

NTN  
Maidam 479

MED

3

Table

(h2) A basic HTML table (h2)  
<table border="1">

```

<tr>
  <th> company </th>
  <th> contract </th>
  <th> country </th>
</tr>
<tr>
  <td> Alfreds </td>
  <td> maria </td>
  <td> germany </td>
</tr>
<tr>
  <td> centro </td>
  <td> Francisco </td>
  <td> mexico </td>
</tr>

```

```

<tr>
</table>
<br/>
<tr>
  <td colspan="3" border="1">
    <th> company </th>
    <th> contract </th>
    <th> country </th>
  </th>
</tr>

```

```

<tr>
  <td rowspan="3" border="1" colspan="3" rowspan="3" style="text-align: center;">
    <td> ID </td>
    <td> 2021-1-60-071 </td>
    <td> 2021-1-60-072 </td>
  </td>
</tr>
<tr>
  <td> 2021-1-60-073 </td>
  <td> 2021-1-60-074 </td>
</tr>
<tr>
  <td> 2021-1-60-080 </td>
  <td> 2021-1-60-081 </td>
</tr>

```

```

<tr>
</table>
<br/>
<table border="1">
<tr>
  <th> group </th>
  <th> 1 </th>
  <th> 2 </th>
</tr>
<tr>
  <td rowspan="3" border="1" colspan="3" rowspan="3" style="text-align: center;">
    <td> ID </td>
    <td> 2021-1-60-071 </td>
    <td> 2021-1-60-072 </td>
  </td>
</tr>
<tr>
  <td> 2021-1-60-073 </td>
  <td> 2021-1-60-074 </td>
</tr>
<tr>
  <td> 2021-1-60-080 </td>
  <td> 2021-1-60-081 </td>
</tr>

```

</table>

(Am)

A basic HTML TABLE

Company	Contract	Country
Alfreds	maria	germany
centro	Francisco	mexico

group		1	2
ID		2021-1-60-071	2021-1-60-072
		074	
		080	081

(h2) A basic HTML table (h2)

```

<table border="1">
<tr>
  <th> company </th>
  <th> contract </th>
  <th> country </th>
</tr>
<tr>
  <td> Alfred </td>
  <td> maria </td>
  <td> germany </td>
</tr>
<tr>
  <td> centro </td>
  <td> Francisco </td>
  <td> mexico </td>
</tr>

```

</table> . (Am)

(h2) A Basic HTML Table (h2) <table border="1">

```

<tr>
  <th> group </th>
  <th> 1 </th>
  <th> 2 </th>
</tr>
<tr>
  <td rowspan="3" border="1" colspan="3" rowspan="3" style="text-align: center;">
    <td> ID </td>
    <td> 2021-1-60-071 </td>
    <td> 2021-1-60-072 </td>
  </td>
</tr>
<tr>
  <td> 2021-1-60-073 </td>
  <td> 2021-1-60-074 </td>
</tr>
<tr>
  <td> 2021-1-60-080 </td>
  <td> 2021-1-60-081 </td>
</tr>

```

</table>

# Model Initiative

## Ch2) Student Proforma (1 hr)

<form>

<p><b> My favorite web language : </b></p>  
{<input type = "radio" name = "weblanguage" value = "HTML" > HTML<br><input type = "radio" name = "weblanguage" value = "CSS" > CSS<br><input type = "radio" name = "weblanguage" value = "JS" > JS<br><br><p><b> my favorite programming task : </b></p>  
{<input type = "checkbox" name = "programming\_task" value = "Front End" > Front End<br><input type = "checkbox" name = "programming\_task" value = "Back End" > Back End<br><input type = "checkbox" name = "programming\_task" value = "DB" > DB<br><br><p><b> my current passion : </b></p>  
<select name = "Passion">  
{<option> Academia</option>  
<option> Industry</option>  
<option> Research</option>  
<option> Enterpro</option>  
</select><br><br><p><b> My name : </b></p>  
{<input type = "text" name = "name" value = "Mr. John Don" > ~~Mr. John Don~~<br><input type = "submit" value = "Submit" >  
<input type = "reset" value = "Rest" >  
</form>

