Web Programming
SPA

Leander Jehl | University of Stavanger

Single Page Application (SPA)

- After initial page load, we can use only AJAX.

- Build page using JS.

time

Load page

HTML & JS

(complete document)

AJAX

click

JSON

JS: update DOM

AJAX

click

JSON

JS: update DOM

web server

after initial page load, only Ajax
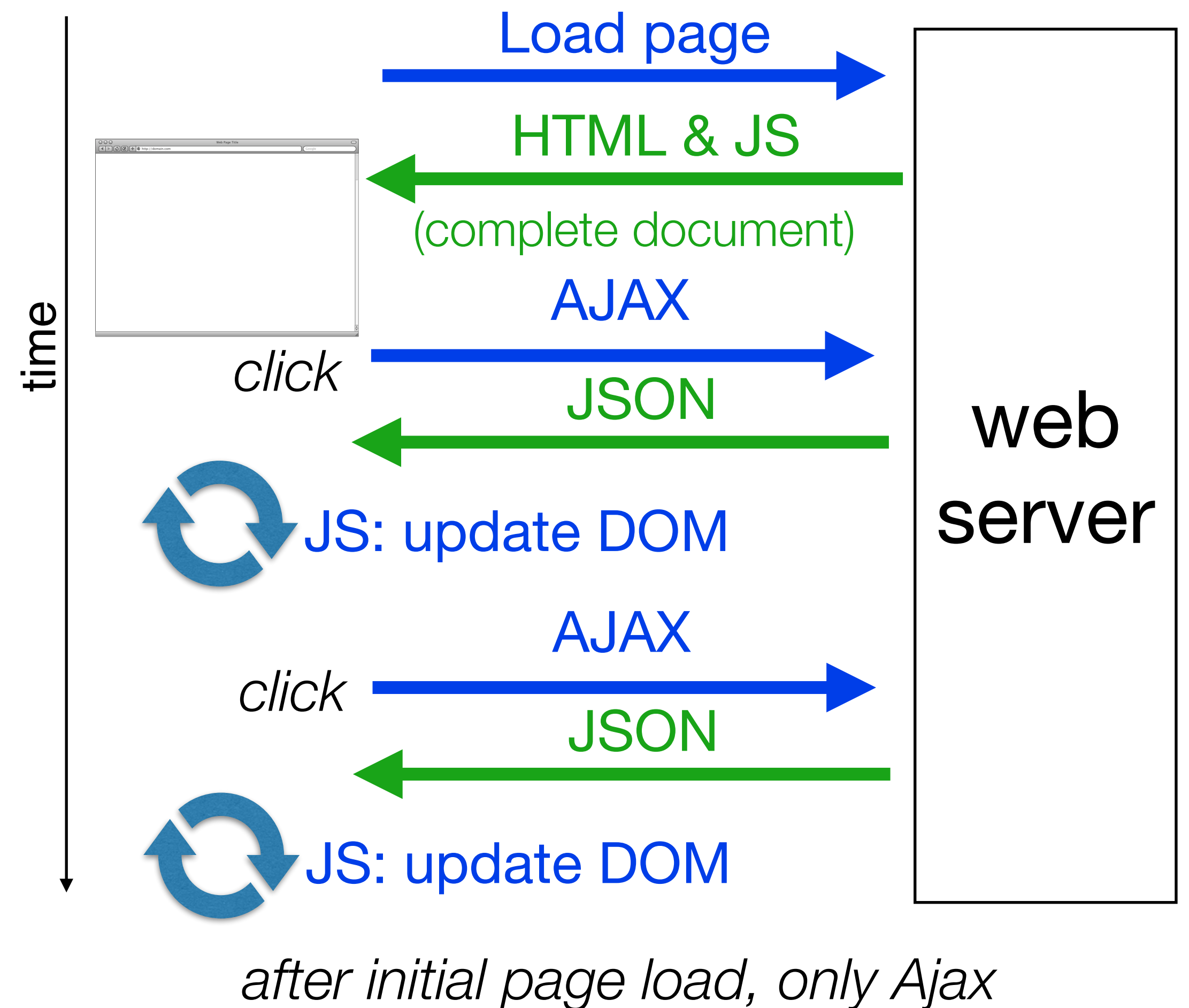
SPA pros and cons

PRO

- Multiple concurrent interaction

- No page refresh (keep state)

- Fast interaction
  - load only data

CONS

- Complex JS code

- Slow initial page load

  - load javascript code

  - update the dom

  - render new dom

Example

https://github.com/dat310-2025/info/tree/master/
examples/spa/mpa

Example
Multi page application (MPA)

