

# CS 1520: Recitation – Week 7

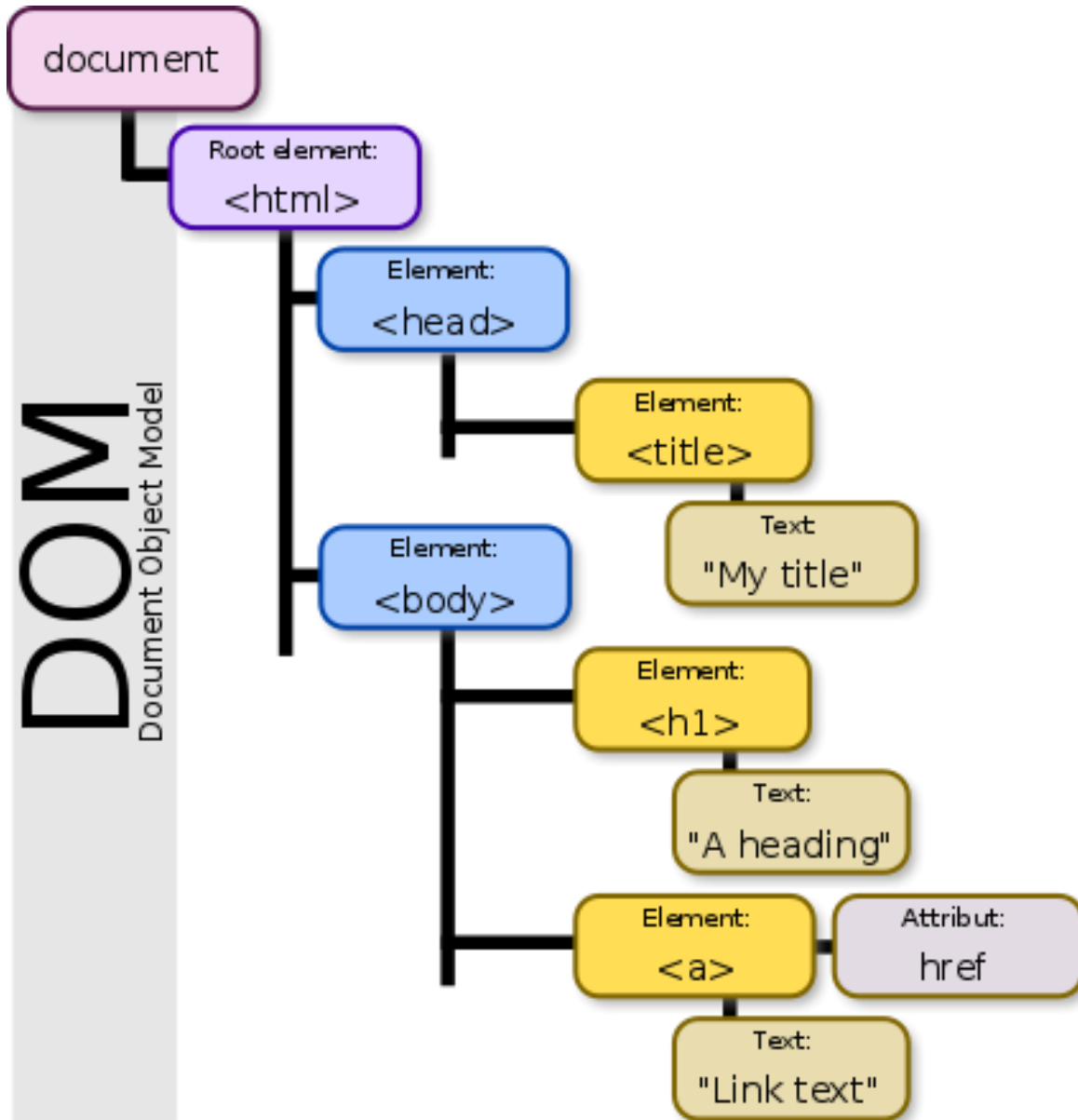
## **DOM**

# Outline

- DOM: Introduction
- Finding a HTML element
- Changing a HTML element
- DOM Events
  - Event Flow

# 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

# Finding HTML 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

## HTML Code Snippet

```
<div id = "book-list">
  <ul>
    <li>
      <span class = "name">
        Goblet of Fire
      </span>
    </li>
    <li>
      <span class = "name">
        Deathly Hallows
      </span>
    </li>
  </ul>
</div>
```

- Query Selector to select the 2<sup>nd</sup> book?
- Query Selector to select all the books?