## Ch2) Student proforma Ch2

<form>

○ HTML  
○ CSS  
○ Java  
Script  
{<p><b> My favorite web language : </b></p>  
{<input type = "radio" name = "weblanguage" value = "HTML" > HTML<br><input type = "radio" name = "weblanguage" value = "CSS" > CSS<br><input type = "radio" name = "weblanguage" value = "Javascript" > Javascript<br><br>

□ Front-End  
□ BACK END  
■ Database

<p><b> My favorite programming task : </b></p>

{<input type = "checkbox" name = "programming\_task" value = "Front-End" > Front-End<br><input type = "checkbox" name = "programming\_task" value = "Back-End" > Back End<br><input type = "checkbox" name = "programming\_task" value = "Database" > Database<br>

<p><b> My current passion : </b></p>

{<select name = "Passion">  
<option> Academia</option>  
<option> Industry</option>  
<option> Research</option>  
<option> Enterpro</option>  
</select><br>

<p><b> My name : </b></p>

<input type = "text" name = "name" value = "Mr. John Don" >  
<input type = "text" name = "name" value = "2025-1-60-023" >  
<input type = "reset" value = "Rest" >

</form>

(Am)

```
<h2> My Favorite Drinks: </h2>
<p> <strong> In unorderend list: </strong> </p>
<ul style="list-style-type: circle;">
<li> coffee </li>
<li> Tea </li>
<li> Juice </li>
</ul>
<br>
<p> <strong> In ordered lists: </strong> </p>
<ol>
```

```
<h2> My favorite Drinks: </h2>
<p> <strong> In unorderend lists: </strong> </p>
<ul style="list-style-type: circle;">
<li> coffee </li>
<li> Tea </li>
<li> Juice </li>
</ul>
<br>
<ul style="list-style-type: square;">
<li> coffee </li>
<li> Tea </li>
<li> Juice </li>
</ul>
```

```
<p> <strong> In order lists: </strong> </p>
<ol type="I">
<li> coffee </li>
<li> Tea </li>
<li> juice </li>
</ol>
```

```
<ol type="a">
<li> coffee </li>
<li> Tea </li>
</ol>
```

```
<h2> My favorite Drinks: </h2>
<p> <strong> In unorderend list: </strong> </p>
<ul style="list-style-type: circle;">
<li> coffee </li>
<li> Tea </li>
<li> Juice </li>
</ul>
<br>
<ul style="list-style-type: square;">
<li> coffee </li>
<li> Tea </li>
<li> Juice </li>
</ul>
<br>
<p> <strong> In ordered list: </strong> </p>
<ol style="list-style-type: i;">
<li> coffee </li>
<li> Tea </li>
<li> Juice </li>
</ol>
<br>
<ol style="list-style-type: a;">
<li> coffee </li>
```

<ol>

<dt> sashimi </dt>

<dd> sliced raw fish that is served </dd>

<dt> scale </dt>

<dd> A device used to accuratly </dd>

<dd> A technique by which the </dd>

<dt> scamorze </dt>

<dt> scamorze </dt>

output

sahimi

sliced

scale

device

A technique

scamorze

scamorze

<li> Tea </li>
<li> Juice </li>
</ol>

(An)

<ol>

<dt> schimit </dt>

<dd> ABCD </dd>

</ol>

output

schimit

ABCD

(An)

