

A1: z-index

A2: float

A3: 100%

A4: class

A5: Web-browser

A6 → ignore this question

A7: all of the above

A8: 59

A9:

	Physical layer	Transport layer	Network layer	Application layer
Put the message on the wire	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Responsible for routing messages	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Responsible for complete end to end delivery	<input type="radio"/>	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
Gives applications the ability to access the services of the other layers	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

B1: What is the difference between GET and POST methods used in forms?

- GET requests can be cached
 - GET requests remain in the browser history
 - GET requests can be bookmarked
 - GET requests should never be used when dealing with sensitive data
 - GET requests have length restrictions
 - GET requests are only used to request data (not modify)
-
- POST requests are never cached
 - POST requests do not remain in the browser history
 - POST requests cannot be bookmarked
 - POST requests have no restrictions on data length

B2: Explain the difference between “display:none” and “visibility:hidden”.

visibility:hidden hides the element, but it still takes up space in the layout. display:none removes the element from the document. It does not take up any space.

B3: 3101

B4:

getElementById: Get the element with the specified id

getElementsByTagName: returns a collection of elements with a specified name

getElementsByClassName: returns a collection of elements with a specified class name(s)

getElementsByTagName: returns a collection of all elements with a specified tag name

For example → see solutionB4_ex_03.html

B5: Explain event propagation and event bubbling using examples.

Event propagation

Propagation refers to how events travel through the Document Object Model (DOM) tree. The DOM tree is the structure which contains parent/child/sibling elements in relation to each other. You can think of propagation as electricity running through a wire, until it reaches its destination. The event needs to pass through every node on the DOM until it reaches the end, or if it is forcibly stopped.

Event bubbling

Bubbling and Capturing are the two phases of propagation. In their simplest definitions, bubbling travels from the target to the root, and capturing travels from the root to the target. However, that doesn't make much sense without first defining what a target and a root is.

The target is the DOM node on which you click, or trigger with any other event.

For example, a button with a click event would be the event target.

The root is the highest-level parent of the target. This is usually the document, which is a parent of the , which is a (possibly distant) parent of your target element.

For example → see solutionB5_ex_03_event_continuation.html and solutionB5_ex_03_event_propagation.html

C1

```
function isArmstrong(num){
    let i = num
    let sum = 0
    while (i > 0){
        let remainder = num%10
        sum += remainder*remainder*remainder
        i = parseInt(i/10)
    }
    if (num === sum){
        console.log("It is an Armstron Number")
    } else {
        console.log("It is not an Armstron Number")
    }
}
```

C2

/* Change the text color of all <p> elements inside the <div> element. */

```
div p{
    color: blue;
}
```

/* Change the background color of all <p> elements that are siblings of element. */

```
ul ~ p {
    background-color: yellow
}
```

/* Make the first letter size 20 for all the <p> elements. */

```
p::first-letter{
    font-size:20px;
}
```

/* Make the color of the <h1> red when the user hovers over the heading. */

```
h1:hover{
    color:red;
}
```

C3

```
let today = new Date()
let xmas = new Date(2024,11,25)

let isPrime = 1;
let days = parseInt((xmas.getTime()-today.getTime())/(24*60*60*1000))
console.log(days)
for (let i = 2; i < parseInt(days/2); i++){
  if (days%i == 0){
    isPrime = 0;
    break;
  }
}

if(isPrime == 0) alert(days + " left, " + days + " is not prime")
else alert(days + " left, " + days + " is prime")
```

C4 → see solutionC4_ex_03.html

C5 → canvas is no longer part of the course.