ACADEMIND

Variables & Constants

Variables

"True" variables where the stored value actually changes

```
let name = 'Anna';
name = 'Max';
```

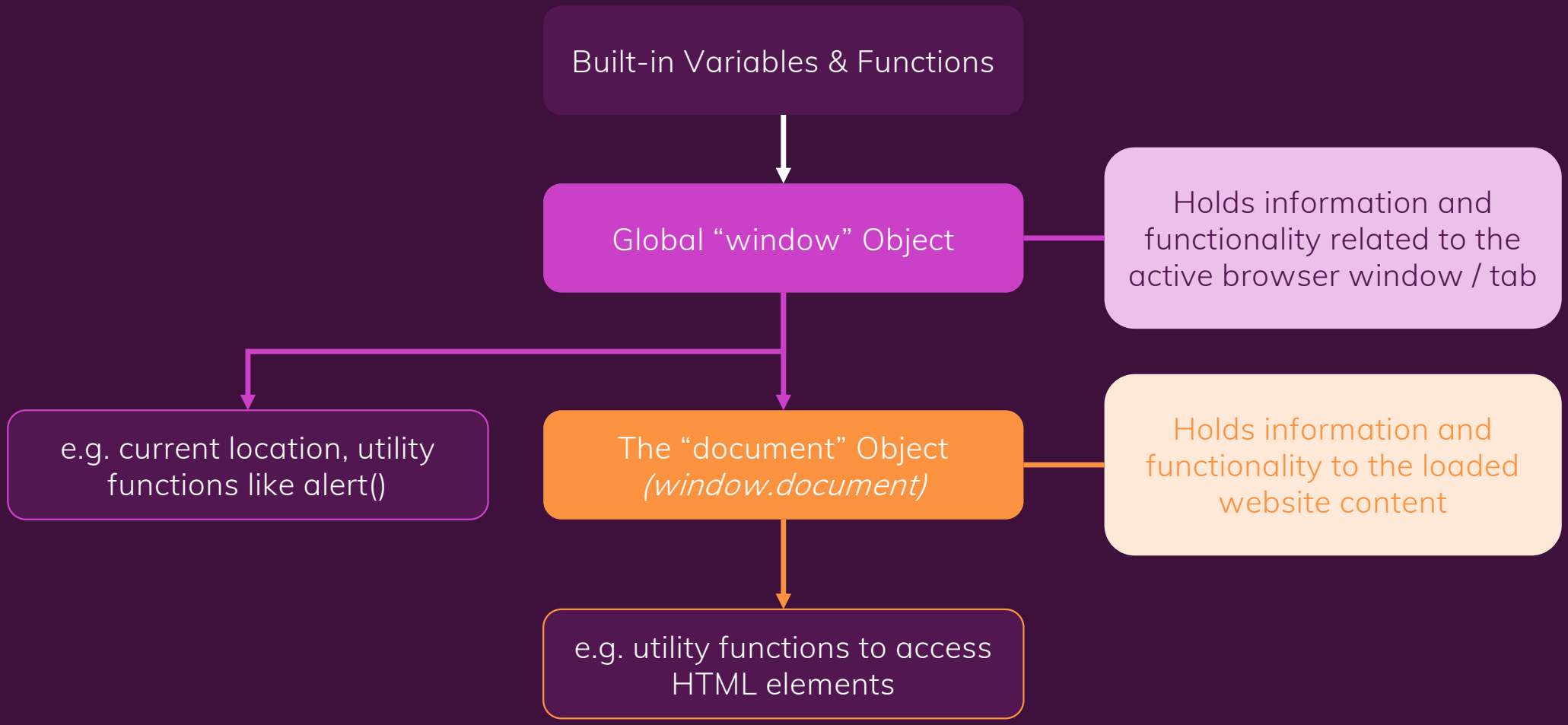
Constants

"Variables" where the value actually never changes

```
const enteredValue = 'Hi';
```

ACADEMIND

“window” & “document”



ACADEMIND

"The DOM"