# Examples

## HTML Code Snippet

```
<div id = "book-list">
  <ul>
    <li>
      <span class = "name">
        Goblet of Fire
      </span>
    </li>
    <li>
      <span class = "name">
        Deathly Hallows
      </span>
    </li>
  </ul>
</div>
```

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

# Examples

## HTML Code Snippet

```
<div id = "book-list">
  <ul>
    <li>
      <span class = "name">
        Goblet of Fire
      </span>
    </li>
    <li>
      <span class = "name">
        Deathly Hallows
      </span>
    </li>
  </ul>
</div>
```

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

# Changing HTML Content

## HTML Code Snippet

```
<p id="p0"> This is a recitation </p>
<p id="p1">This is CS 1550!</p>

<script>
document.getElementById("p1").innerHTML = "This
is CS 1520!";
</script>
```

## Output

```
This is a recitation
This is CS 1520!
```

# Ordering matters!

- Let's go back to last week's quiz
- How do you have the script in the head AND make it replace the content of the tag in the body?

# Changing HTML Attributes

## HTML Code Snippet

```


<script>
document.getElementById("image").src =
"landscape.jpg";
</script>
```

## Output

```
//Shows image of a
landscape
```



# Changing HTML Style

## HTML Code Snippet

```
<p id="p1">Penn State</p>
<p id="p2">University of Pittsburgh</p>
<p id="p3">Carnegie Mellon</p>

<script>
document.getElementById("p2").style.color = "blue";
document.getElementById("p2").style.fontFamily =
"Consolas";
document.getElementById("p2").style.fontSize =
"larger";
</script>
```

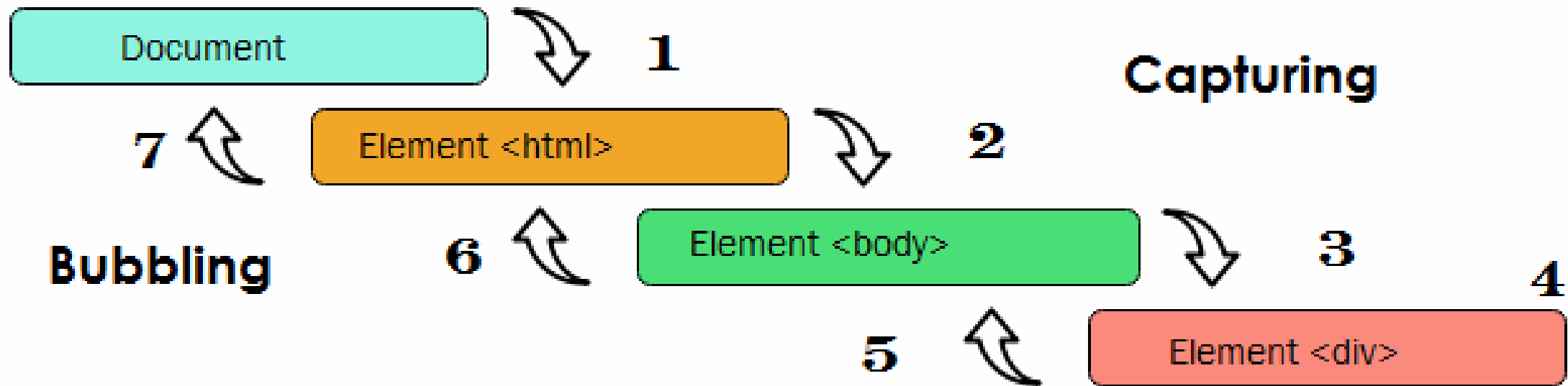
## Output

- Penn State
- University of Pittsburgh
- Carnegie Mellon

# DOM Events

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

# Javascript Events : DOM Event Flow



# Event Bubbling and Capturing

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

# Event Bubbling and Capturing

- 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)`

# Exercise 1 ( See week7-ex1.html)

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

- **Output 1**

child clicked

parent clicked

- **Output 2**

parent clicked

# Exercise 1.1(See week7-ex1\_1.html)

- What is the output when child button is clicked?

- **Output**

parent clicked

child clicked

## Exercise 2 (week7\_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 (week7\_exercise2.html)

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

# Exercise 2 (week7\_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 (week7\_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>``). This is because actionTable eventlistener takes priority.

# Exercise 2 (week7\_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](http://www.tutorialspark.com/javascript/JavaScript%20Event%20Flow.php)
- [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp)