



Routing

So router....

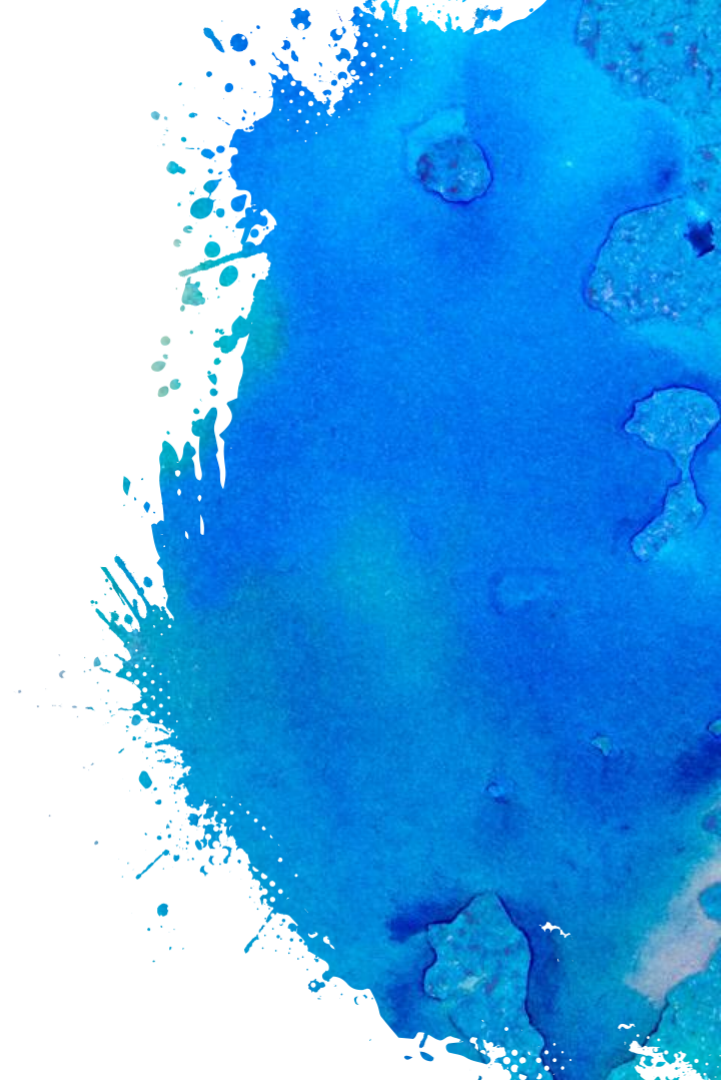
Comes handy being able to modify the URL in SPAs, for client side ... and also to us devS !

