**TodoList需求分解：**

* **动态渲染列表**
* **Input框添加todo**
* **鼠标移入效果**
* **添加一个todo（改变todo的done状态）**
* **对props进行限制类型和必要性的限制**
* **删除一个todo（filter过滤指定idtodo项）**
* **底部功能实现**

**动态渲染列表**

Tips：组件之间的数据传递需要使用App组件（父组件）向子组件传值，传递props

App.jsx=>List.jsx=>Item.jsx

将数据存储在App.jsx的state中的数组里，但是todos的每一个项有多个属性，用于id辨识、事情存储、判断完成，因此数组中的每一项又是一个对象（记住：状态todos存在App上的）

一些文字和图案

描述已自动生成

解构赋值拿到todos，props传到List组件中





List组件中解构赋值接值，然后Item组件复用，动态渲染Item组件



文本

描述已自动生成

这里属性过多，采用批量传递props的方法传递属性，key的标识问题用todo的id解决

Item组件中接值，假数据事件替换为todos的name属性，input框采用defaultChecked属性用于初始化勾选状态（有bug），但是没写死，后面可以点击可以变化，checked就直接写死



文本

低可信度描述已自动生成

**Input框添加todo**

Tips：子组件向父组件传递信息，需要父组件向子组件传递一个函数（props传），子组件通过调用函数传递数据（传值）

App.js(function(a){xxx})=>Header.jsx(function(a)调用)=>App.jssetState=>List改值

从App组件向Header.jsx中传入函数addTodo



传了之后，就来写addTodo函数，addTodo的作用是用于添加一个todo，因此接收到Header传入的值也是一个todo对象，接到数据后解构赋值拿到对象，原有数组加上扩展运算符更新数组，最后setState()完成todos数组更新。

文本

描述已自动生成

App.js中的代码实现后，返回Header.jsx中，给input框插入键盘事件onKeyUp（更精准）



要实现的需求是按下回车键，传入input框中的数据到App.js中，因此得拿到输入内容，且要对回车键进行判断，接着通过调用函数将新的todo对象返回给App.js，更高的需求是判断输入是否为空

文本

描述已自动生成

传入事件对象，结构赋值拿到keyCode和target，Keycode就是键盘的值的ASCII码

注意区分target.value和keyCode的区别，前者是input框中的值，后面是事件对象按下的当前输入值的ASCII码

回车的ASCII码是13，所以采用if(keycode===13)进行判断，只有回车时才执行准备todo对象，传递todo对象的操作

接着就准备todo对象，id需要产生一个唯一标识，可以用时间戳，随机数作为标识，但是都有一定漏洞，这里我们引入uuid/nanoid（yarn add nanoid下载），产生唯一随机数作为id





name是input的输入值，所以是target.value,新的todo都没做，所以done:false

最后调用props上的addTodo函数，把新的todoObj对象传到App.js中，直接调用函数，修改App.js的state，List.jsx的props也随之连续自动修改，需求就已实现

**鼠标移入效果**

在最外层div中绑定鼠标移入和移出事件 onMouseEnter和onMouseLeave

为了不写两个事件回调函数，我们将两个事件都用handleMouse这一个函数进行处理，通过传入true和false进行区分，但是这样就是之前的问题，函数会自身先行调用，因此要设置高阶函数



文本

描述已自动生成

这样鼠标移入移出就会有对应的flag值

接着设置state，给mouse属性初始化默认为false

后续用flag的值进行修改替换



最后运用mouse的true false，结合三目运算符，进行样式的修改替换





true就第一个样式，false就第二个样式

**添加一个todo（改变todo的done状态）**

点击input单选框后，改变item组件中的done属性

先为item中input单选框插入绑定一个onChange事件，指定一个changeCheck高阶回调函数（用高阶函数的原因在于**要传id进去作为标识**），然后用事件对象target的checked属性可以传回勾选状况

接着要实现用item向App传checked的数据了

Item.jsx=>List.jsx=>App.js

因此要用子组件调用父组件函数的方法实现

这里设置一个updateTodo用于更新Item状态

updateTodo中传入id和done两个形参（id用于匹配，done用于接checked值），然后解构赋值从state拿todos，关键步骤在于遍历todos，用map返回改值后新的对象，其实只是覆盖选中的单选框的done，其他的属性都是复制过来的，但是关键点在于id匹配，**用todos中的id和传进来的id’进行匹配比较**，选中的单选框done才会改变，其他的原封不动，这样才行。接着拿新的todos去setState改值就实现了done的改变（checked属性与状态done一致）

