```
/**
* @type {Context}
*/
({
   /**
    * {@link
https://microsoft.github.io/monaco-editor/docs.html#interfaces/editor.I
StandaloneEditorConstructionOptions.html}
    * {@link
https://microsoft.github.io/monaco-editor/docs.html#interfaces/editor.I
StandaloneDiffEditorConstructionOptions.html}
    */
   options: {
       theme: "vs",
       language: "plaintext",
       wordWrap: "on",
       accessibilitySupport: "on",
       accessibilityPageSize: 20,
       scrollBeyondLastLine: true,
       ignoreTrimWhitespace: true,
       useInlineViewWhenSpaceIsLimited: true,
   },
   /**
    * {@link
https://microsoft.github.io/monaco-editor/docs.html#interfaces/editor.I
ActionDescriptor.html}
    */
   actions: {
       "user.toggleWordWarp": {
           label: "Toggle Word Wrap",
           keybindings: [
               "Alt+KeyZ",
           ],
           contextMenuGroupId: "view",
           contextMenuOrder: 4.5,
           run(editor) {
               let wrappingInfo =
editor.getOption(monaco.editor.EditorOption.wrappingInfo);
               editor.updateOptions({
                   wordWrapOverride2: wrappingInfo.wrappingColumn == -1
? 'on' : 'off',
               });
           },
       },
       "user.toggleIgnoreTrimWhitespace": {
           label: "Toggle Show Trim Whitespace Diff",
           precondition: "isInDiffEditor",
           contextMenuGroupId: "view",
           contextMenuOrder: 4.6,
           run() {
               let editor = editorUtil.editor;
               let value =
editor._options?.ignoreTrimWhitespace?.value;
               editor.updateOptions({
                   ignoreTrimWhitespace: !value,
               });
           },
       },
   },
   init: [
       {
           func() {
               // console.log("ext-code-editor.init");
           },
           args: [],
           injectImmediately: true,
       },
       {
           injectImmediately: true,
           func() {
               let addTouchTools = () => {
                   self.removeEventListener('touchstart',
addTouchTools, true);
                   console.debug('addTouchTools');
                   editorUtil.addActions({
                       "user.undo": {
                           label: "Undo",
                           precondition: "textInputFocus &&
!editorReadonly",
                           contextMenuGroupId: " 1_modification",
                           contextMenuOrder: 11,
                           run(editor) {
                               editor.trigger("", "undo");
                           },
                       },
                       "user.redo": {
                           label: "Redo",
                           precondition: "textInputFocus &&
!editorReadonly",
                           contextMenuGroupId: " 1_modification",
                           contextMenuOrder: 12,
                           run(editor) {
                               editor.trigger("", "redo");
                           },
                       },
                       "user.paste": {
                           label: "Paste Clipboard Text",
                           precondition: "textInputFocus &&
!editorReadonly",
                           contextMenuGroupId: "9_cutcopypaste",
                           contextMenuOrder: 4,
                           async run(editor) {
                               // Requires ACCESS_LEVEL >=30
                               if (self.browser) {
                                   try {
                                       await
browser.permissions.request({ permissions: ['clipboardRead'] });
                                   } catch (e) {
                                       console.warn(e);
                                   }
                               }
                               try {
                                   let text = await
navigator.clipboard.readText();
                                   if (!text) return;
                                   editor.executeEdits("", [{
                                       forceMoveMarkers: true,
                                       range: editor.getSelection(),
                                       text,
                                   }]);
                               } catch (e) {
                                   console.warn(e);
                               } finally {
                                   // await
browser.permissions.remove({ permissions: ['clipboardRead'] });
                               }
                           },
                       },
                       "user.delete": {
                           label: "Delete",
                           precondition: "!editorReadonly &&
editorHasSelection",
                           contextMenuGroupId: "9_cutcopypaste",
                           contextMenuOrder: 5,
                           run(editor) {
                               editor.executeEdits("", [{
                                   forceMoveMarkers: true,
                                   range: editor.getSelection(),
                                   text: "",
                               }]);
                           },
                       },
                       "user.selectAll": {
                           label: "Select All",
                           precondition: "textInputFocus",
                           contextMenuGroupId: "9_cutcopypaste",
                           contextMenuOrder: 6,
                           run(editor) {

editor.setSelection(editor.getModel().getFullModelRange());
                           },
                       },
                       "user.find": {
                           label: "Find...",
                           precondition: "textInputFocus &&
!findWidgetVisible",
                           contextMenuGroupId: "view",
                           contextMenuOrder: 4.9,
                           run() {
                               editor.trigger("", "actions.find");
                           },
                       },
                   });
               };
               self.addEventListener('touchstart', addTouchTools,
true);
           },
       },
   ],
})
```