<ol style="list-style-type: a;">
<li> coffee </li>

`<style>`

`#Par1, #Par2, #Par3`

`h1`

`{`

`font-family: Arial;`

`text-align: center;`

`text-align: justify;`

`font-family: Arial`

`text-transform: capitalize;`

`color: green;`

`letter-spacing: 1px;`

`color: black;`

`<div>`

`<h1> TEXT FORMATTING </h1>`

`<p id="Par1">`

This text is styled some of the

`<p>`

`<p id="Par2">`

This is another paragraph

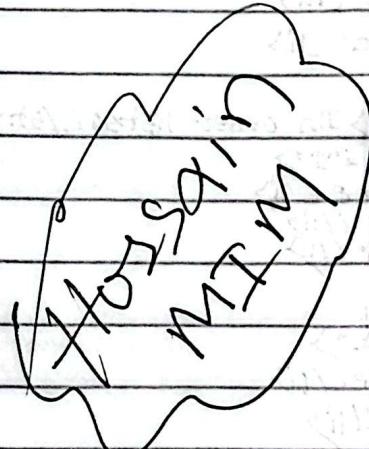
`<p>`

`<p id="Par3">`

This is a third paragraph

`</p>`

`</div>`



Code:—

```
<div>
  <a href="https://mail.google.com" target="blank">
    
  </a>
</div>
```

(a) tag:— The href attribute specifies the URL to redirect when the image is clicked.

<img> tag:— The src attribute provides the image source URL and alt specifies alternative text.

• Target Attribute:— target="blank" opens the link in a new tab for better user experience.

Code:—

```
<div>
  <a href="https://mail.google.com" target="blank">
    
  </a>
</div>
```

(a) tag:— The href attribute specifies the URL to redirect when the image is clicked.

<img> tag:— The src attribute provides the image source URL and alt specifies alternative text.

target="blank":— target="blank" opens the link in a new tab for better user experience.

The three(03) semantic markups includes <P>, <Pre> and <Span>. Among these the <Pre> tag is used to preserve spaces and line breaks in the HTML.

<P>

(A) This paragraph contains a lot of lines

</P>

<Pre> (preformatted tag)

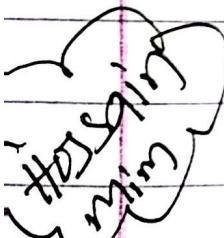
This paragraph

containing a lot of space  
in the source code

</Pre>

<P>

The number of lines on the <Span style="font-weight: bold;"> size  
</Span> of the browser

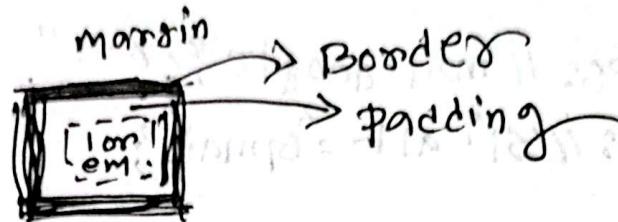
(B) 

<P>

This number of lines in a paragraph  
depend on the <Span style="font-weight: bold;"> size  
</Span> of the browser window

</P>

Border



<style>

.box {

width: 150 px;

height: 100 px;

margin: 50 px;

padding: 10 px;

border: 10 px red;

background-color:

text-align:

box-sizing:

</style>

MD

3

{更多}

<h2> Registration Form </h2>

<form>

<div class="box">

lorem ipsum

</div>

<style>

.box {

width: 150 px;

height: 100 px;

margin: 50 px;

padding: 10 px;

border: 10 px red;

text-align: justify;

</style>

<div class="box">

· ABCD

</div>

(AM)

Border:— Every Box has a border even if it is invisible to be "0" px wide. It separates the edge of one box from another.

margin:— sits outside the edge of the border

padding:— The space between the border of a box and its content.

(AM)

- JS
- ✓(iii) const is block-scoped
  - ✗(iv) Implicit global variables (no declaration necessary)
  - ✗(v) var is function-scoped or globally-scoped
  - ✓(vi) let is block-scoped

var x = 10;

Output 1: 10 The value of "x" declared with var

let y = 20;

Output 2: 20 The value of "y" declared with let

const z = 30;

Output 3: 30 This means "z" is a const

console.log(x); → 01

Output 4: 100 If block, x gets reassigned

console.log(y); → 02

Output 5: 200 y declared inside the block with let

if (true)

Output 6: 300 block in const

var x = 100;

Output 7: 50 The variable a is implicitly declared  
on keyword (var, let, const) is used

let y = 200;

Output 8: 100 The globally-scoped var x was  
overwritten to 100 before it

const z = 300;

Output 9: 20 The value is same let is block-scoped

a = 50;

