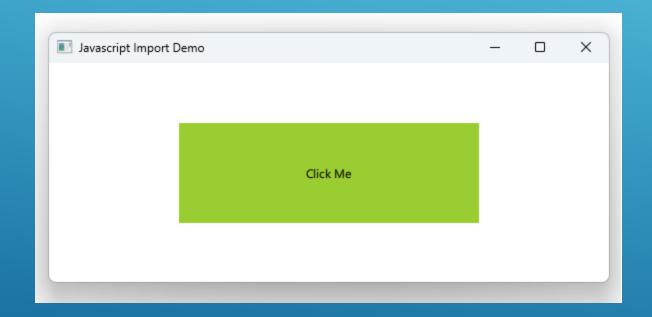# Notes to self

```
. Exploring JS modules
 . doc : https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules
 . The doc is targetted to js as used in browsers but the syntax applies to what
    we'll be using in QML here.

 . In js modules, anything you want to be visible in external files needs to be
    explicily exported. This applies to function names, variables, ...

 . Modules give you the flexibility of controlling what's exported in other files.
 . In qml they are stored in files with a .mjs extension
 . Support in Qt Creator isn't great yet:
    . There isn't a direct option to create .mjs files
    . The syntax for exporting and importing still throws squiggle error lines
    . But they work fine as we're about to see

.Content :
    . Here we see how to import everything (*) from a mjs file.
    . Exporting the add function from utilities2.js for use in main.qml
    . The subtract function from utilities2.js isn't exported. Can't use
       it in main.qml
    . Just because add is imported in utilities1.js doesn't mean you
        can use it directly in main.qml that imports utilities1.js
    . There are ways to make content in utilities2.js also be visible
       by re-exporting in utilities1.js but the syntax doesn't work
       when you've imported everything with * like we've done here

    . One can browse the docs to learn more
```

JS Modules

utilities1.js

```javascript
export function add(age1,age2){
        return age1 + age2
}

 function subtract(age1,age2){
    return age1 - age2;
}
```

utilities2.js

```javascript
import * as Utilities2 from "utilities2.mjs";

export function combineAges(age1,age2){
    return Utilities2.add(age1,age2)
}

export function ageDiff(age1,age2){
    return Utilities2.subtract(age1,age2)
}
```

main.qml

```qml
import "utilities1.mjs" as Utilities1

MouseArea {
    anchors.fill: parent
    onClicked: {
      /*
      //Calling a properly exported mathod
       console.log("Our ages combined yield :" + Utilities1.combineAges(33,17))
        */
        /*
        //Can't call a method that's not exported : subtract isn't exported from utilities2.mjs
        console.log("Age diff : " + Utilities1.ageDiff(33,17))
        */

        //Just because add is usable from utilities1.js doesn't mean main.qml
         // can use it. utilities2.js doesn't export it. You can export explicit names
        //as we'll see in a later lecture.
         console.log ("The sum is : " + Utilities1.add(33,17))//Error
    }
}
```