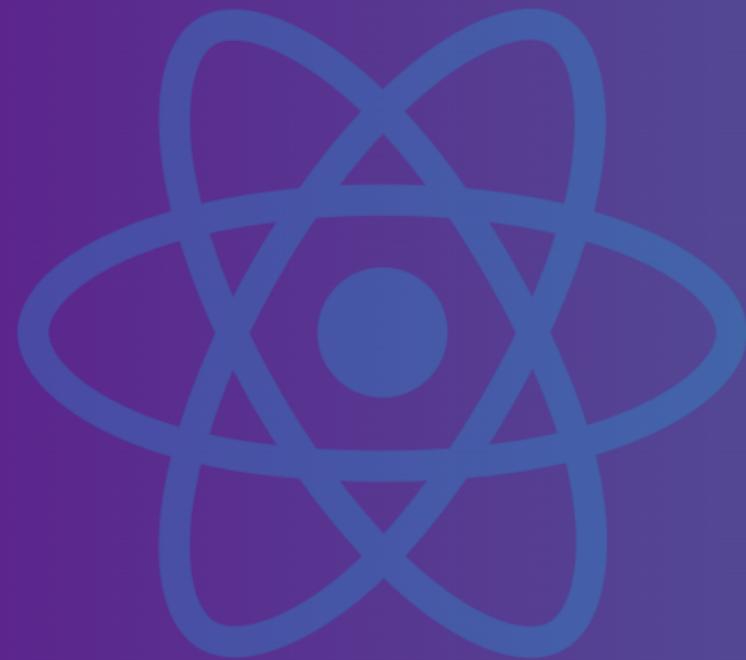


most useful

# VS Code Shortcuts

for beginners



 Sahith Bodla

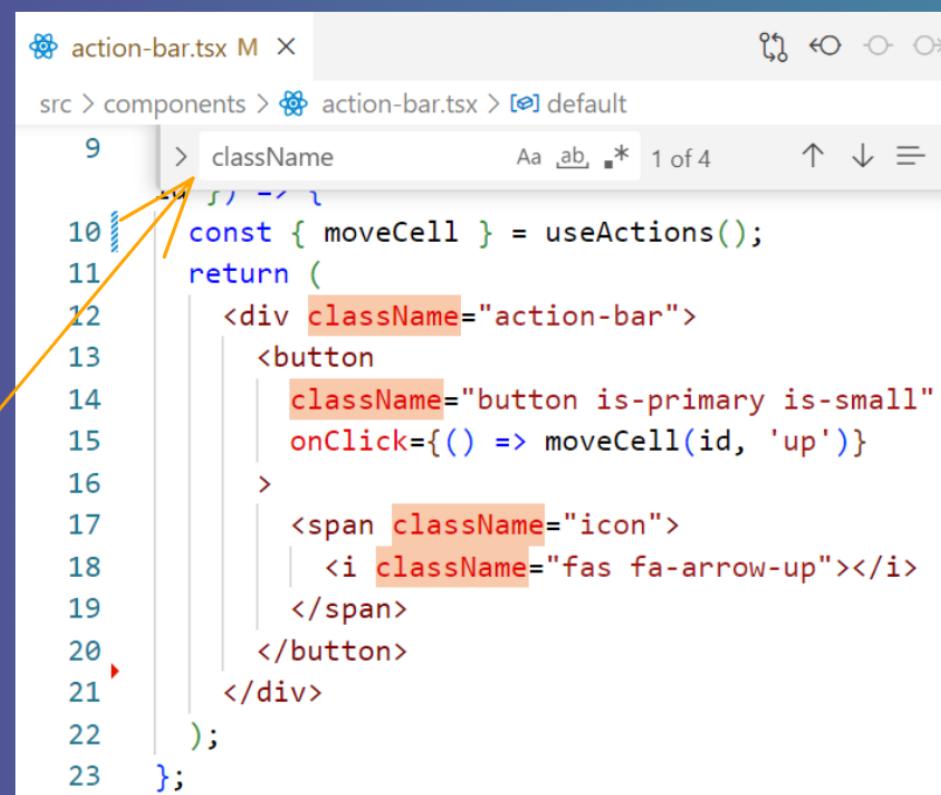


As a devoted Visual Studio Code user, I've discovered key shortcuts that greatly enhance productivity. I've curated ones that streamline tasks and boost efficiency, proving invaluable by saving time and simplifying processes. Excited to share these gems with fellow VS Code enthusiasts

