Toegepaste Informatica

2TX

2022-2023

Webontwikkeling 3

JS: DOM

J. Pieck, G. Jongen, A. Vranken, J. Van Hee

JS: DOM

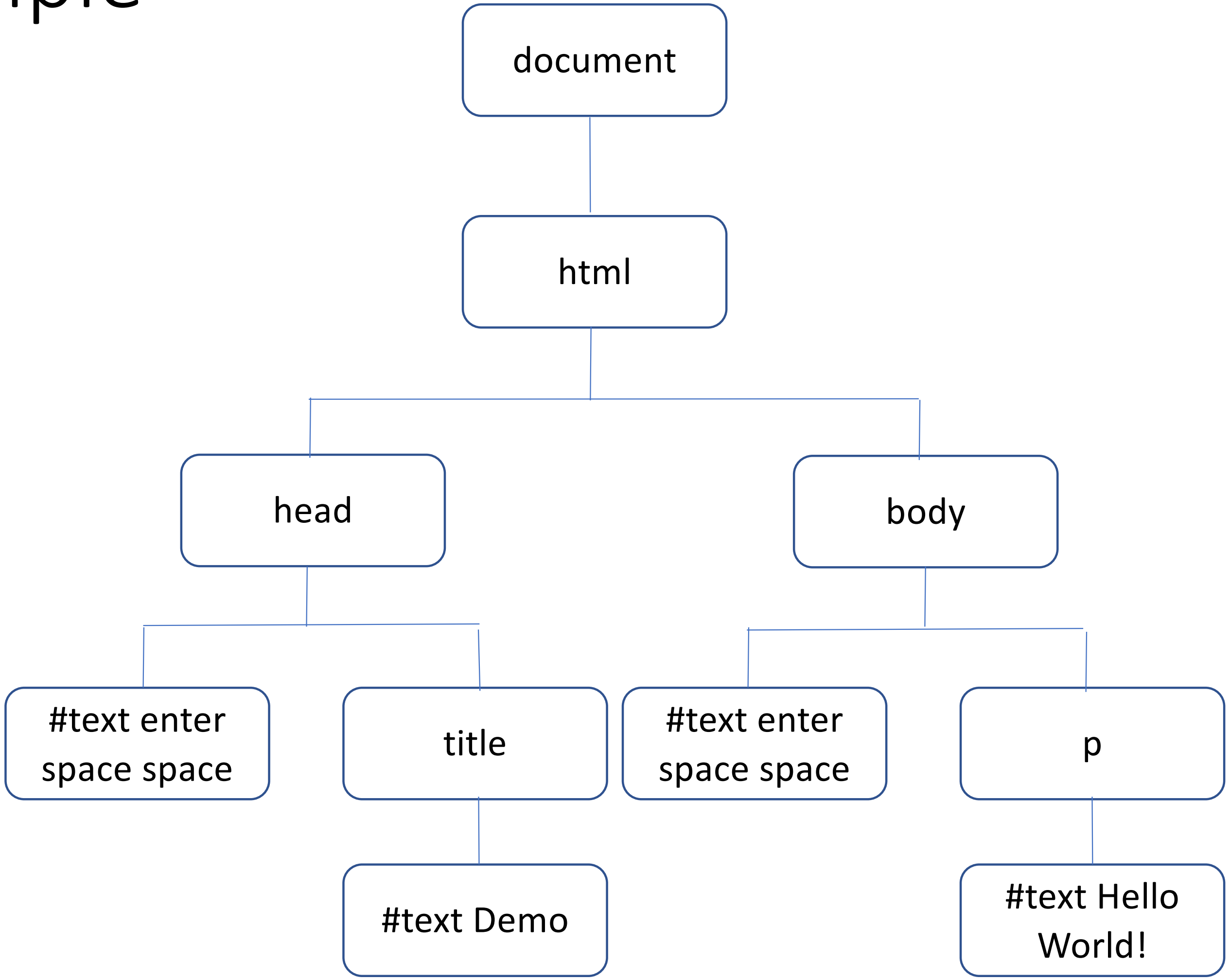
- DOM Manipulation
  - DOM tree
  - Finding elements in DOM tree
    - getElementById
    - querySelector
  - Traversing DOM tree
  - Adding &changing attributes
  - Creating & adding elements
- Events
  - Event listener

