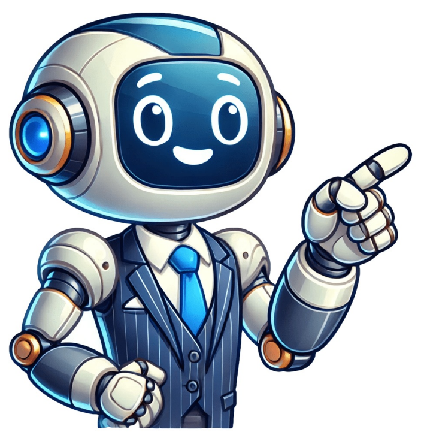

Hello, I am trying to get a button component which is nested in my “CharPic” component to update a value in the state of my App component. I have a handleClick function in my App component which changes the state property menuPage to 0 or 1 depending on the name of a “Button” argument. There is a button in the “CharPic” component: It should change which components are displayed. I am trying to make it like a menu in a game where you can switch between inventory and battle screens. When I change the state property menuPage manually, it works. I can’t seem to get the button to do as I want it. This is what I have so far. I am passing the function handleClick as a property to the “CharPic” component. The “CharPic” component is then passing the handleClick function to the button component it is rendering. This button has an onClick which activates a sendToParent function, which in turn returns handleClick passing the button’s name as an argument. I feel like the info from sendToParent is not making it all the way up to the main App component and is getting stuck in the “CharPic” component, but I can’t figure out how to send it further along. Any help would be greatly appreciated! In my current calculator project (I’m trying to get the handleClick function to take in a div’s id as its argument. The snippet is below: class Buttons extends React.Component{ runParentHandleClick = () => { this.props.handleClick(id) } render(){ return(9 8 7 6) } } This is the only way I’ve figured out how to pass a unique argument into the handleClick function so that it can then display the number in the calculator screen. The problem is that I don’t know how to write the syntax for it properly. If anyone has any ideas on other ways that I could achieve this (without passing in the div’s id) I’m open to trying them out as well. Take a look at the official doc here. reactjs.org A JavaScript library for building user interfaces I’ve read through the official documents and they weren’t of any help. I’ve spent several hours on this problem, going through the official docs, stack overflow and other places and still can’t get it to work. I could try scrapping this approach and copying the approach from the one provided in the main project example, but that would require a several hour redo again. In your hardcoded approach you can just pass any argument you wish (you don’t need id for that): onClick={() => this.runParentHandleClick('5')} In this way parent will know that 5 has been pressed. Is this what you’re trying to achieve? Semantically though, you might think of your buttons as inputs, each with certain value. And then you can do more generic: onClick={(e) => this.runParentHandleClick(e.target.value)} It seems like you are doing this the wrong way, you should be rendering a list of objects instead of hardcoding all the items one by one. Here’s an example: class Buttons extends React.Component { render() { const items = [{id: "six", number: 6}, {id: "seven", number: 7}, {id: "eight", number: 8}. {id: "nine", number: 9}] {items.map(el => {el.number})} } } I am not sure why you are using names for the ids, but this is how you render a list in React. Now you just have to create a method that gets the object that was clicked: handleClick = event => { console.log(event.currentTarget) } At the end it would look like this: class Buttons extends React.Component { handleClick = event => { console.log(event.currentTarget) } render() { const items = [{id: "six", number: 6}, {id: "seven", number: 7}, {id: "eight", number: 8}. {id: "nine", number: 9}] {items.map(el => (this.handleClick(item)}> {el.number}))} } } If this is confusing to you I’d highly recommend you to read the React docs or relearn some of the React main concepts from whatever tutorial you are doing. @snigo I’m doing this for the calculator project, and so I’m trying to pass the e.target.value into the handleClick function’s argument. The problem is that whenever I code it the way you described, nothing shows up. @Gilbert1391 I tried doing it your way and its still not displaying any numbers that I input. Would you be able to take a look at my actual codepen linked above and see if it’s a problem elsewhere. All the code you wrote made perfect sense and I’m sure my understanding of React is more or less up to par, it’s just the syntax that’s messing me up. Also, in the handleClick method you wrote event.currentTarget. Shouldn’t it be event.target.value? I just noticed there’s a syntax error in the array of objects, there’s a dot after the third object and it should be a comma. Yeah I fixed that already and it wasn’t the issue. Whenever the calculator first loads everything displays fine. But as soon as I press any of the keys then the zero disappears from the display and none of the other numbers show up. snigo: In your hardcoded approach you can just pass any argument you wish (you don’t need id for that): onClick={() => this.runParentHandleClick('5')} In this way parent will know that 5 has been pressed. Is this what you’re trying to achieve? I forked your pen and made a few adjustments with the advice from the poster above. I also edited your handleClick function. Maybe this will help you out codepen.io ... Thanks camelcamper, it does seem to help. I just wanted to know why you used the ES6 backticks and $ when defining the this.setState.input. Also if you could walk me through the logic of your changes that would also help. In particular, why arguments are needed in both handleClick and runParentHandleClick and not just