# Search for text in opened file

	
Ctrl + F	Cmd + F

Here, searching  
for a text  
**ClassName**



The screenshot shows a code editor window titled "action-bar.tsx M X". The file path is "src > components > action-bar.tsx > [0] default". The code is a TypeScript component definition:

```
9  > className
10 const { moveCell } = useActions();
11 return (
12   <div className="action-bar">
13     <button
14       className="button is-primary is-small"
15       onClick={() => moveCell(id, 'up')}
16     >
17       <span className="icon">
18         <i className="fas fa-arrow-up"></i>
19       </span>
20     </button>
21   </div>
22 );
23 }
```

A yellow arrow points from the text "Here, searching for a text **ClassName**" in the slide content to the word "ClassName" in the code editor.

# Search for text in all files



Ctrl + Shift + F



Cmd + Shift + F

Here, it show results in all the files

A screenshot of a code editor interface showing search results. At the top, there is a search bar with the text "className". To the right of the search bar are icons for font size (Aa), a magnifying glass (ab), and a refresh symbol (\*). Below the search bar, a message says "27 results in 9 files - Open in editor". The results list shows two files: "action-bar.tsx" and "add-cell.tsx". The "action-bar.tsx" file has 27 results, indicated by a number in a circle at the end of its row. The results in "action-bar.tsx" are:

```
<div className="action-bar">  
  className="button is-primary is-small"  
  <span className="icon">  
    <i className="fas fa-arrow-up"></i>
```

The "add-cell.tsx" file has 9 results, indicated by a number in a circle at the end of its row. The results in "add-cell.tsx" are:

```
<div className={`add-cell ${forceVisible && ...`}>  
  <div className="add-buttons">
```

# Select current line



Ctrl + L

Cmd + L

```
export default ActionBar;
```

```
export default ActionBar;
```

Now, you can copy/delete this line

# Add selection to next match

	
Ctrl + D	Cmd + D

```
<div className="action-bar">  
  <button  
    className="button is-primary is-small"
```

Now, both are selected, if you click again, it will select the next match as well and the selection goes on with this shortcut....

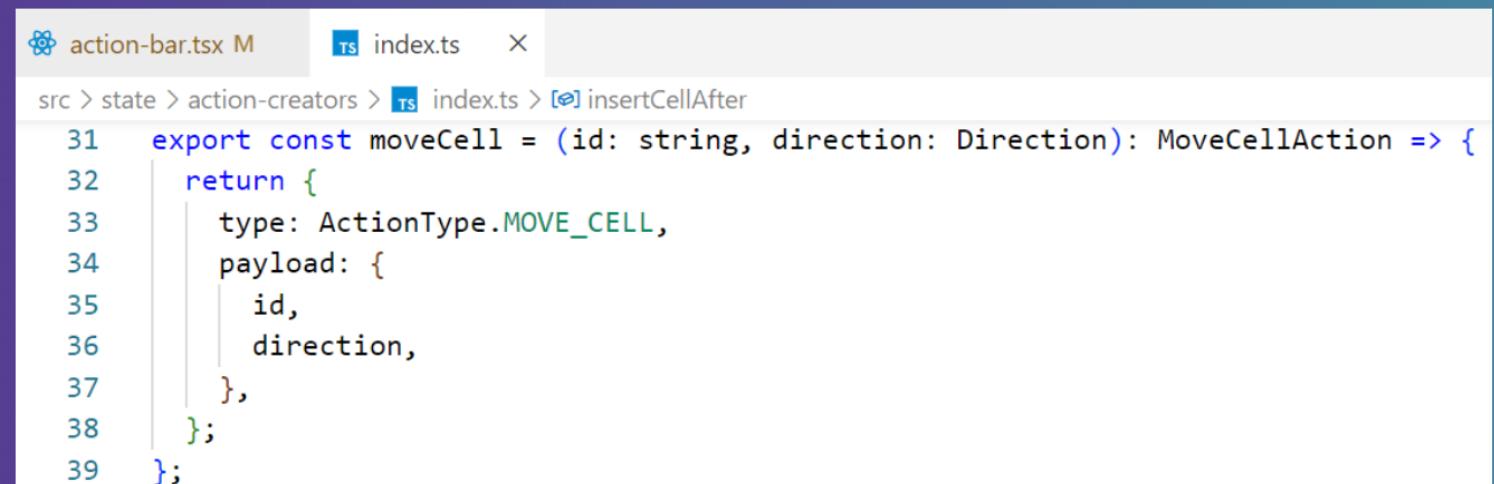
```
<div className="action-bar">  
  <button  
    className="button is-primary is-small"
```