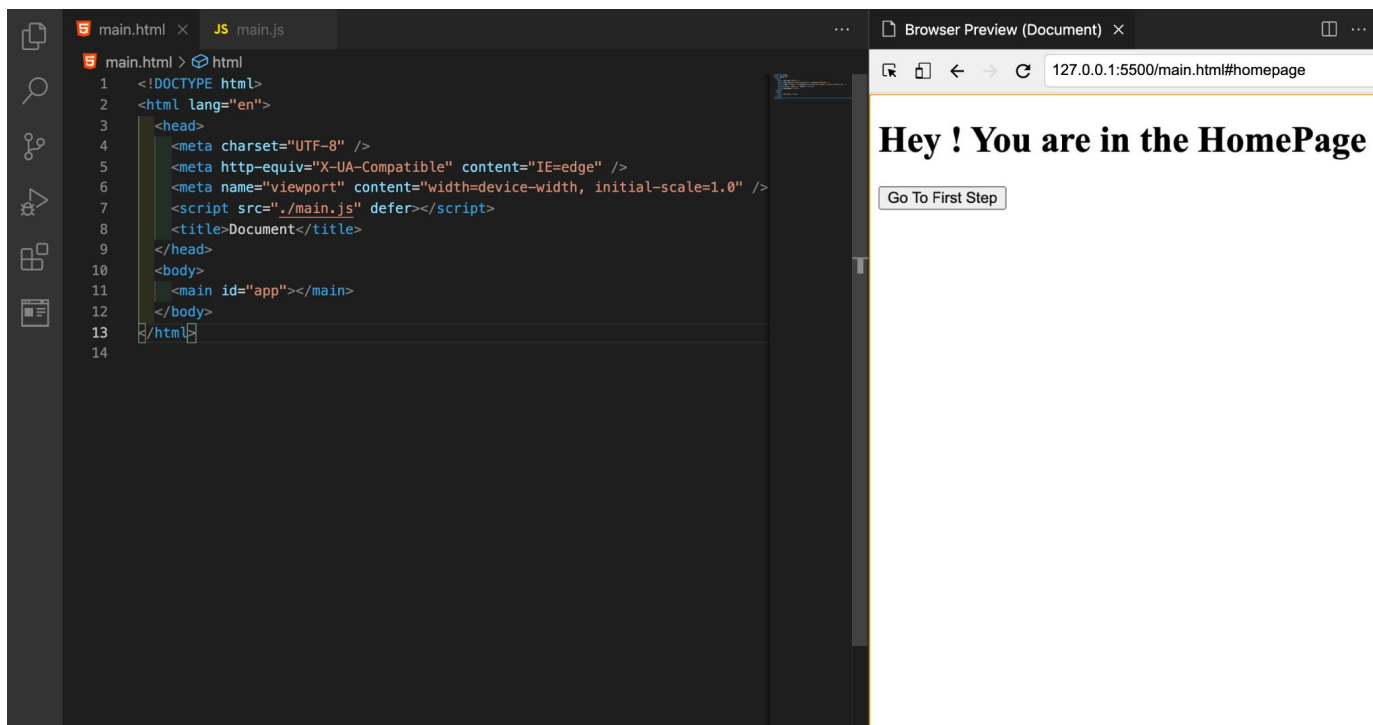
It's always a good idea that every element of our project goes in the same line, doesn't it?



Let's see it in action

Peer Coding , let's go !
Don't worry about documentation, I got you covered in this slides





Let's use the famous **<template>** tags ! That are now like family to you, I hope :)

The image shows a code editor on the left and a browser preview on the right. The code editor displays a JavaScript file named `main.js` with a `navigate` function and two callback functions, `goToHomepage` and `goToFirstStep`. A green box highlights the logic in the `navigate` function that checks the `location.hash` and calls the appropriate callback. The browser preview on the right shows the rendered HTML, which includes a heading "Hey ! You are in the HomePage" and a button labeled "Go To First Step". The browser's address bar shows the URL `127.0.0.1:5500/main.html#homepage`.

```
JS main.js > navigate
1
2 let templateApp = `<template id="tiny-example">
3   <h1 id="title"></h1><button id="goTo">Click Me</button></template>`;
4 let mainTag = document.getElementById("app");
5 let buttonApp;
6
7 navigate();
8
9 function navigate() {
10
11   /* Cant use switch here beacuse requires constant values,
12    * and JS does not support it
13   */
14   if (location.hash === "" || location.hash === "#" ||
15       location.hash === "#homepage") {
16     goToHomepage();
17   }
18   else if (location.hash === "#first-step") {
19     goToFirstStep();
20   }
21 }
22
23
24 function goToHomepage() {
25   location.hash = "homepage";
26   homepageDisplays();
27 }
28
29 function goToFirstStep() {
30   location.hash = "first-step";
31   firstStep();
32 }
33 }
```

