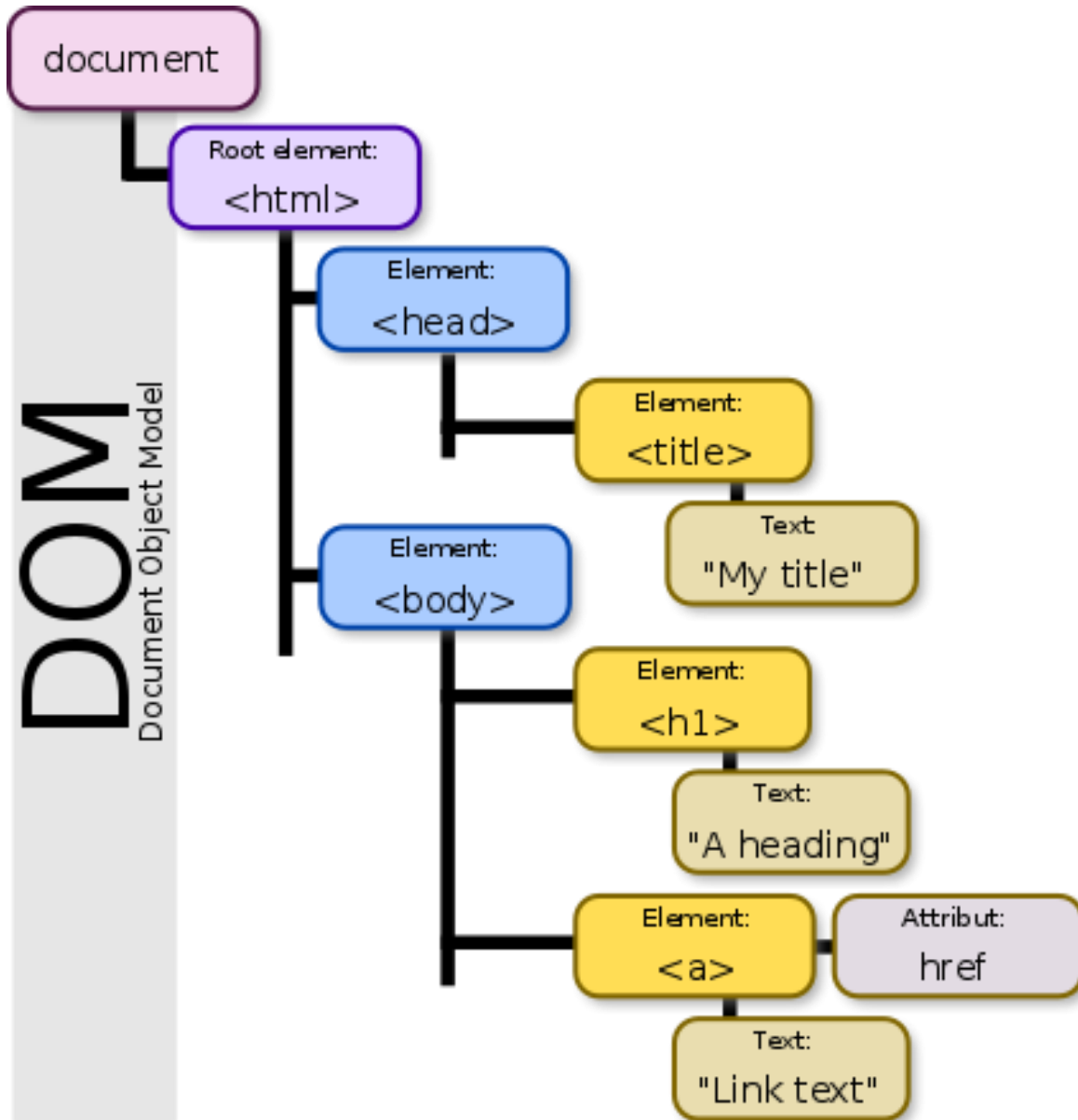


CS 1520: Recitation 2

DOM and Events

DOM – Document Object Model

- Created by the browser when a webpage is loaded
- We use JS to read or change the DOM



