Week2 Live session

Prerequisite:
Install VS code IDE https://code.visualstudio.com/ Type your html, CSS code and run it through vs code extension `Live server`.

CDN https://getbootstrap.com/
Bootstrap 5.0: https://getbootstrap.com/docs/5.0/getting-started/introduction/
00000000

Q1 How many bits are required to store a text document consisting of 8000 alphanumeric characters using 8-bit ASCII?

A. 32000 bits
B. 64000 bits
C. 16000 bits
D. 42000 bits

1. **8-bit ASCII**: Each character requires 8 bits.

   For 8000 characters:
   8000 characters×8 bits/character=64000 bits

2. **UCS-2**: Each character in UCS-2 uses 16 bits.

   For 8000 characters:
   8000 characters×16 bits/character=128000 bits

3. **UCS-4**: Each character in UCS-4 uses 32 bits (4 bytes).

   For 8000 characters:
   8000 characters×32 bits/character=256000 bits

4. **UTF-8**: This can vary since UTF-8 uses 1 to 4 bytes per character depending on the character. For basic Latin characters, it's 8 bits (1 byte).

   For 8000 basic Latin characters:
   8000 characters×8 bits/character=64000 bits

Week2 Live session

Q2. Let `L = {'a', 'b', 'c', 'd', 'A', 'B', 'C', 'D', '0', '1', ' '}` be a complete character set (i.e., only these characters can be used to represent text in the document). If a document that uses fixed encoding for all characters is created using the character set L and has a disk size of 2 Kilobytes, the number of characters in the documents would be_____. [Take 1 Byte = 8 bits, 1 KB = 1000 Bytes, 1 MB = 1000 Kilobytes and so on.]

[MCQ: 3 points]

A. 2000
B. 4000
C. 8000
D. 16000

Answer:

To determine the number of characters in the document, let's follow the steps carefully:

1. **Disk Size**: The document size is given as 2 Kilobytes:
   2 KB=2×1000 Bytes=2000 Bytes
2. **Character Set Size**: The character set L consists of 11 characters: {'a', 'b', 'c', 'd', 'A', 'B', 'C', 'D', '0', '1', ' '}.
3. **Bits Required to Represent Each Character**: Since there are 11 characters, we need enough bits to represent all of them. The smallest power of 2 that can represent 11 characters is 4 bits (since 2^4 = 16 combinations).
4. **Calculating Number of Characters**:
   - Each character requires 4 bits.
   - Total number of bits in the document: 2000 Bytes×8 bits/Byte=16000 bits
   - Number of characters that can be stored:

     Number of characters=16000 bits/4 bits/character=4000

Thus, the number of characters in the document would be **4000**.

Week2 Live session

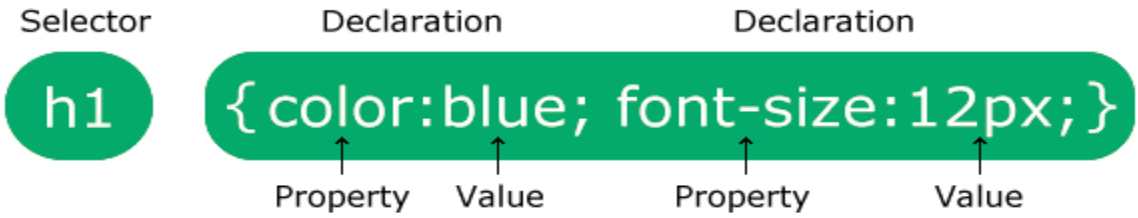
Q3. Which of the following is a correct way to create a hyperlink in HTML?

```
A. <a href = "www.onlinedegree.iitm.ac.in"> IITM Online Degree Website </a>
B. <a url = "www.onlinedegree.iitm.ac.in" IITM Online Degree Website /a>
C. <a link = "www.onlinedegree.iitm.ac.in"> IITM Online Degree Website </a>
D. <a> www.onlinedegree.iitm.ac.in <IITM Online Degree Website /a>
```

Exercise: https://www.w3schools.com/tags/tag_a.asp

Week2 Live session

A CSS rule consists of a selector and a declaration block. CSS Syntax:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example: In this example all <p> elements will be center-aligned, with a red text color:

```
p {
  color: red;
  text-align: center;
}
```

Example Explained

- `p` is a selector in CSS (it points to the HTML element you want to style: <p>).
- `color` is a property, and `red` is the property value
- `text-align` is a property, and `center` is the property value

Exercise:https://www.w3schools.com/css/css_syntax.asp

Week2 Live session

Q4. Consider the following code segment.

```
<style>
        *{
            margin: 10px;
            border-radius: 10px;
        }
    </style>
```

What styling will be done by the above CSS in an HTML document?

A. A margin and border-radius of 10 pixels will be applied to all block tags.
B. A margin and border-radius of 10 pixels will be applied to all inline tags.
C. A margin and border-radius of 10 pixels will be applied to all tags having class = '*'.
D. A margin and border-radius of 10 pixels will be applied to all tags present in the body.