Browser Preview (Document) ×

127.0.0.1:5500/main.html#homepage

Hey ! You are in the HomePage

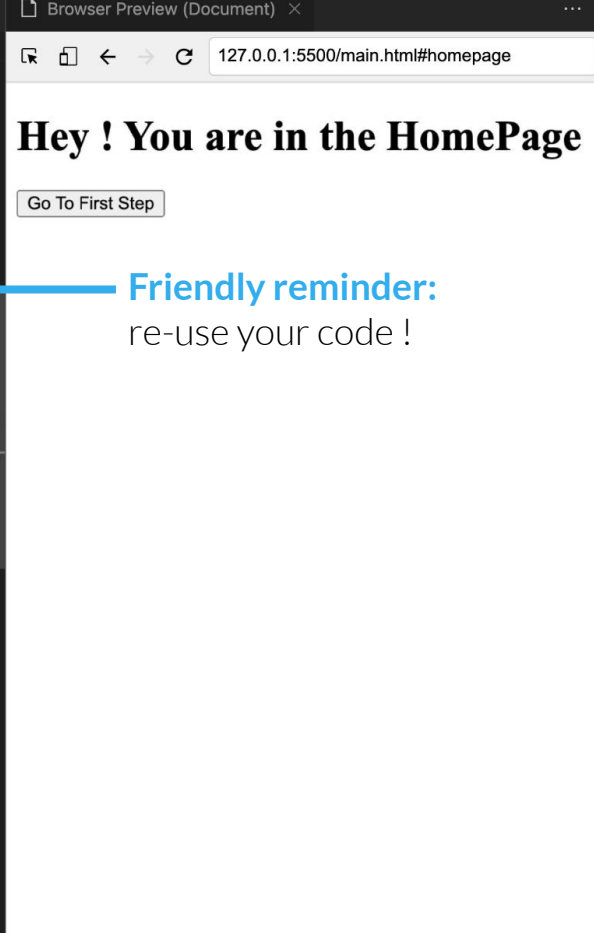
Go To First Step

Basically you will follow the same pattern:

1. an event fires up modifying the *hashtag* in the URL
2. that hashtag will send you by the *navigate()* function wherever you want to go

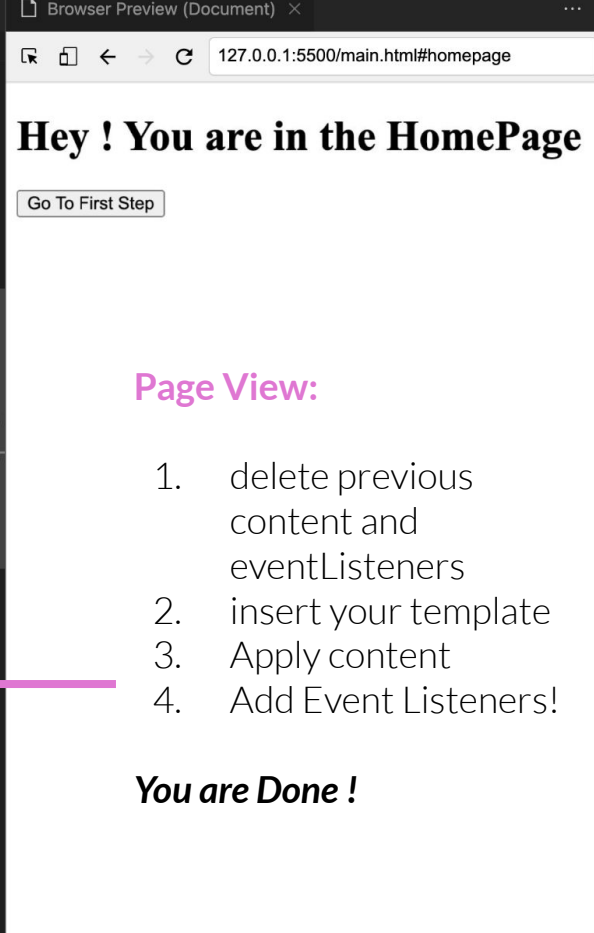
Decide which **hashtags** you want to use and define the functions that the **navigate()** will use as callback

```
main.html JS main.js
JS main.js > homepageDisplays
33
34 // We will use it always, Better to define it !
35 function insertTemplate() {
36     mainTag.insertAdjacentHTML("afterbegin", templateApp);
37
38     let templateContent = document.getElementById("tiny-example").content;
39     let cloneNode = templateContent.cloneNode(true);
40
41     mainTag.lastChild.remove();
42     mainTag.append(cloneNode);
43 }
44
45
46
47
48 function homepageDisplays() {
49     //remove every peace of content and eventListeners before adding
50     if (mainTag.content) {
51         buttonApp.removeEventListener("click", goToHomepage)
52         mainTag.childNodes.forEach(element => {
53             element.remove()
54         });
55     }
56
57     insertTemplate();
58     document.getElementById("title").textContent =
59     "Hey ! You are in the HomePage"
60     buttonApp = document.querySelector("#goTo");
61     buttonApp.textContent = "Go To First Step";
62     buttonApp.addEventListener("click", goToFirstStep)
63 }
64
65
66
67
68
69
```