文本

描述已自动生成

然后从App.js=>List.jsx=>Item.jsx传递函数，用调用函数的方式传值

App.js=>List.jsx



List.jsx=>Item.jsx





Item.jsx undateTodo() onChange事件对象接到checked值,传id和checked进updateTodo



文本

描述已自动生成

**对props进行限制类型和必要性的限制**

需要下载prop-types库

yarn add prop-types

然后在需要限制的组件中引入prop-types即可



Eg：在List组件中写限制

App.js向List.jsx传了两个props，一个todos数组，一个updateTodo函数，因此限制

文本

描述已自动生成

**删除一个todo（filter过滤指定idtodo项）**

与之前改item done属性的思路基本一致，传递匹配id进行删除

Item.jsx=>List.jsx=>App.jsx

这次btn上绑定单击事件时不用高阶函数回调，直接用内联回调形式，通过回调中调用handleDelete函数，向handleDelete函数中传id实现数据传递



手机屏幕的截图

描述已自动生成

接着就是在App.js中完成要调用的函数，这里我们设置deleteTodo

首先解构赋值拿到todos，接着就是关键之处，使用filter数组过滤方法，滤去与点击对象相同id的对象，然后setState更新todos数组

文本

描述已自动生成

同样把deleteTodo通过props向下传递App.js=>List.jsx=>Item.jsx,步骤同updateTodo一致

最后在点击事件回调调用的函数中调用props传进来的handleDelete函数传递id

但是需要进行一个用户确认的过程，因此采用window.confirm进行用户弹窗确认，同时这个方法会返回一个布尔值，故可以用if进行确认判断，true则执行deleteTodo

文本

描述已自动生成

**底部功能实现**

要更新全部和已完成的数量并要把状态为done的item通过footer的btn删除

从App.js拿到todos用props传给footer.jsx，footer.jsxr里render解构赋值接todos，总数就是todos.length，已完成的用数组方法reduce实现（reduce传两个参数，一个回调，一个初始统计值，回调中也可传两个参数，一个上次调用时的返回值，一个当前调用的返回值。Reduce通过是否done进行计算，done为true +1，false+0，这样拿到done为true的统计值）

文本

中度可信度描述已自动生成

下一步就把数据渲染到下边



下一需求实现：当所有都done时，全选btn自动勾选，这里在checked中进行一次判断，doneCount===total同时total不为0时返回true勾选，不然删完为0了全选还勾选



但是这样会报错，单选框就写死了，四个全部选上时就不能改了，用defaultChecked不报错，但逻辑不匹配，defaultChecked只在第一次奏效，后面为true时不变，所以只能用checked结合onChange事件使用，onChange中回调用于修改状态中done的状态，点击时时所有单选框选中，状态在App身上，所以函数设在App里，props传到footer，footer函数调用时传event.target.checked用于**保持App中状态与footer checked属性一致**，让单选框的同时选中和取消

文本

描述已自动生成



文本

描述已自动生成

注意：react没有双向绑定，子组件只能主动向父组件传参，任何父子通信都需要自己构造双向绑定，调用函数实现

接着再修改原来item单选框使用defaultChecked的问题，改为checked就行

最后做清除所有完成button交互，绑定单击事件到btn，同样为父子通信，回App写函数，要过滤掉done为true的todos，所以用数组方法filter进行过滤，留下done:false 的todos



文本

描述已自动生成

文本

描述已自动生成

todoList相关知识点

1. 拆分组件、实现静态组件，注意className、style的写法
2. 动态初始化列表，如何确定将数据放在哪个组件的state中？

---某个组件使用：放在自身的state中

---某些组件使用：放在他们共同的父组件state中（官方称此操作为：状态提升）

1. 关于父子之间通信：

---【父组件】给【子组件】传递数据：通过props传递

---【子组件】给【父组件】传递数据：通过props传递，要求父提前给子传递一个函数

1. 注意defaultChecked和checked的区别，类似的还有：defaultValue和value

\*\*\*\*\*核心：状态在哪里，操作状态的方法就在哪里