State in JS is lost

Example

https://github.com/dat310-2025/info/tree/master/
examples/spa/norouter

Example SPA

Route does not change
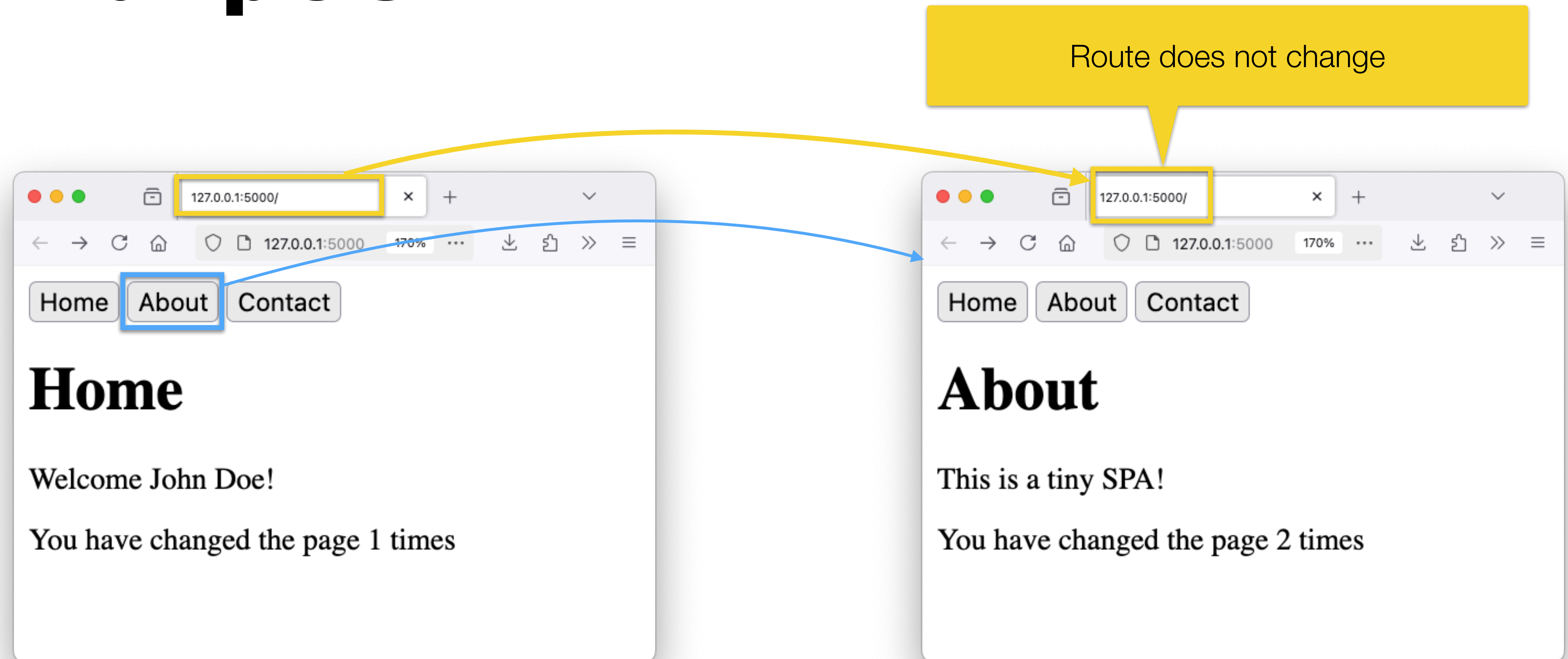
Routing in JS

Hashrouting

Use #… part of URI for part of route handled in JS

- Links that only change the hash:

```
<a href="/#/">Home</a>
<a href="/#/about">About</a>
<a href="/#/contact">Contact</a>
```
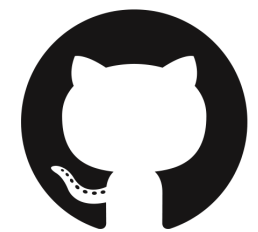
- React to change in hash in JS

```
window.onhashchange = function(){ ... };
```
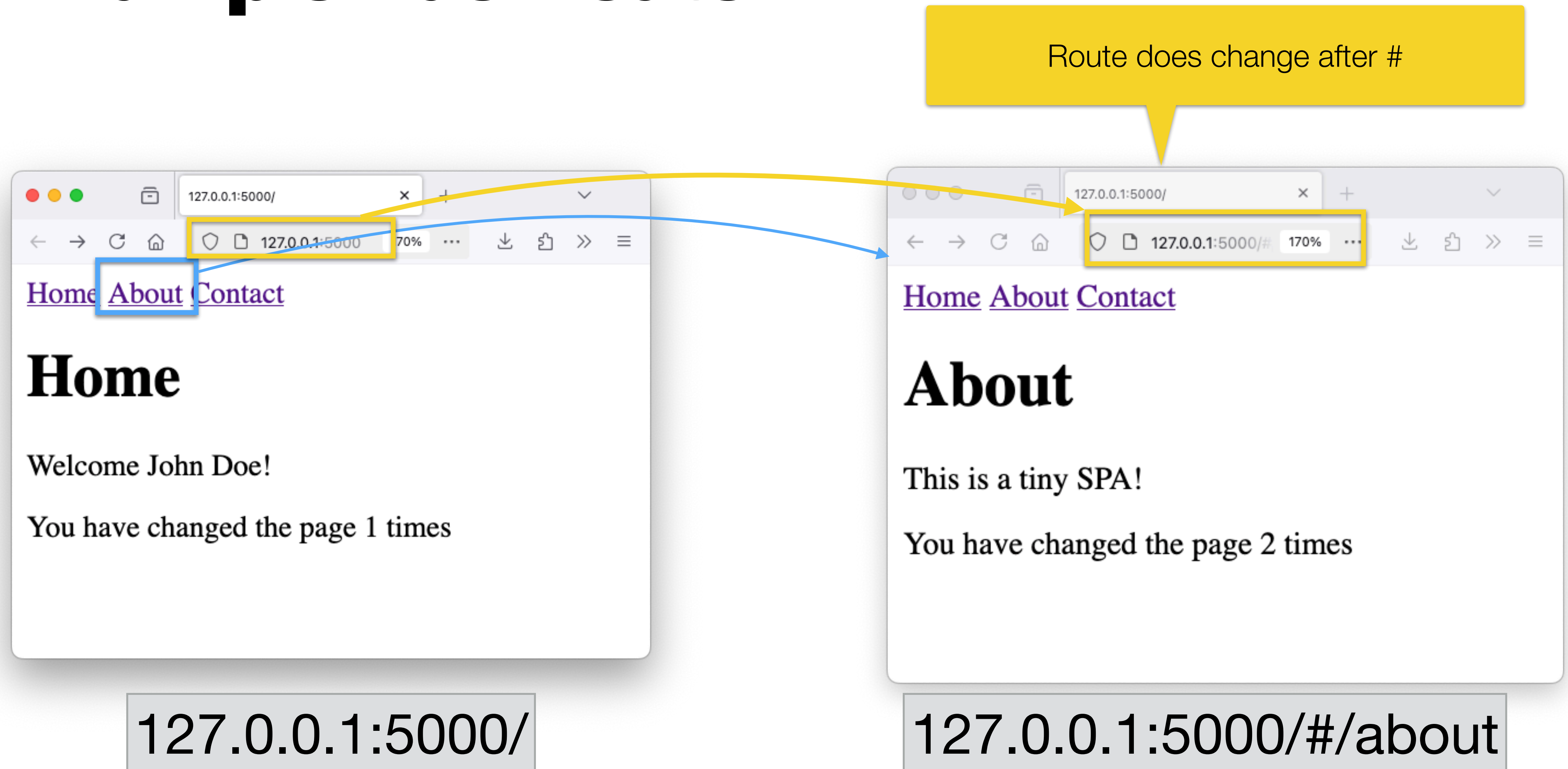
- Read hash part in JS

```
render(window.location.hash);
```

Example

https://github.com/dat310-2025/info/tree/master/
examples/spa/hashrouter

Example Hashrouter

Route does change after #

Home About Contact

Home

Welcome John Doe!

You have changed the page 1 times

127.0.0.1:5000/

Home About Contact

About

This is a tiny SPA!

You have changed the page 2 times

127.0.0.1:5000/#/about

History API

History API allows to add to browser history

- Add new entry to history, updating URL, without reload

```javascript
history.pushState({},"", path);
```

- Creat flask route that ignores part handled in JS

```python
@app.route('/', defaults={'path': ''})
@app.route('/<path:path>')
def catch_all(path):
    return app.send_static_file("index.html")
```

- Read hash part in JS

```javascript
render(window.location.pathname);
```

History API

History API allows to add to browser history

- Go back in history

```
history.go(-1);
```

- Listen for event going back

```
window.addEventListener("popstate", function(event){})
```
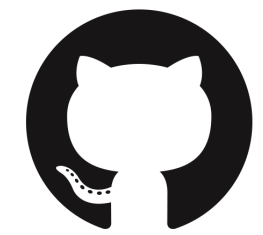
- Avoid reloading

```
event.preventDefault();
```

Example

https://github.com/dat310-2025/info/tree/master/
examples/spa/historyrouter

Exercise #1 & 2

github.com/dat310-2025/info/tree/master/
exercises/spa

Resources

- Stack overflow answer: https://stackoverflow.com/a/67041356

Local storage

Local storage

Allows pages to store data long time in the browser

- Store data as key value pair:

```
window.localStorage.setItem("key", "valuestring")
```

- Get value by key:

```
window.localStorage.getItem("key")
```

  - Available on pages with the same top level domain

- Can store more complex data using JSON

```
window.localStorage.setItem("key", JSON.stringify(value))
```