Output 10: 30 The value is same const is block-scoped

console.log(x); → 04

Output 11: 50

console.log(y); → 05

Output 12: 20

console.log(z); → 06

Output 13: 30

console.log(a); → 07

}

console.log(x) → 08

Output 14: 50

console.log(y) → 09

Output 15: 20

console.log(z) → 10

Output 16: 30

console.log(a) → 11

~~output~~  
Total price of 6 kg date is : BDT 10800  
Total price after discount is : BDT 97200  
Regular Date - Total Price of 10 kg is BDT 8500

class Date {

constructor(amount, unitPrice) {

this.amount = amount;  $u(kg)$

this.unitPrice = unitPrice;

~~Body~~

~~Date~~

~~Class~~

calculateDatePrice() {

console.log('Total price of this.amount  $kg$  date is: BDT {this.amount \* this.unitPrice}');

class premiumDate extends Date {

constructor(amount, unitPrice, discount = 0) {

super(amount, unitPrice);

this.discount = discount;

}

get discount() {

return this.discount;

}

set discount(value) {

this.discount = value;

}

calculateDiscount() {

~~const~~ totalPrice = this.amount \* this.unitPrice;

~~const~~ discountPrice = totalPrice - (totalPrice \* this.discount / 100);

/100;

console.log('Total price after discount is: BDT {this.discount price}');

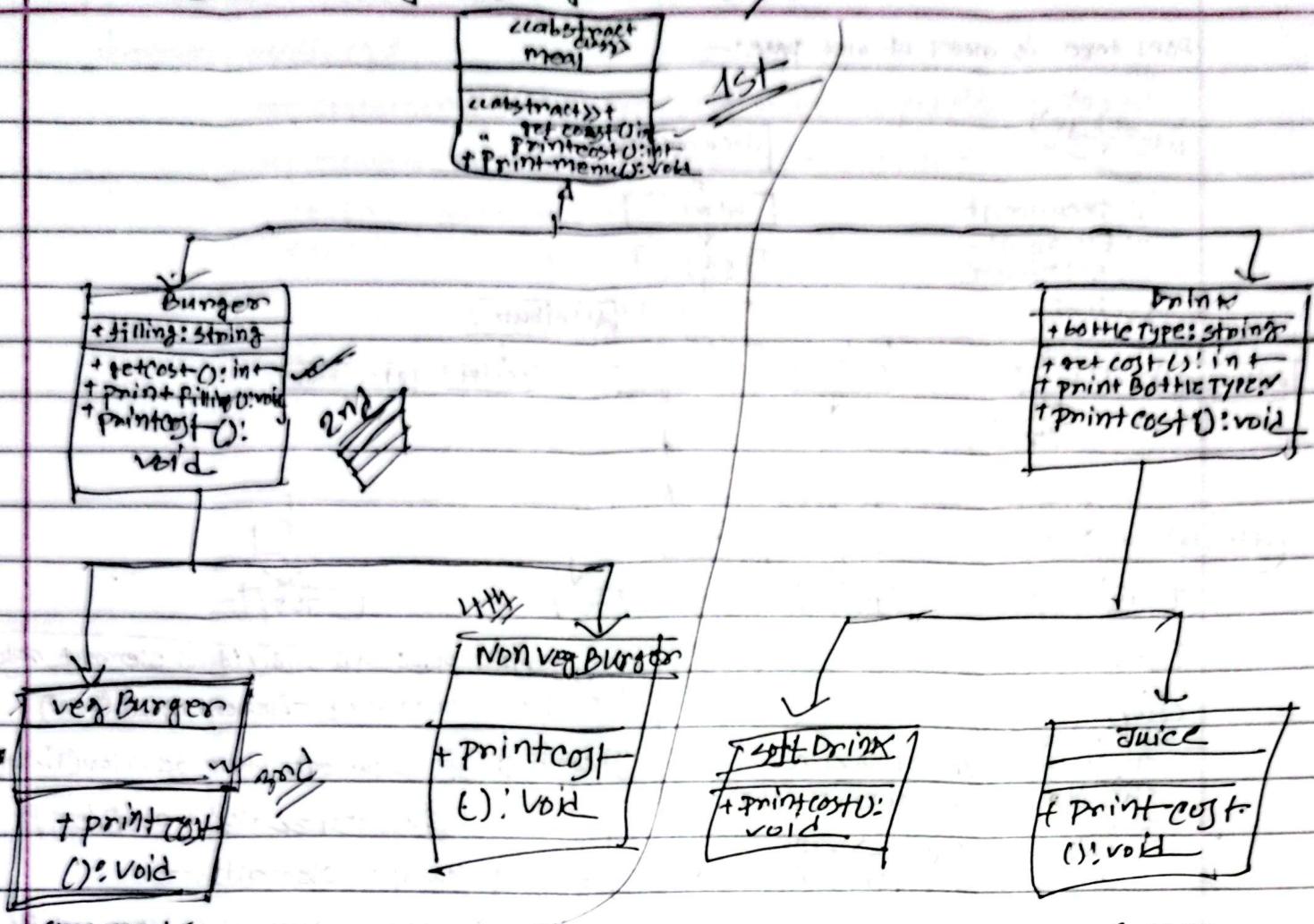
~~R~~  
class regularDate extends Date {