DOM tree

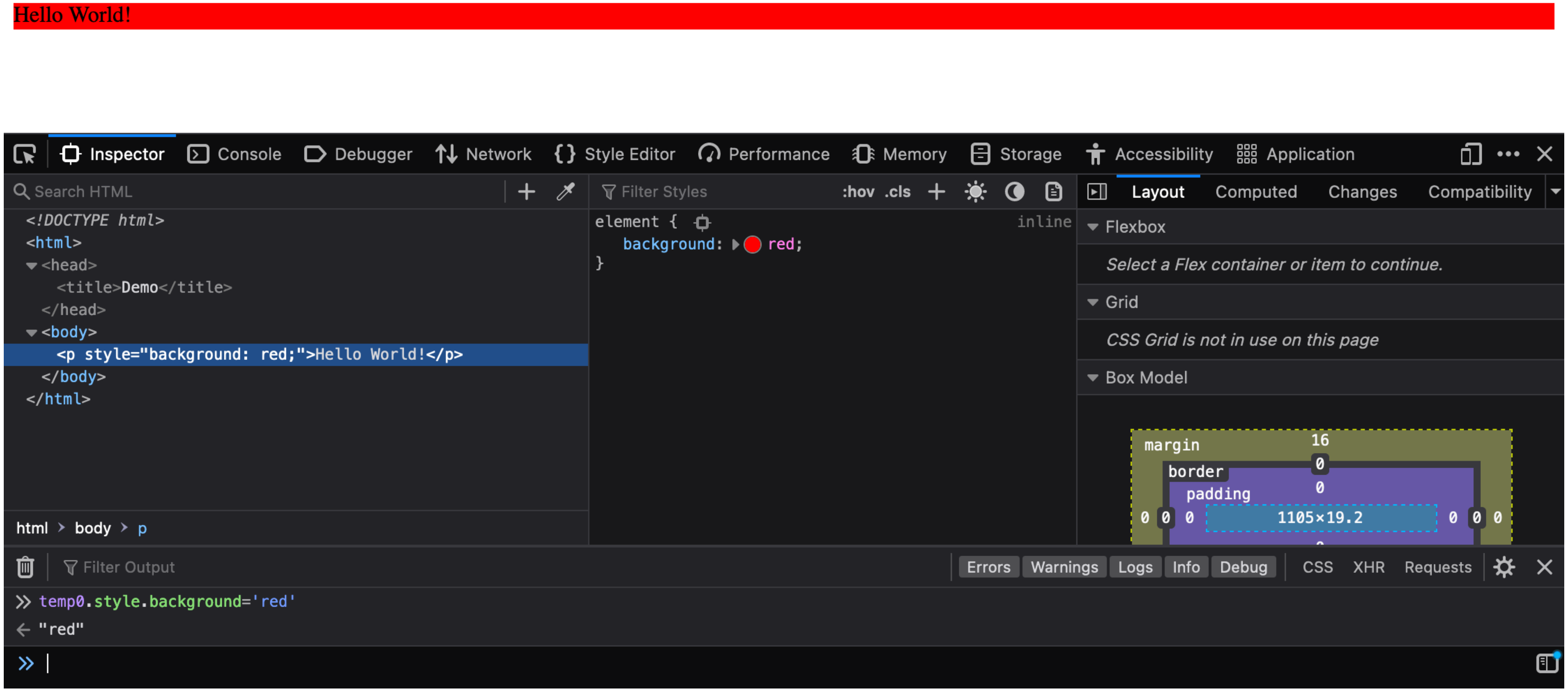
- DOM = Document Object Model
  - Every HTML element is an object
- DOM Tree
  - The HTML elements of a particular page are organised as a tree with parent nodes and children
    - Element nodes (HTML)
    - Text nodes
    - Attribute nodes

DOM tree - Example

```
<!DOCTYPE html>
<html>
<head>
  <title>Demo</title>
</head>
<body>
  <p>Hello World!</p>
</body>
</html>
```



