JÖNKÖPING UNIVERSITY
School of Engineering
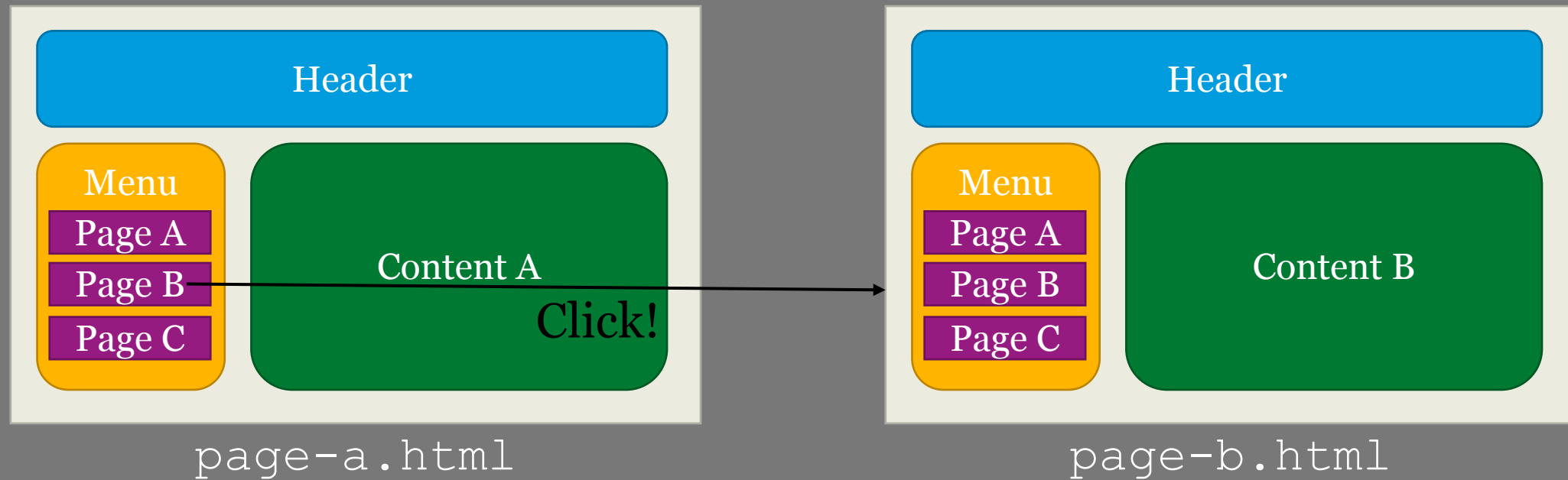
AJAX

Peter Larsson-Green

Jönköping University

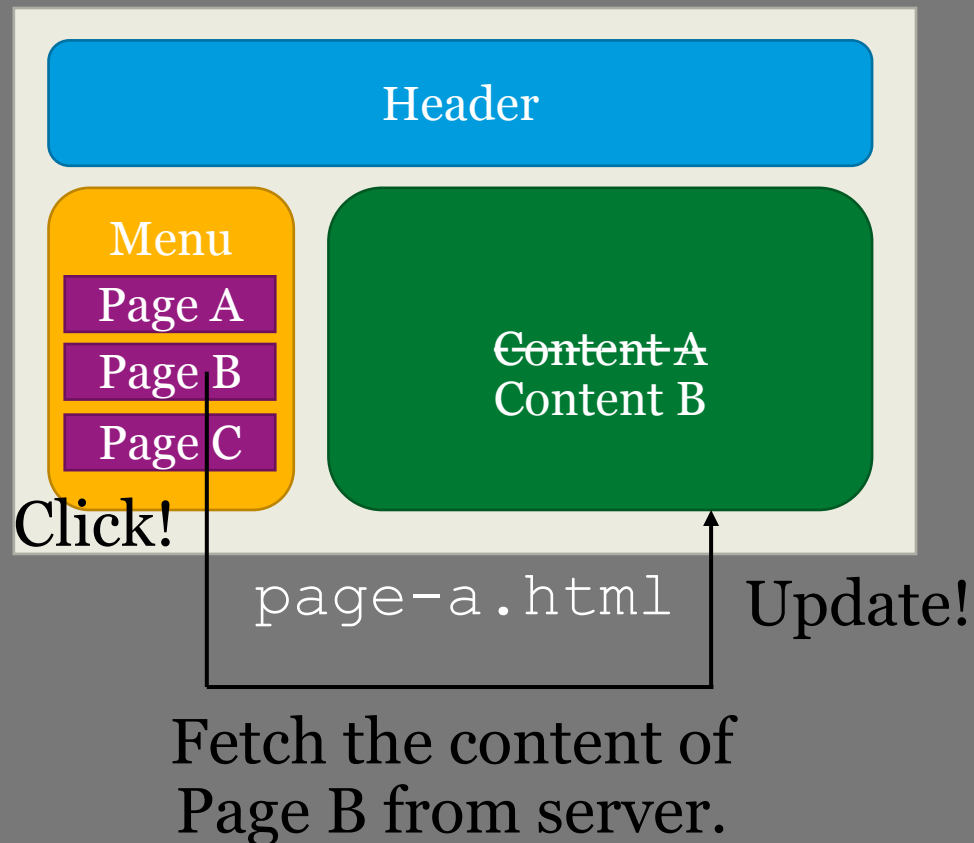Spring 2019

TRADITIONAL WEBSITES

page-a.html

page-b.html

We fetch information we already have.