calculateDatePrice() {

console.log('Regular Date - Total price of this.amount  $kg$  is: BDT {this.amount \* this.unitPrice}');

menu items veg burger  
 filling: vegetable filling  
 veg burger cost: BDT 200

Read the following class diagram carefully:



class meal {  
 constructor(name){  
 if (new.target == meal) {  
 throw new Error ("cannot instantiate");  
 }  
 this.name = name;  
 }  
 getCost(){  
 throw new Error ("Am't getCost implemented");  
 }  
 printCost(){  
 throw new Error ("Am't printCost implemented");  
 }  
 printMenu(){  
 console.log ('menu item: ' + this.name);  
 }  
}

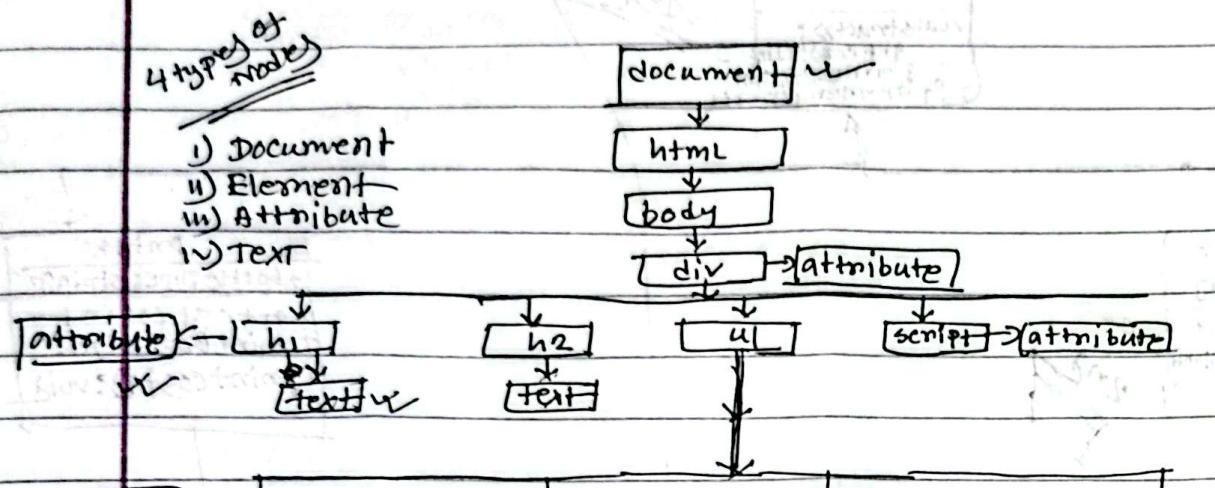
class Burger extends meal {  
 constructor(name, filling){  
 super(name);  
 this.filling = filling;  
 }  
 printFilling(){  
 console.log ('filling: ' + this.filling);  
 }  
 printCost(){  
 console.log ('this.name with this.filling cost : BDT 250');  
 }  
}

class vegBurger extends Burger {  
 constructor(){  
 super("vegBurger", "vegetable Filling");  
 }  
 printCost(){  
 console.log ('veg Burger costs: BDT 200');  
 }  
}

class nonvegBurger extends Burger {  
 constructor(){  
 super("non-veg Burger", "chicken Filling");  
 }  
 printCost(){  
 console.log ('non-veg Burger cost: BDT 300');  
 }  
}