one of them. @camelcamper Another thing I can’t seem to understand from your changes is why something like onClick = {this.props.handleClick(“9”)} brings back an error and won’t get the number to display, but onClick = {() => this.props.handleNumbers(“9”)} will work. What does adding the () => into the onClick method actually do? 1 Like In first example you’re passing the result of a function, not the function itself. Try typeof this.props.handleClick(“9”) Embustero: I just wanted to know why you used the ES6 backticks and $ when defining the this.setState.input. it is an easier and nicer way to concatenate strings //ES6 this.setState({ input: `${this.state.input}${x}` }) //vanil JS this.setState({ input: this.state.input + x }) although it doesnt really look like it in your handleClick function, this example concatenating a date string would be much nicer with ES6 backticks let currentDate = "Today is the " + this.date + " of " + this.month" + " " + this.year" let currentDate = "Today is the ${this.date} of ${this.month} ${this.year} Embustero: Also if you could walk me through the logic of your changes that would also help. In particular, why arguments are needed in both handleClick and runParentHandleClick and not just one of them. Your runParentHandleClick is redundant because you are already passing the handleClick function from the parent to the child component, You are setting an attribute named handleClick and assigning it your function this.handleClick so it can be accessed in your child component as a prop with this.props.handleClick So you can delete the runParentHandleClick and use the parents handleClick as a prop. Then you are just passing the number/value of the button as a parameter into the function so it knows which number to update the state with, which then re-renders the screen component. Embustero: Another thing I can’t seem to understand from your changes is why something like onClick = {this.props.handleClick(“9”)} brings back an error and won’t get the number to display, but onClick = {() => this.props.handleNumbers(“9”)} will work. What does adding the () => into the onClick method actually do? This is something I know works but I dont know why and have been meaning to learn, so cheers for the motivation You are passing the number/value of the button as a parameter which forces us to use the parenthesis and this executes the function on render onClick={this.handleClick('9')} You need to pass a reference to the function so if you have a function with no parameters then you can simply write onClick={this.renderNumber} However you are passing an parameter so we can use an ES6 arrow function to wrap around an event handler and pass the parameter () => this.handleClick('9') The react docs on functions that @shimphillip posted explains this and has heaps of other useful info shimphillip: Take a look at the official doc here. reactjs.org Passing Functions to Components – React Embustero: All the code you wrote made perfect sense and I’m sure my understanding of React is more or less up to par, it’s just the syntax that’s messing me up. Also, in the handleClick method you wrote event.currentTarget. Shouldn’t it be event.target.value? Actually you don’t need to pass the event object to the handleClick method as you just want the properties defined in your objects, specifically the id property. You just need to pass an argument as a reference to that object. handleClick = item => { console.log(item) } I wrote the code in a rush with no testing and doing guessing work based on what you said, but it should work unless there is any other typo error. Rect syntax is just JavaScript combined with HTML, it’s not something too unfamiliar. If it’s a bit confusing it’s probably because you are not used to OOP (Object Oriented Programming) and working with objects. I was there myself when I started out and then I decided to do a 6 hours course about OOP, after that I felt way more confident working with React. this is the course in case you are wondering, I highly recommend it. I also noticed you were asking about template literals, which makes me think you are probably not too familiar with ES6 and that could lead to some difficulties. Before moving on I would encourage you to get familiar with some essential JS/ES6 features such as: let/const objects this arrow functions destructuring template literals spread operator classes modules After you have a basic knowledge of all the above, things are going to be way easier. I’m not saying this to sound condescending, I’m just trying to help you out, because we can give you the code you need, but you should be able to understand most of it, otherwise you won’t be able to come up with solutions yourself. By the way, are you using CRA (Create React App) or codepen to build this project?, if it’s the latter, I would advise you to start using CRA to build React apps. Anyways man, building is the best way to learn and that’s exactly what you are doing. If you have more questions I’ll be happy to help. I took a look at your pen, first thing I noticed is that it seems like something is not working because I press buttons and the value does not change at all. But more importantly, the handleClick method has way too many things going on, which makes your app fragile and hard to maintain. Your methods should always be specific, ideally each method should handle 1 thing, this way your code is easier to read and to maintain. I think you need to refactor your code and break this app in small components. I’ve never made a calculator before, but right