

≡ planning budget project ≡

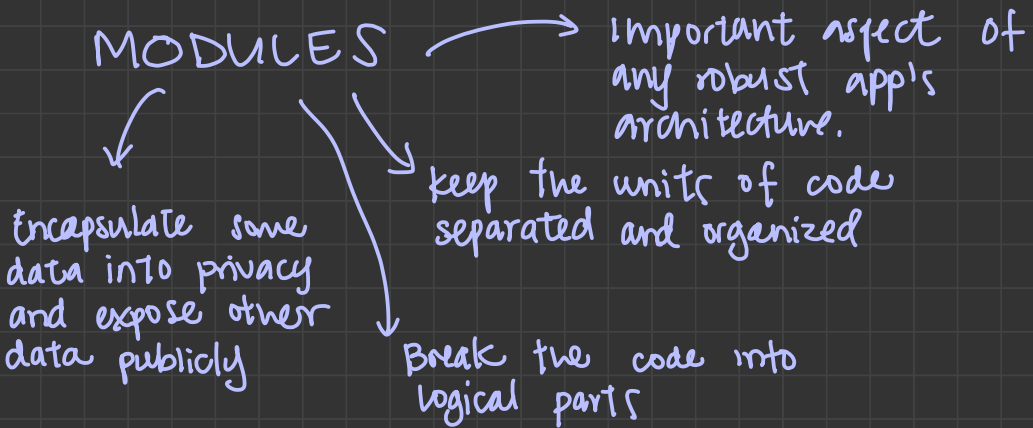
1. Identify the most important parts of the web app

TO-DO LIST

- add event handler
- get input values
- add the new item to the data structure
- add the new item to the UI
- calculate budget
- update the UI.

2. How to structure the code.

MODULES



eg. →

UI module

DATA module

controller module

module pattern in order to get related pieces of code together
inside of independent and organized units

```
graph TD; A[module pattern] --> B[public]; A --> C[private]; C --> D[data encapsulation];
```

public private → data encapsulation

Remember a IIFY allows data privacy because it creates a new scope that is not visible from the outside scope.

The secret is to return an object with the things we want to be public

Separation of concerns → each controller should be interested in do one thing independently

```
var budgetController = (function() {  
  
    var x = 23;  
  
    var add = function(a){  
        return x + a;  
    }  
  
    return {  
        publicTest: function(b){  
            console.log(add(b));  
        }  
    }  
  
})();
```

```
>> budgetController.x  
← undefined  
>> budgetController.add  
← undefined  
>> budgetController.add(5)  
! ▶ Uncaught TypeError: budgetController.add is debugger eval code:1:18  
    not a function  
      <anonymous> debugger eval code:1  
    [Saber más]  
>> budgetController.publicTest(5)  
28 app.js:11:15  
← undefined  
>>
```

sólo podemos acceder mediante este método
a los datos, the one exposed. CLOSURES

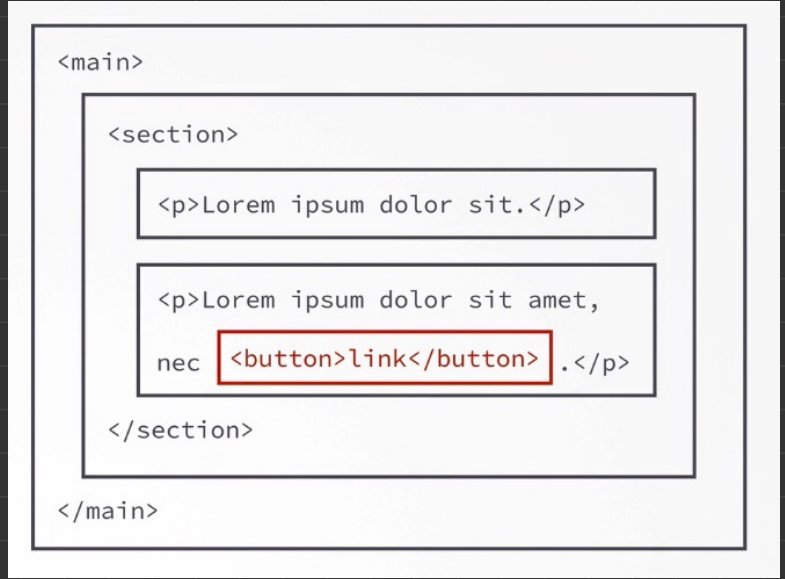
Event delegation

Event bubbling



when we trigger an event in an element, the same event happens in the elements parents. One at a time. All the way up.

The target element is the one that initially started the event.



Event delegation is to wait until the action bubbles up and parents elements so we can add the event handler to a parent.

Event Delegation is to set the event not on the element that triggers it but in the parent.

Use cases:

1. when we have an element with lots of child elements. Instead of adding an event handler to all of them we add the event handler to the parent and determine through 'target' in which element the event was triggered.

2. when we want an event handler attached to an element that is not yet in the DOM when our page is loaded

DOM Traversing → move up from one element to its parents

NodeList

use the slice method to convert a node list into an array