The DOM is working with an element == the DOM working with a node the represents that HTML element

DOM tree is model of web page:-



```

<html>
  <body>
    <div id="page">
      <h1 id="headline">List</h1>
      <h2 id="Buy groceries</h2>
      <ul>
        <li id="one" class="hot">
          <em>fresh</em> avocados
        </li>
        <li id="two" class="hot">
          cashew nuts
        </li>
        <li id="three" class="hot">honey</li>
        <li id="four" >buy balsamic vinegar</li>
      </ul>
      <script src="js/list.js"></script>
    </div>
  </body>
</html>
  
```

DOM Queries: select an individual element node

- 1) const el = document.getElementById('one')
  - 2) const elFirst = document.querySelector('li.hot');
- Quickest & most efficient

Selecting multiple elements:-

- 1) const elements = document.getElementsByClassName('hot');

- 2) const element = document.getElementById('one');

getElementsByClassName('li');

- 3) const element = document.querySelectorAll('li.hot');

return array of objects  
contains many nodes

## IIFE → Immediately Invoked Function Expression -

```

let value = (function (ccp, credit) {
    let sumCCP = 0;
    let sumCR = 0;
    for (var i = 0; i < ccp.length; i++) {
        sumCCP += ccp[i] * credit[i];
        sumCR += credit[i];
    }
    return sumCCP / sumCR;
})
(ccp, credit);
console.log(value);

```

3.5

~~MID Exercise Solving~~

Hossain MIM

```

function Person(first, last) {
    this.first = first;
    this.last = last;
    this.fullname = function () {
        return this.first + ' ' + this.last;
    }
    this.fullnameReversed = function () {
        return this.last + ', ' + this.first;
    }
}

```

```

person.prototype.greet = function () {
    return this.first + " say hi";
}

```

```

const simon = new Person("simon", "wilson");
console.log(simon.greet());
console.log(simon.newPerson("Delvin", "Thomas").greet());

```

simon say hi  
Delvin say hi

## Anonymous Function

```

const findNegPos = function (num) {
    if (num == 0) {
        console.log("The number is zero");
    } else {
        console.log("The number is positive");
    } else {
        console.log("The number is negative");
    }
}
findNegPos(-12)

```

## Arrow Function

An arrow function uses the  $\Rightarrow$  thus syntax for shorter function

```
const findNegPos = (number) => {
```

Am to the question no:01

```
<h2> My favorite drinks: </h2>
<ul style="list-style-type: none;">
  <li> coffee </li>
  <li> Tea </li>
  <li> Juice </li>
</ul>
<br>
<ul style="list-style-type: square;">
  <li> coffee </li>
  <li> Tea </li>
  <li> Juice </li>
</ul>
```

~~Q1~~ <ul style="list-style-type: none;">
 <li type="A">
 <li> coffee </li>
 <li> Tea </li>
 <li> Juice </li>
 </li>

(Am)

Am to the question no:02

```
<dl>
  <dt> sashimi </dt>
  <dd> ABCD </dd>
  <dt> scale </dt>
  <dd> A device </dd>
  <dt> A technique </dt>
  <dd> CSE </dd>
  <dt> CSE </dt>
</dl>
```

⇒

output

sashimi

ABCD

scale

A device

A technique

Am to the question no:03

{style}

```
width: 150px;
height: 100px;
margin: 50px;
border: 10px red;
padding: 100px;
text-align: justify;
```

```
</style>
<div class="box">
  ABCD MIMI
</div>
```



output

margin

Border

padding

(Am)

Am to the question no:04.