DOM tree in Web Developer Tool



Hello World!

```
<!DOCTYPE html>
<html>
▾<head>
   <title>Demo</title>
 </head>
▾<body>
   <p style="background: red;">Hello World!</p>
 </body>
</html>
```

```
element {                                   inline
    background: ▸● red;
}
```

```
>> temp0.style.background='red'
← "red"
```

Finding elements in DOM tree

- getElementById
  - returns an Element object representing the element whose id property matches the specified string
  - since element IDs are required to be unique if specified, they're a useful way to get access to a specific element quickly

```
<form id="add-animal">
<p>
<label for="name">Name:</label>
<input id="name" type="text"/>
</p>
<p>
<input id="submit" type="submit" value="Add" />
</p>
</form>
```

```
document.getElementById("add-animal")
```

Finding elements in DOM tree

- querySelector
    - returns the first Element within the document that matches the specified (CSS) selector, or group of selectors
    - if no matches are found, null is returned

```html
<form id="add-animal">
<p>
<label for="name">Name:</label>
<input id="name" type="text"/>
</p>
<p>
<input id="submit" type="submit" value="Add" />
</p>
</form>
```

```javascript
document.querySelector('form')
```

Finding Elements in DOM tree

- https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementById
- https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementsByTagName
- https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementsByClassName
- https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelector
- https://developer.mozilla.org/en-US/docs/Web/API/Document/querySelectorAll

Traversing DOM tree

- Some methods to traverse throughout the DOM tree
  - parentNode
  - childNodes[*nodenumber*]
  - firstChild, lastChild
  - nextSibling, previousSibling

```html
<form id="add-animal">
<p>
<label for="name">Name:</label>
<input id="name" type="text"/>
</p>
<p>
<input id="submit" type="submit" value="Add" />
</p>
</form>
```

```javascript
// value of the input field with id name
// by id ...
// const name = document.getElementById("name").value
// by children ...
const name = document.getElementById("add-animal").children[1].children[1].value
```

Traversing DOM tree

- https://developer.mozilla.org/en-US/docs/Web/API/Element/children
- https://developer.mozilla.org/en-US/docs/Web/API/Node/firstChild
- https://developer.mozilla.org/en-US/docs/Web/API/Node/parentElement

Adding & changing attributes

- Sets the value of an attribute on the specified element. If the attribute already exists, the value is updated; otherwise a new attribute is added with the specified name and value.
  - setAttribute(name, value)
  - element.attribute = value

```
document.getElementById("name").type=”button”
```

```
<form id="add-animal">
<p>
<label for="name">Name:</label>
<input id="name" type="text"/>
</p>
<p>
<input id="submit" type="submit" value="Add" />
</p>
</form>
```

```
document.getElementById("name").setAttribute("style", "background-color: green")
OR
document.getElementById("name").style.background-color ="green”
```

Adding and changing attributes

- https://developer.mozilla.org/en-US/docs/Web/API/Element/setAttribute
- https://developer.mozilla.org/en-US/docs/Web/API/Element/removeAttribute

Creating & adding elements

- Some methods for creating and adding elements
  - createElement
  - appendChild

```
<main>
<h2>Animals currently on The Farm</h2>
<div id="animals"></div>
</main>
```

```
const p = document.createElement("p")
p.innerHTML = “something”
document.getElementById("animals").appendChild(p)
```

Events

- Events are actions or occurrences that happen in the system you are programming, which the system tells you about so your code can react to them.
  - The user selects, clicks, or hovers the cursor over a certain element
  - The user chooses a key on the keyboard
  - A web page finishes loading
  - A form is submitted
  - ...

Event listener

- addEventListener()
  - sets up a function that will be called whenever the specified event is delivered to the target
  - common targets are element, or its children, document, and window, but the target may be any object that supports events
- there also exists a function removeEventListener()

```
<form id="add-animal">
<p>
<input id="submit" type="submit" value="Add" />
</p>
</form>
```

```
document
  .getElementById("add-animal")
  .addEventListener("submit", (event) => {
        event.preventDefault()
        handleAddAnimal()
   })
```

Event listener

- https://developer.mozilla.org/en-US/docs/Web/Events#event_listing
  e.g. mouseover, mouseout, click, blur, focus, input

References

- https://developer.mozilla.org/en-US/docs/Learn/JavaScript
- https://www.freecodecamp.org/news/manage-default-behavior-in-browser/