Week2 Live session

Q5. Consider the following code segment.

HTML code

```
<!DOCTYPE html>
<div { color: yellow; }
html lang="en">
<head>
    <title>Document</title>
    <link rel="stylesheet" href="style.css">
    <style>
        div{
            color:blue
        }
    </style>

</head>
<body>
    <div style="color: red;">IITM 1</div>
    <!-- <div>IITM 1</div>  -->
</body>
</html>
```

CSS code

```
div{color:yellow}
```

What will be the color of text IITM 1?

! Important>>Inline>> Internal>>External

Ans: It will take red color

Week2 Live session

6) Consider the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
    <style>
        #iitm{color: red}
        .iitm{color:green}
        div{color:yellow}
   </style>
</head>
</head>
<body>
   <div class = "iitm" id="iitm" >IITM Online</div>
</body>
</html>
```

What will be the color of text "IITM Online" if the above styling is used?

- ○ Black
- ○ Red
- ○ Green
- ○ Yellow

Ans: red color
Element- department, classes-class, student-id
Based on selector:
! important>>>id>>class>> element

Week2 Live session

`!important` > ID selector > Class selector > Element selector

Based on this hierarchy, here's the breakdown:

- **Element: Represents a basic HTML element selector like `div`, `p`, etc.**
- **Class: Denoted by `.classname`, targets elements with a specific class.**
- **ID: Denoted by `#idname`, has a higher specificity and targets a unique element.**
- **`!important`: Overrides any other styles, even if the specificity is lower.**

So if we assign terms to these:

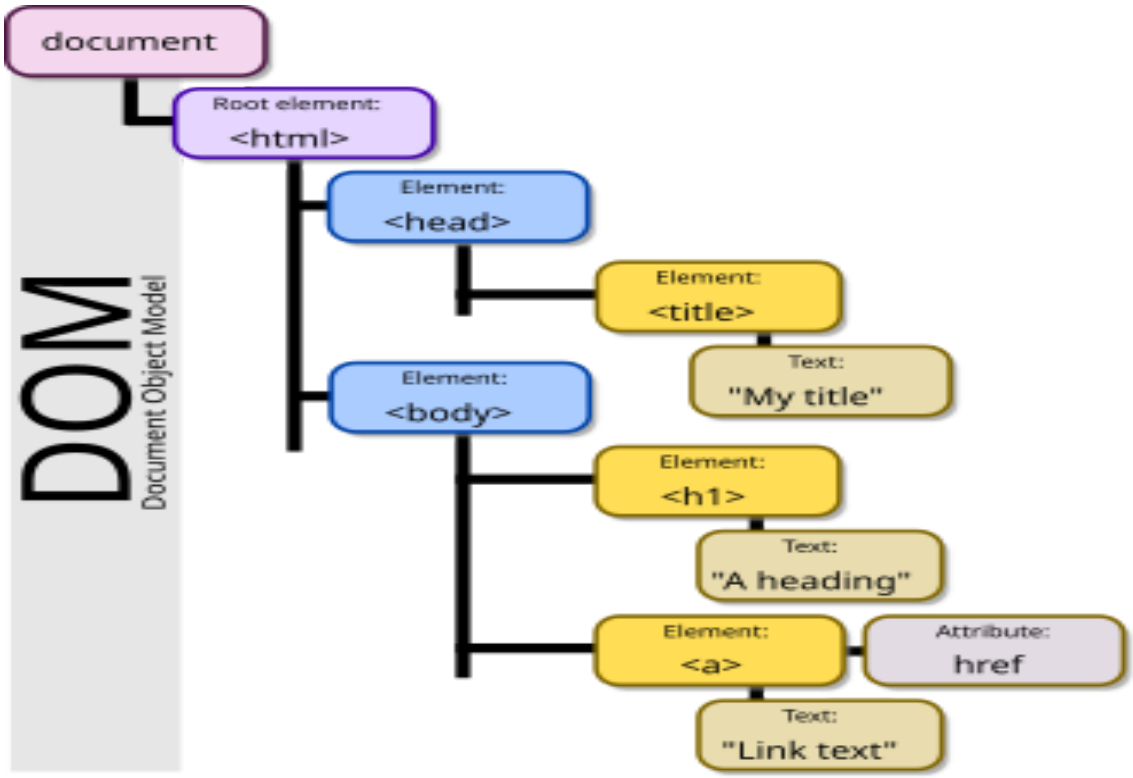
- **Element: "department" (basic structure, less specific).**
- **Class: "class" (more specific, applies to multiple elements).**
- **ID: "student" (most specific, applies to a unique element).**

In the provided code, the element `div` is assigned the ID `iitm` and the class `iitm`. Since the ID selector (`#iitm`) is more specific than the class selector (`.iitm`) or element selector (`div`), the color will be red.

So based on the CSS hierarchy:

- **Final Color: Red (due to the ID selector `#iitm` being more specific).**

Week2 Live session



[Document Object Model - Wikipedia]

Week2 Live session

Summary Link: Summary Mad1

To calculate the two's complement of a binary number (which is used to represent negative numbers in binary), follow these steps:

Steps to Calculate Two's Complement:

1. **Write the binary representation** of the positive number (in its unsigned binary form).
2. **Invert the digits** of the binary number (also called finding the one's complement). Change all `1`s to `0`s and all `0`s to `1`s.
3. **Add 1** to the inverted binary number (one's complement) to get the two's complement.

Example: Finding the Two's Complement of +6 (which is represented as `0110` in binary) to represent -6.

1. Start with the binary representation of 6:
   `0110`
2. Invert the digits (find one's complement):
   `1001`
3. Add 1 to the one's complement:
   `1001 + 1 = 1010`

Thus, `1010` is the two's complement representation of -6 in binary.

Key Points:

- The most significant bit (leftmost) in two's complement notation represents the sign of the number:
  - `0` for positive numbers.
  - `1` for negative numbers.
- Two's complement makes addition and subtraction easier because the same hardware can handle both signed and unsigned numbers without extra logic for negative numbers.

Week2 Live session

Signed and unsigned numbers refer to two ways of representing integers in binary, with the key difference being whether or not the number can represent negative values.

1. Unsigned Numbers

- **Definition**: Unsigned numbers can only represent non-negative integers (0 and positive numbers). All bits in an unsigned number represent the magnitude (size) of the number.