- Graphically, a tree of elements/nodes

Interacting with the DOM

- Edit HTML elements on the page
- Change element attributes, styles
- Inserting new HTML elements to the page
- Attach event listeners to elements

Get Element By ID

- `document.getElementById('Id_name')`
- Example (from <https://www.starwars.com/>)

```
> document.getElementById('nav-drawer');
```

```
< ▼<div id="nav-drawer">  
  ▼<nav id="nav-content">  
    ▶<div class="nav-login show">...</div>  
    ▶<ul id="section-links" itemscope itemtype="https://www.schema.org/  
SiteNavigationElement">...</ul>  
      <div class="bar-break"></div>  
    ▶<ul id="social-links">...</ul>  
  </nav>  
</div>
```

Get Elements By Class Name or Tag

- `document.getElementsByClassName("Class_Name")`
- `document.getElementsByTagName("Tag_Name")`
- Returns a **HTMLCollection** of elements with the class name and tag name passed as parameter respectively
 - **NOT** an array
 - Traverse using a *for* loop (not by using *forEach*)

```
> var classes = document.getElementsByClassName('nav-login show');
```

```
< undefined
```

```
> classes
```

```
< ▼ HTMLCollection(2) [div.nav-Login.show, div.nav-Login.show] ⓘ
```

```
  ▶ 0: div.nav-login.show
```

```
  ▶ 1: div.nav-login.show
```

```
    length: 2
```

```
  ▶ __proto__: HTMLCollection
```

```
> for(i = 0; i < classes.length; i++){  
    console.log(classes[i])  
}
```

```
  ▶ <div class="nav-login show">...</div>
```

VM4543:2

```
  ▶ <div class="nav-login show">...</div>
```

VM4543:2

```
< undefined
```

Query Selector

- `element = document.querySelector(selectors);`
- `selectors` are valid CSS selector strings
- `elements = document.querySelectorAll(selectors);`
 - Returns all elements with the matching selectors

Examples

- `<div id = "book-list">`
 - ``
 - ``
 - ``
book 1
 - ``
 - ``
 - ``
 - ``
book 2
 - ``
 - ``
 - ``
- `</div>`

- Query Selector to select the 2nd book?
- Query Selector to select all the books?

Examples

- `<div id = "book-list">`
 - ``
 - ``

``
Goblet of Fire
``

``
 - ``

``
Deathly Hallows
``

``
 - ``
- `</div>`

- Query Selector to select the 2nd book?
- `document.querySelector("#book-list li:nth-child(2) .name)`

Examples

- `<div id = "book-list">`
 - ``
 - ``

``
Goblet of Fire
``

``
 - ``

``
Deathly Hallows
``

``
 - ``
- `</div>`

- Query Selector to select all the books?
- `document.querySelectorAll(#book-list li .name)`

DOM Events

- List of DOM Events
 - https://www.w3schools.com/jsref/dom_obj_event.asp
- `element.addEventListener('click', function(e){})`

Event Bubbling and Capturing

- Bubbling Phase
 - Bottom up event trigger
- Capture Phase
 - Top down event trigger
- `element.addEventListener('click', function(e){}, userCapture)`

