после добавления таски!