- Takes extra time ☹
- Wasting data ☹
- Loading... ☹

"MODERN" WEBSITES

Header

Menu
- Page A
- Page B
- Page C

~~Content A~~
Content B

Click!

`page-a.html`   Update!

Fetch the content of
Page B from server.

Single Page Application:

- A single HTML file, and a lot of client-side JavaScript code.

AJAX

Asynchronous JavaScript and XML.

In old web browsers: `XMLHttpRequest`

In modern web browsers: `fetch`

THE `fetch` FUNCTION

```javascript
fetch('http://www.the-website.com/api/humans', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({name: "Alice"})
}).then(function(response){
  // Got back an HTTP response ☺
}).catch(function(error){
  // Something went wrong ☹
})
```

```
POST /api/humans HTTP/1.1
Host: www.the-website.com
Content-Type: application/json
Content-Length: 17

{"name": "Alice"}
```
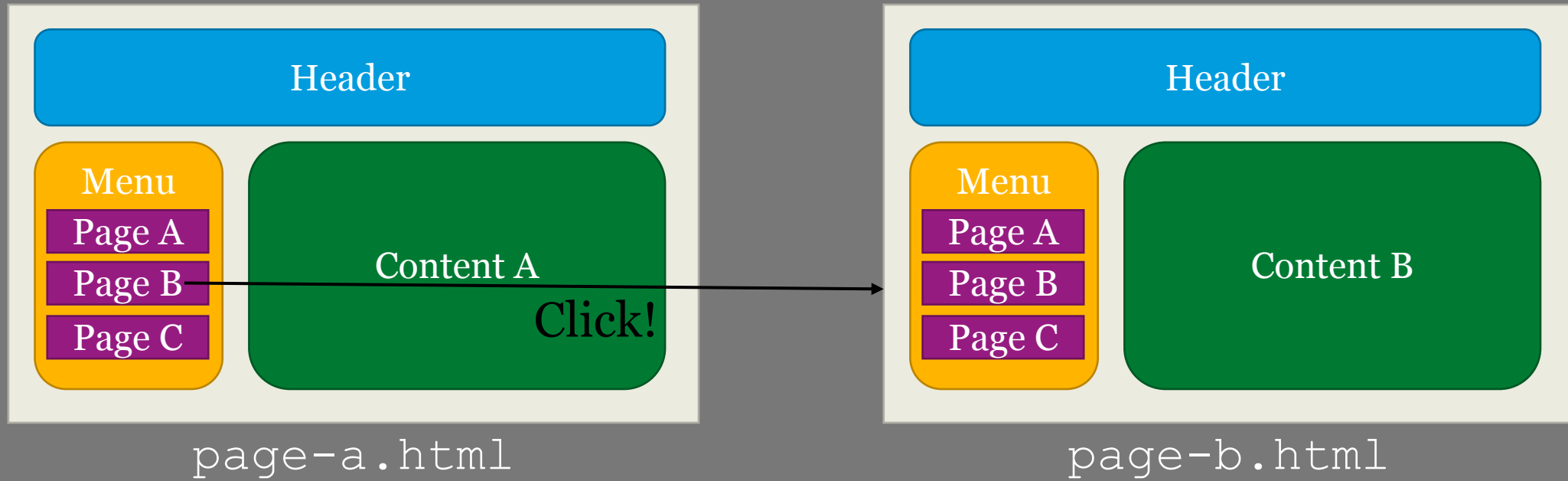
THE `fetch` FUNCTION

```javascript
fetch('http://www.the-website.com/api/humans/4', {
  // ...
}).then(function(response){
  const statusCode = response.status
  const contentType = response.headers.get("Content-Type")
  return response.json()
.then(function(human){
  const name = human.name
}).catch(function(error){
  // ...
})
```

```
HTTP/1.1 200 OK
Content-Length: 15
Content-Type: application/json

{"name": "Bob"}
```

MANAGING THE HISTORY

MANAGING THE HISTORY

Add history entry:

```
history.pushState({page: "about"}, "Title", "/about")
```

Listen for history changes:

```
window.addEventListener('popstate', function(event){
  const state = event.state
})
```

MANAGING THE HISTORY

Example