Now with only 2 views one could not mind, but imagine having hundreds ! **Always think big !**

```
main.html JS main.js
JS main.js > homepageDisplays
33
34 // We will use it always, Better to define it !
35 function insertTemplate() {
36
37     mainTag.insertAdjacentHTML("afterbegin", templateApp);
38
39     let templateContent = document.getElementById("tiny-example").content;
40     let cloneNode = templateContent.cloneNode(true);
41
42     mainTag.lastChild.remove();
43     mainTag.append(cloneNode);
44
45 }
46
47
48 function homepageDisplays() {
49     //remove every peace of content and eventListeners before adding
50     if (mainTag.content) {
51         buttonApp.removeEventListener("click", goToHomepage)
52         mainTag.childNodes.forEach(element => {
53             element.remove()
54         });
55     }
56
57     insertTemplate();
58     document.getElementById("title").textContent =
59     "Hey ! You are in the HomePage"
60     buttonApp = document.querySelector("#goTo");
61     buttonApp.textContent = "Go To First Step";
62     buttonApp.addEventListener("click", goToFirstStep)
63 }
64
65
66
67
68
69
```



Page View:

1. delete previous content and eventListeners
2. insert your template
3. Apply content
4. Add Event Listeners!

You are Done !

Decide which **hashtags** you want to use and define the functions that the **navigate()** will use as callback

```
JS main.js > homepageDisplays
9
10 function firstStep() {
11
12   if (mainTag.content) {
13     buttonApp.removeEventListener("click", goToFirstStep)
14     mainTag.childNodes.forEach(element => {
15       element.remove()
16     });
17   }
18   insertTemplate();
19   document.getElementById("title").textContent =
20     "Hey ! You are in the First Step"
21   buttonApp = document.querySelector("#goTo");
22   buttonApp.textContent = "Go To HomePage";
23   buttonApp.addEventListener("click", goToHomepage)
24 }
25
```

127.0.0.1:5500/main.html#first-step

Hey ! You are in the First Step

Go To HomePage

Ups.... same story here

Can you imagine a way to
standardize this?

Using Objects in this subject could come in **handy ...** but it doesn't matter in this case study :)

Our STAR is:

- hash -

We will use the string that comes after the # (hashtag) to elaborate the different routes or directions to be followed.

location.hash;

The hash property of the Location interface returns a USVString containing a '#' followed by the fragment identifier of the URL — the ID on the page that the URL is trying to target.



location.hash===“x”

So... our code basically “routes” or send you to the script that its needed depending of the hash value.

Let's say x is “homepage”, so the router will be setted to get the part of the code where the view for the homepage it's being displayed. (hopefully)



Useful Links

Some docu

<https://developer.mozilla.org/en-US/docs/Web/API/USVString>

<https://developer.mozilla.org/en-US/docs/Web/API/Location/hash>

Another idea

<https://medium.com/@brvanmanuele/how-i-implemented-my-own-spa-routing-system-in-vanilla-js-49942e3c4573>

Git-hub

The code for this example is in here, don't go nuts:

<https://github.com/Paola3stefania/templates-router-lesson>