<h2> A basic HTML table </h2>
<table border="1">
 <tr>
 <th> company </th>
 <th> contract </th>
 <th> company </th>
 </tr>
 <tr>
 <td> A </td>
 <td> B </td>
 <td> C </td>
 </tr>
</table>

00

<h2> A basic table </h2>

```
<table border="1">
  <tr>
    <th> group </th>
    <th> 1 </th>
    <th> 2 </th>
  </tr>
  <tr>
    <td rowspan="3">ID</td>
    <td> 2021-1-60-071 </td>
    <td> 2021-1-60-072 </td>
  </tr>
  <tr>
    <td> 081 </td>
    <td> 082 </td>
  </tr>
</table>
```

(Am)

### Ans to the question no: 05

a) `a` tag:- The href Attribute

specify the "URL" to redirect when the image is clicked.

b) `img` tag:- The src attribute provides the image source "URL" and alt specifies alternative text.

c) Target Attribute:- target = "blank" opens the link in a new tab for better user experience.

Code implement:-

```
<div>
  <a href="https://mail.google.com" target="blank">
    
  </a>
</div>
```

### Ans to the question no: 03

The Three(`o3`) semantic markups (`p` (`pms`) and (`span`) use the code.

Among the (`p`) tag used the para & turn (`par`) tag used space and the (`span`) & `o3` use to specify the word or line edit of the page.

Code:-

```
<p>
  This paragraph
  contains a lot of
  in the source code
</p>
```

`pms`

This paragraph
contains
in the source code

</pms>

`p`

The number `span` style
"font weight: bold;"  
of line `span`  
in a paragraph depen

</p>

Ans

Stand structural:-

<html><body><img>

Semantic:-

Not table

### Ans to the question no: 08

<h2> Student Profile </h2>

<form>

<p><b> my favorite web language </b></p>

<input type="radio" name="A" value="HTML"> HTML <br>

<input type="radio" name="B" value="C++"> C++ <br>

<input type="radio" name="C" value="JS"> JS <br>

<p><b> My favorite programming tools : </b></p>

<input type="checkbox" name="A" value="Front"> Front <br>

<input type="checkbox" name="B" value="Back"> Back <br>

<input type="checkbox" name="C" value="DB"> DB <br>

<p><b> my current passion : </b></p>

<select name="passion">

<option> Academic </option>

<option> Industry </option>

<option> Research </option>

<option> ABCD </option> <br> <br>

<p><b> my name : </b></p>

<input type="text" name="name" value="HOSSAIN MEM" > <br> <br>

<input type="submit" value="Submit" >

<input type="reset" value="Reset" >

</form>

Ans

## Table

Aյ to take question no: 07

(form) (b) Personal information (p)

\* (label) Email ID: (label) (br)
 { input type = "email" name = "A" placeholder = "Enter your E-mail ID" required } (br) (br)

(label) Password: (label) (br)
 { input type = "password" name = "B" placeholder = "Enter your password" required } (br) (br)

(label) Security Question: (label) (br)
 { select name = "question" required }
 option value = "Select a question" Select a question (option)
 option value = "What is your name?" What is your name? (option)
 option value = "What's your name?" What's your name? (option)

(br) (br)

(label) Answer: (label) (br)
 { input type = "text" name = "A" value = "Mimbin Hussain" } (br) (br)

(p) (b) Personal Information

(label) Full names: (label) (br)
 { input type = "text" name = "A" placeholder = "Enter your name" required } (br) (br)

(label) Date of birth: (label) (br)
 { input type = "date" name = "date" } (br)

(label) NID Number: (label)
 { input type = "tel" name = "A" placeholder = "Enter your Phone No" required } (br)

(br) (br)

(/form) (Any)

Aյ to take question: \*

```

class meal {
  constructor(name) {
    if (new.target === meal) {
      throw new Error("cannot abstract");
    }
    this.name = name;
  }
  get cost() {
    throw new Error(`getcost implement`);
  }
  printCost() {
    throw new Error(`printcost implement`);
  }
  printMenu() {
    console.log(`menu Item: ${this.name}`);
  }
}

class Burger extends meal {
  constructor(name, filling) {
    super(name);
    this.filling = filling;
  }
  printFilling() {
    console.log(`filling: ${this.filling}`);
  }
  printCost() {
    console.log(`${this.name} with ${this.filling} cost 200BDT`);
  }
}

class vegBunge extends Burger {
  constructor() {
    super("vegBunge", "vegetable Filling");
  }
  printCost() {
    console.log(`vegBunge cost: 200BDT`);
  }
}

