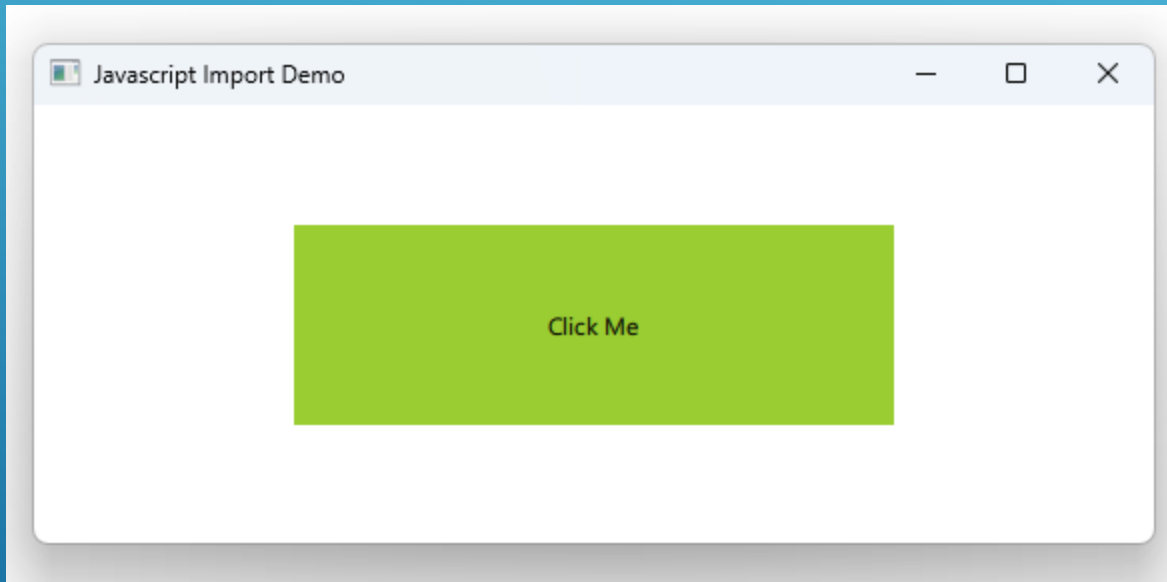


Notes to self

- . `Qt.include` also makes content from `utilities2.js` available in `qml` files importing `utilities1.js`
- . For example, in `main.qml` we can access the `add` function, even if it was just imported in `utilities1.js`.
- . `Qt.include` pollutes the scope of `utilities1.js` and is not advised in projects.
- . It's instead recommended to use `js` modules where you have greater control on what is exported from your `js` files. We see that in the next lecture.

JS Indirect Import



utilities2.js

```
function add(a,b) {  
    console.log("Method from utilities2.js called")  
    return a + b  
}
```

utilities1.js

```
Qt.include("utilities2.js")

function greeting()
{
    console.log("Hello there from external JS file : utilities1.js")
}

function combineAges( age1, age2)
{
    return add(age1,age2)
}
```

main.qml

```
import "utilities1.js" as Utilities1

Window {
    visible: true
    width: 640
    height: 480

    Rectangle {
        color: "yellowgreen"

        MouseArea {
            anchors.fill: parent
            onClicked: {
                //console.log("Our ages combined yield :" + Utilities1.combineAges(33,17))
                console.log("Our ages combined yield :" + Utilities1.add(33,17))
            }
        }
    }
}
```