Event Bubbling

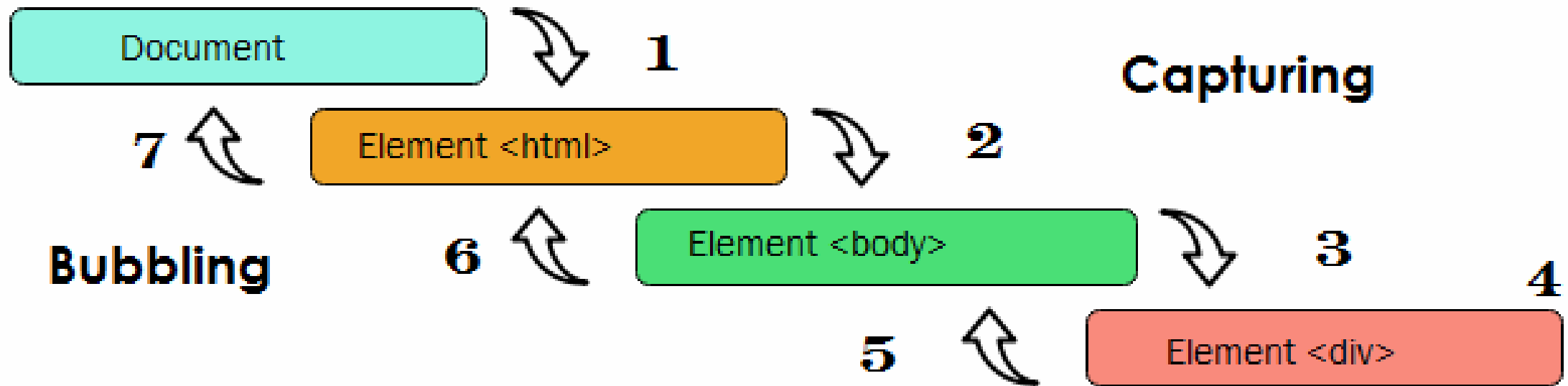
- Bubbling Phase
 - Bottom up event trigger
- Capture Phase
 - Top down event trigger

true => Event executed in the capture phase
false => Event executed in the bubbling phase



- `element.addEventListener('click', function(e){}, userCapture)`

Javascript Events : DOM Event Flow



Exercise 1 (See Rec2-ex1.html)

- What is the output in the console when
 - the *child* button is clicked?
 - the *other* button is clicked?

- **Output**

child clicked

parent clicked

Exercise 1.1(See Rec2-ex1_1.html)

- What is the output when child button is clicked?

- **Output**

parent clicked

child clicked

Exercise 2 (Rec2_exercise2.html)

- What alerts will be produced (and in what order) if a user clicks on `fuz`?
- What alerts will be produced (and in what order) if a user clicks on `baz`?
- If possible, write Javascript to replace the comment in the source code (`// YOUR CODE WOULD GO HERE`) such that clicking on `foo` will cause `Alert4` to be produced as the first alert without altering the other EventListeners. If this is not possible, explain why not.
- How would you target `<td> foo </td>` without using `getElementsByClassName`, `getElementById()`, `getElementsByTagName()`?

Exercise 2 (Rec2_exercise2.html)

- What alerts will be produced (and in what order) if a user clicks on `fuz`?
 - Alert 1, Alert 2

Exercise 2 (Rec2_exercise2.html)

- What alerts will be produced (and in what order) if a user clicks on `baz`?
 - Alert 1, Alert 3, Alert 2

Exercise 2 (Rec2_exercise2.html)

- If possible, write Javascript to replace the comment in the source code (``// YOUR CODE WOULD GO HERE``) such that clicking on ``foo`` will cause ``Alert4`` to be produced as the first alert without altering the other EventListeners. If this is not possible, explain why not.
- Can't do it unless you set capture to false on actionTable eventlistener. (If assigning eventlistener to the ``<td>foo</td>``). Because actionTable eventlistener takes priority.

Exercise 2 (Rec2_exercise2.html)

- How would you target `<td> foo </td>` without using `getElementsByClassName`, `getElementById()`, `getElementsByTagName()`?
 - `document.querySelector(#action-row td);`

References

- <https://www.w3.org/TR/DOM-Level-3-Events/#event-flow>
- <https://www.youtube.com/watch?v=sfKDOOJgbSI>
- <https://www.youtube.com/playlist?list=PL4cUxeGkcC9gfoKa5la9dsdCNpuey2s-V>
- http://www.tutorialspark.com/javascript/JavaScript_Event_Flow.php