```

(Any)

### Ans to the question no: 13

class Date:

```

class PremiumDate extends Date {
    constructor(amount, unitprice, discount) {
        super(amount, unitprice);
        this.discount = discount;
    }
    get discount() {
        return this.discount;
    }
    set discount(value) {
        this.discount = value;
    }
    calculateDiscount() {
        const totalprice = this.amount * this.unitprice;
        const discountPrice = totalprice - (totalprice * this.discount);
        console.log(`Total of ${this.amount} kg premium date after ${this.discount}% discount`);
    }
}

```

```

class Regular extends Date {
    constructor(amount, unitprice, discount) {
        super(amount, unitprice);
    }
}

```

class Regular extends Date { }

```

const totalprice = this.amount * this.unitprice;
console.log(`Total of ${this.amount} kg premium date after ${this.discount}% discount`);

```

```

const premiumDate = new PremiumDate(6, 180, 10);
premiumDate.calculateDiscount();

```

(A)

### Anonymous

```

const findNegPos = function (number) {
    if (number < 0) {
        return "Negative";
    } else {
        return "Positive";
    }
}

```

An arrow function used due to this syntax for shorter function

### IIFE

### Ans to the question no: 14

```

let value = (function (corp, credit) {
    let sumCorp = 0.0;
    let sumCredit = 0;
    for (var i = 0, k = corp.length; i < k; i++) {
        sumCorp += corp[i] * credit[i];
        sumCredit += credit[i];
    }
    return sumCorp / sumCredit;
})([3, 3.5, 9], [3, 3, 4]);
console.log(value);

```

Output:- 3.5

(Ans)

### transform <script>

### Ans to the question no: 12

```

function transform(name) {
    const nameInput = document.getElementById('name');
    let nameValue = nameInput.value.trim();
    if (nameValue) {
        let uppercaseName = nameValue.toUpperCase();
        const greetingMessage = `Hi ${uppercaseName}!!! Good Afternoon!`;
        document.getElementById('greeting-message').innerHTML = greetingMessage;
        nameInput.value = uppercaseName;
    } else {
        alert("Please Enter your name first!");
    }
}

```

</script>

(Ans)

### Any to any question no: 16

<script>

```
const firstDiv = document.getElementById('div1');
const secondDiv = firstDiv.nextElementSibling;
const thirdDiv = secondDiv.nextElementSibling;
const newParagraph1 = document.createElement('p');
newParagraph1.textContent = "This is Real time data for Alex";
const newParagraph2 = document.createElement('p');
newParagraph2.textContent = "This is Real time data for Romeo";
if (secondDiv) {
    secondDiv.appendChild(newParagraph1);
}
thirdDiv.appendChild(newParagraph2);
```

</script>

- i) Access the 1st Div using the ID.
- ii) use the nextElementSibling property to access the 2nd and 3rd Div.
- iii) Create a new<p> element here dynamically
- iv) Append the new<p> to the 2nd and 3rd divs.

### Any to any question no: 15 (02)

Output:

simon says hi

Delvin says hi

iv) Append the new(p) to the 2nd and 3rd div's

Any to the question no : 15 (02).

Output:

simon says hi

delvin says hi

→ Question type for MID EXAM

Topic-1

i) ordered list →

ii) unordered list →

iii) definition list →

iv) nested list →

v) Image page (45-48) → Quiz Structure Question & NTN made em

100% Important →

Exam 2010

Topic-2

css into short code only →

Topic-3

Question 2010 →

Topic-4

i) IIFE page 30)

ii) Arrow function → 11

iii) avoid var code - 13

iv) Invocation

v) TODO

vi) closure → most important

Topic-5

i) domtree is model of web page

↳ code over tree

↳ tree over code