- **Range**: If you have an n-bit number, the range of an unsigned number is from 0 to 2^n - 1.
    - For example, with 4 bits, the unsigned range is 0 to 2^4 - 1 = 15, so the numbers representable are 0 to 15.

Example:

- **4-bit unsigned binary**:
    - `0000` = 0
    - `0110` = 6
    - `1111` = 15 (the maximum for 4 bits)

2. Signed Numbers

- **Definition**: Signed numbers can represent both positive and negative integers. One bit is reserved to indicate the sign (positive or negative), and the rest represent the magnitude of the number.
    - **Sign bit**: The leftmost (most significant) bit in a signed number is the **sign bit**.
        - `0` represents a positive number.
        - `1` represents a negative number.
    - **Two's complement** is the most common method used for representing signed numbers.
- **Range**: For an n-bit signed number, the range is from -2^(n-1) to 2^(n−1)−1.
- For example, with 4 bits, the signed range is from -2^3 = -8 to 2^3 - 1 = 7.

Example:

- **4-bit signed binary** (using two's complement):
    - `0000` = 0
    - `0110` = 6
    - `1111` = -1 (two's complement of `0001`)
    - `1001` = -7 (two's complement of `0111`)
    - `1000` = -8 (two's complement of `1000`)

Week2 Live session

Key Differences:

- **Unsigned numbers**: All bits are used for magnitude, and the number is always positive or zero.
- **Signed numbers**: One bit is used for the sign, allowing for both positive and negative values.

In both signed and unsigned numbers, the number of bits available limits the range of values that can be represented.

ASCII, UCS and UTF

1. **ASCII**: **American Standard Code for Information Interchange**
    - A 7-bit character encoding standard used to represent text in computers, communication equipment, and other devices.
2. **UCS**: **Universal Character Set**
    - A character encoding standard defined by the ISO/IEC 10646, capable of representing characters from all known written languages.
3. **UTF**: **Unicode Transformation Format**
    - A family of character encodings that can encode all possible characters (code points) in the Unicode standard. Examples include UTF-8, UTF-16, and UTF-32.

Each of these standards plays a significant role in text representation and communication between machines.

Bits and **bytes** are fundamental units used to measure data in computing, but they differ in size and their role in data storage and transmission.

1. Bits:

- **Definition**: A bit (short for **binary digit**) is the smallest unit of data in a computer. It can have one of two values: `0` or `1`.
- **Size**: A single bit represents one binary value (either `0` or `1`).
- **Usage**:
    - Bits are used in basic binary operations and represent the most fundamental form of data storage.
    - They are commonly used in networking to measure data transmission speed (e.g., Mbps = megabits per second).

2. Bytes:

- **Definition**: A byte is a group of **8 bits** and is the basic unit used to represent a character or store larger amounts of data.

Week2 Live session

- **Size**: 1 byte = 8 bits.
- **Usage**:
    - Bytes are used to represent characters (like letters, numbers, symbols) in character encoding schemes (e.g., ASCII, UTF-8).
    - They are also used to measure the size of files and storage capacity (e.g., kilobytes, megabytes, gigabytes).

Key Differences:

Bits	Bytes
Smallest unit of data (1 or 0)	Larger unit, composed of 8 bits
Used to measure data transfer speeds	Used to measure data storage or memory
Symbol: b (e.g., Mbps)	Symbol: B (e.g., MB)
Represents binary values	Represents characters or larger data

Example:

- **1 byte** can store a single character (like 'A').
- **1 bit** can store a single binary digit (0 or 1). It takes 8 bits (1 byte) to represent most characters in standard encoding systems like ASCII.