You, 13 m

# Go to Definition

	
F12 or Ctrl + Click	F12 or Cmd + Click

const { moveCell } = useActions();



The screenshot shows a code editor with two tabs: 'action-bar.tsx M' and 'index.ts'. The 'index.ts' tab is active, displaying the following code:

```
src > state > action-creators > [ts] index.ts > [🔗] insertCellAfter
31 export const moveCell = (id: string, direction: Direction): MoveCellAction => {
32   return {
33     type: ActionType.MOVE_CELL,
34     payload: {
35       id,
36       direction,
37     },
38   };
39 }
```

It will take  
you to the  
definition  
of moveCell

# Multiple Cursors

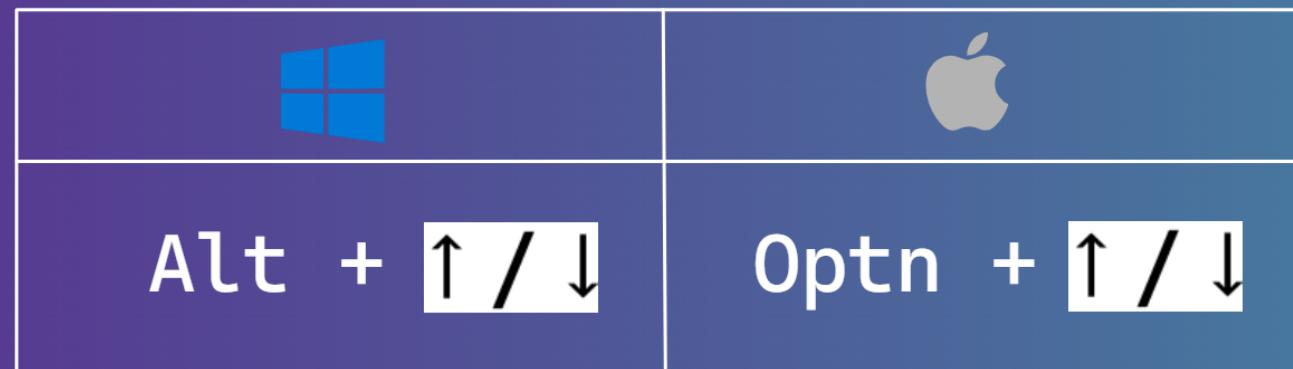
	
Alt + Click	Optn + Click

```
<div className={`add-cell ${force
  <div className="add-buttons">
    <button
      className="button is-rounde
      onClick={() => insertCellAf
```

```
<div className={`add-cell ${force
  <div className="add-buttons">
    <button
      className="button is-rounde
      onClick={() => insertCellAf
```

It will be helpful if you want to delete/replace multiple occurrences at once.

# Move line up/down



```
<AddCell forceVisible={cells.length === 0} prevCellId={null} />
{renderedCells}
```

Alt + ↓



```
{renderedCells}
<AddCell forceVisible={cells.length === 0} prevCellId={null} />
```

# Copy line up/down



Shift + Alt + ↑ / ↓

Shift + Optn + ↑ / ↓

```
<AddCell forceVisible={cells.length === 0} prevCellId={null} />
{renderedCells}
```

Shift + Alt + ↓



```
<AddCell forceVisible={cells.length === 0} prevCellId={null} />
<AddCell forceVisible={cells.length === 0} prevCellId={null} />
{renderedCells}
```

# Toggle line comment

	
Ctrl + /	Cmd + /

```
export default CellList;
```

Ctrl + /



Ctrl + /

```
// export default CellList;
```

# while dealing with HTML

```
<div className="action-bar">  
  .container.flex>.box-1+.box-2|
```

hit enter



```
<div className="action-bar">  
  <div className="container flex">  
    <div className="box-1"></div>  
    <div className="box-2"></div>  
  </div>  
</div>
```

But how 🤔 ?

Here, div will be created with "container" and "flex" className

If we don't provide any text before first dot, by default div will be created

```
div.container.flex>.box-1+.box-2
```

Here arrow will create child HTML element. As we don't have any HTML element mentioned before dot, hence div will be created as child

Here, plus creates its sibling HTML element

While there are still many shortcuts, there's no need to remember all of them. These limited shortcuts can significantly boost your productivity.



Cheers,  
Sahith Bodla

<https://www.linkedin.com/in/sahith-bodla-71701416b/>

follow for more