off the bat this is how my first approach would probably look like: class App extends React.Component { // No need to call constructor to initialize state anymore state { value: 0 } render() { return( ); } } There could be typos as I typed all this directly on here, and of course you might need to change the CSS to make it look the same, but the important thing here is how we are breaking this calculator in small components. You will pass props (properties) to each component based on the props they need. Take the Input component for example, what props does this component need?, well it needs the value property from the state to display in the UI that value as the input value, that is why in the code above I passed the value property from the state as a props. const Input = ({ value }) => { return } This Input component is a functional component, with this implementation whatever the value of the state is, that is what the value of this input element will be. Now I don’t know if in codepen you can work with modules, if not, that’s why I recommended you to use CRA to build React apps, it’s tricky at first, but it’s better you get used to it from the beginning, besides it’s just better for so many reasons. To continue with this implementation, you just need to break the problem into smaller problems, and create a method for each one of those problems. Just let me know if you have further questions. Thanks for all the responses and explanations. Yes I’m working on this in codepen and maybe I’ll check CRA to see if it would make any difference, but honestly codepen seems to be doing the same job as Visual Studio Code when it comes to coding React so I don’t see how CRA would be that much better. Initially I did break my code down into smaller modules but then decided to go a different route. There are some benefits for using CRA over codepen, but if you don’t want to go through the installation and such, you can also use sandbox.io, it is definitely a better option than codepen as it allows you to work with modules. @Embustero I took a look at your code, you have a typo when you are passing the initialize() method as a prop to your Buttons /> component, here’s the typo: initialize = {this.initalize}, it should be “this.initialize”. I would probably define that method as “handleClear” and then the name as a prop would be onClear={this.handleClear} and follow naming convention. If you don’t follow naming convention then you will probably deal with plenty of bugs, you just experienced one in a relatively small app. Page 2 Thanks I really appreciate it. I mean I spent something like 3 hours yesterday going over my code, other peoples code, and tutorial after tutorial to figure out why the clear button wouldn’t work. Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation . No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. Tell us what’s happening: Hello, I’m a bit confused about what is the purpose of this binding function, this.handleClick = this.handleClick.bind(this), in here? Why do we need to mention handleClick on both left and right sides as well as what does bind() do here? Your code so far class MyComponent extends React.Component { constructor(props) { super(props); this.state = { name: 'Initial State' }; this.handleClick = this.handleClick.bind(this); } handleClick() { // change code below this line this.setState({ name: 'React Rocks!' }); // change code above this line } render() { return (Click Me {this.state.name}); } }; Your browser information: User Agent is: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.122 Safari/537.36. Challenge: Set State with this.setState Link to the challenge: freecodecamp.org Learn to code. Build projects. Earn certifications.Since 2015, 40,000 graduates have gotten jobs at tech companies including Google, Apple, Amazon, and Microsoft. 在html标签中使用onclick绑定事件时 报错 Uncaught ReferenceError: handleClick is not defined at HTMLButtonElement.onclick 在html标签中使用onclick如何绑定事件 点击事件 function handleClick() { console.log("--handleClick()") } 接下来先看一下导致报错的代码 点击事件 !(function (window) { function handleClick() { console.log("--handleClick()--") } })(window) 这里导致报错的原因是因为我们这个使用立即执行函数（作用域） !(function () { })() 其实这里报错信息已经给了我们提示 Uncaught ReferenceError: handleClick is not defined 这里告诉我们 handleclick这个方法未定义 在html标签中绑定事件，这个事件的查找方式是在当前html页面的window对象中去查找这个事件方法 点击事件 function handleClick() { console.log("--handleClick()--") } function handleBtn() { console.log("--handleBtn()--") } console.log(window) 接下来我们看一下在立即执行函数中window对象 点击事件 !(function (window) { function handleClick() { console.log("--handleClick()--") } function handleBtn() { console.log("--handleBtn()--") } console.log(window) })(window) 解决方法如下: 点击事件 !(function (window) { handleClick = function () { console.log("--handleClick()--") } console.log(window) })(window) Hi, I cannot call a function with onclick() and would appreciate any help. The js file itself is in a sub-folder (not at the bottom of the html page). One of those is to call a javascript function from a button. The javascript file is in an js-folder, which is in an asset folder. I get the error: (index):55 Uncaught ReferenceError: fillForm is not defined at HTMLButtonElement.onclick ((index):55) Please note that there are other functions on the js page, but I copied the only one that is problematic. All other functions were working as expected. Thank you in advance. JavaScript This is my page Submit addEventListener('load', main); function main(){ removeEventListener('load', main); } export function fillForm(){ let x = document.getElementById("oneInput").value; let