Document **O**bject **M**odel

The data representation ("internal representation") of the parsed HTML code

The browser parses our HTML code and saves all elements as JavaScript objects

→ Our JavaScript code is able to interact with the DOM and extract data from it or manipulate it

ACADEMIND

From HTML To DOM

Our HTML Code
(sent from server to browser)

```
<!DOCTYPE html>
<html lang="en">
  <head>
    …
  </head>
  <body>
    <h1>Hi there!</h1>
    <p>
      This is a
      <a href="…">link</a>
    </p>
  </body>
</html>
```

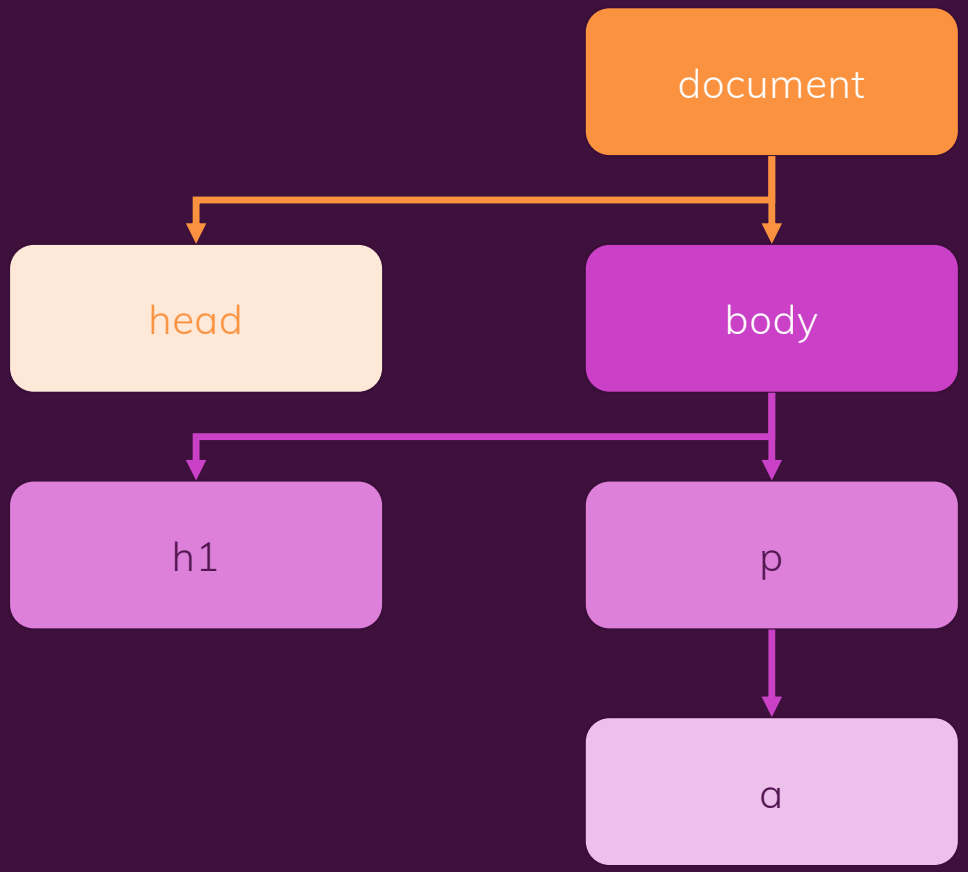
The DOM ("document" object)
(translated data representation)

```
{
  …,
  body: {
    …,
    children: [
      …,
      {
        nodeName: 'P',
        children: [
          { nodeName: 'A' }
        ]
      },
    ]
  }
```

This is vastly simplified – the "document" object contains way more data!

