Modules in Java Script

- -> Highly self contained and have their own declicated functionality
- I we generally have multiple Is files for our whole project
- -by that we want and export the same

Before diving in let's see what ways we already know?

- 1) We can have functions (they can have their own variables), but for shared data we will need global variables (which is bad right?
- 2) We can have multiple script files (if they have global variables with the same name, it's going to override it)

@codeWithSimman

Modules give us a better way to organise variables and functions so when we group these together It makes sense

Where do modules come?

Global supe

@codeWithSimvan

I module scope

Function scope

> Block Scope

Ways to achieve module scope

Before

1) IIFE (Immediatly invoked function expression)

A Let's say want to wrap your entire Is content inside an IIFE

var module 1 = (function () {

var name = 'Simran';

return {

want other files

name: name to have access, you

can return and keep the rest private to the 11FE

3)();

This is called the revealing module pattern

you only reveal (return) what you want from the module

The other file that now wants to use variable name can simply do modelel. name acode with Simman

Note: after wrapping your code in an IIFE it's no longer a part of the global scope and we're not polluting the global scope.

But what about the variable modules? Well it's still a global variable

- * So we can still overnide module 1 in another file (problem 1)
- * We need to keep track of all the dependencies.

For example

@codewithSimran

<script &rc='./script1.js>

<script &rc='./script2.js>

<script &rc='./script3.js>

Now if script1 and script3 use a bunchion say Name that is defined inside script2, script1 wonthaw access to it (because script 2 is included after script 1) and we'll get an error.

Summary

a problems

@codeWithSimvan

1) Still a global variable (module variable) 2) need to maintain the dependencies

Scanned with CamScanner