text = ""; if (isNaN(x) || x < 1 || x > 10) { text = "Input not valid"; } else { text = "Input OK"; } document.getElementsByClassName("comment").innerHTML = text; } JavaScript errors can be frustrating to debug, especially when dealing with events like onclick. One common error is the Uncaught ReferenceError, which occurs when a function or variable is not defined in the current scope. In this article, we will see the concepts of the onclick function, understand why the Uncaught ReferenceError happens, and discuss how to resolve it.1. ReferenceError in JavaScriptA ReferenceError in JavaScript occurs when you try to access a variable or function that does not exist in the current scope. Essentially, you are referencing something that JavaScript has not yet defined or declared. This typically happens when you misspell a variable name, use a variable before declaring it, or attempt to access something that doesn't exist. ReferenceError Message:Uncaught ReferenceError: something is not definedThis error can also arise when handling onclick events if the function or variable you're calling within the event handler isn't properly defined.2. Understanding the onclick Function in JavaScriptThe onclick event handler is a built-in function in JavaScript used to manage how DOM elements respond when clicked by the user. It allows you to attach a function to be executed when a user clicks on a specific HTML element, such as a button.Syntax:target.onclick = functionRef;target: The DOM element on which the event listener is added.functionRef: The function that will be executed when the event occurs. This function usually takes an Event object as an argument, providing details about the event.When does onclick work?The onclick event occurs when a user clicks on an element. Specifically, it is triggered when the mouse button is pressed and released on the same element.Example: HTML Page Title Click Me! document.getElementById("myButton").onclick = function () { alert("Button Clicked!"); }; In this example, when the button is clicked, an alert box with the message "Button Clicked!" will appear.Example 1: Let's come over an example when Uncaught ReferenceError happens at HTMLButtonElement.onclick. HTML Document Submit function listen(this) { this.style.backgroundColor = 'red'; this.innerHTML = "Hii GFG , How r u???" } Output:Here error arises due to this passing during function definition but inside a function, this refers to the window object not the id of that element. so it creates an issue and returns an error on the console and the program does not give the desired output.Example 2: This is the solution to the above issue that arises in JavaScript: HTML Document Submit function listen(get) { console.log(this) get.style.backgroundColor = 'red'; get.innerHTML = "Hii GFG , How r u???" } Output:Here does not create an issue because passing any random parameter as an argument in the function definition does not pass this as an argument. How to make power button to work? There is a link: I try to execute handleClick() if active (from parent) is true but i don’t know that is possible. You can pass parent state via props to child component, and use it, not change it tho. I mean you could pass function that does change state. Um for example if you have app component that has this.state = { word: "" } and you want to change app state when something happend in child component you can make function that has something like handleWord() { this.setState({ word: "WE got it now!" }) } And then pass handleWord to child component to be used for example on click... Oh and don’t forget to bind handleWord. (your passing function) In your case you would pass parent(app) state and then check if this.props.appState.active is true, if it is you execute your function. Hope that was atleast somewhat helpful, and not THAT wrong, i am not 100% on this tbh. In parent app i have this.setState({ word: "WE got it now!" }) when i press a button i change the word to "something" In child component i have an another button. When i click on this button i want to see in console word: "something" If you just want to access parent state you would simply pass parent state to child. i am not sure if that is what you are going for. Um i think you could do something like... changing state after button is clicked and then use setState callback. setState has callback option, which is called after state is changed, i think, so you could do your loging inside callback if that is an option. Or if that is not, only thing that comes to my mind is to pass parent state and use it when that other button is clicked, like this: And then log word in that child component in another function call function() { console.log(this.props.word) } Sorry that maybe i don’t understand... there is my code: class App extends React.Component { constructor(props) { super(props); this.state = { active: false }; this.handlePower = this.handlePower.bind(this); } handlePower() { this.setState({ active: !this.state.active }); } render() { return (////)} } /// class DrumPad extends React.Component { handleClick = () => { console.log(this.props.active); //undefinited ??? } render() { return ( ); } } Its fine, its my lack of react knowledge and ability to explain, no worries. Hm you’re passing onChange prop to input and not to DrumPad component, therefore DrumPad doesn’t know what onChange is. Try to console.log(this.state) in DrumPad component and you’ll see. For example for DrumPad to know what is onChange you would do something like this: div className="checkboxSwitch">