ACADEMIND

The DOM Tree



ACADEMIND

Selecting DOM Elements

“Drill Into Elements”

```
document.body.children[0].firstChild
```

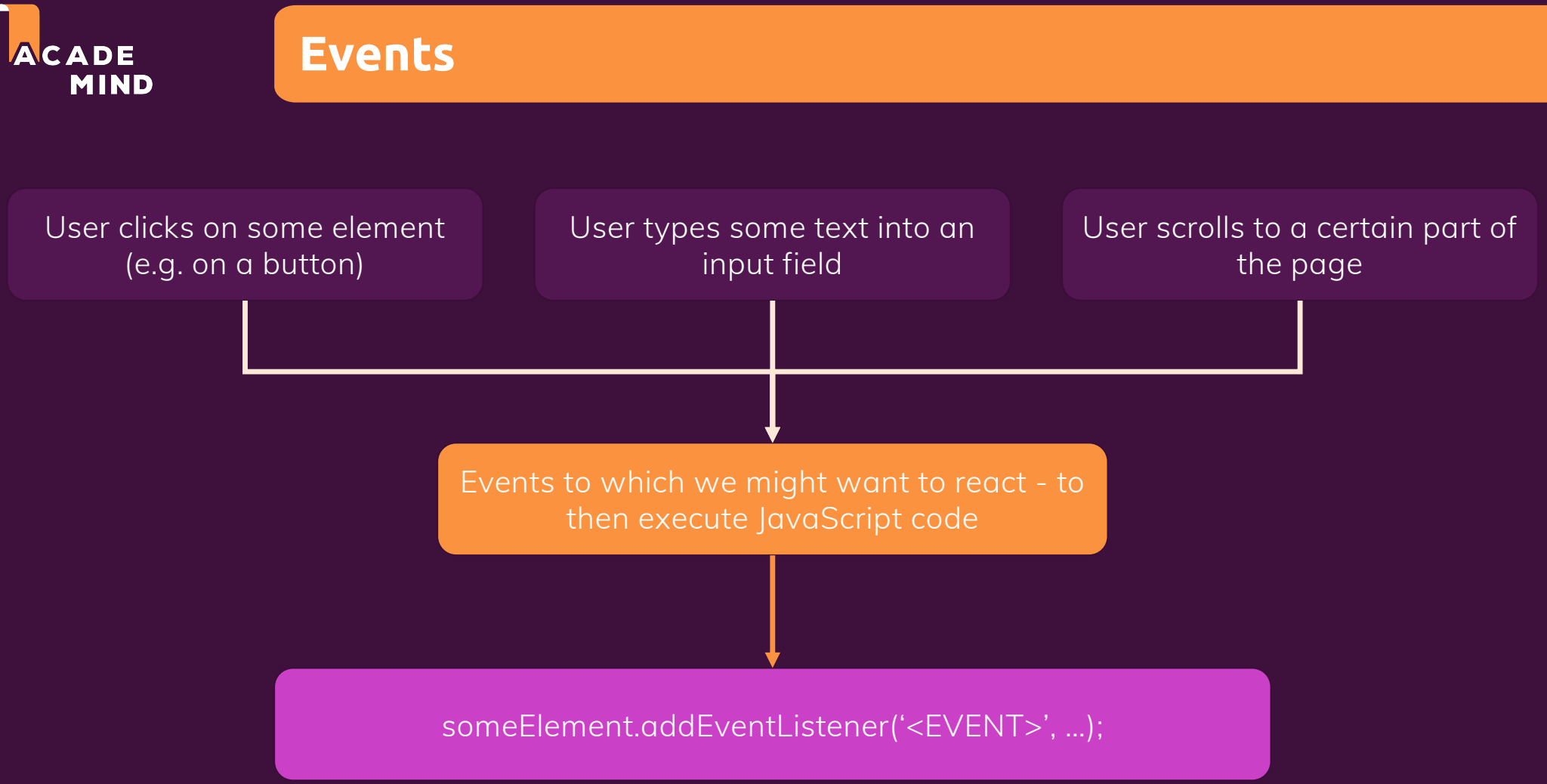
You have to know the DOM structure and if it changes, your code needs to change as well

Query Elements

```
document.getElementById('some-id');
document.querySelector('.some-class');
```

Selecting elements works like in CSS and hence no exact DOM structure knowledge is required

Events


User clicks on some element (e.g. on a button)
User types some text into an input field
User scrolls to a certain part of the page
Events to which we might want to react - to then execute JavaScript code
someElement.addEventListener('<EVENT>', ...);