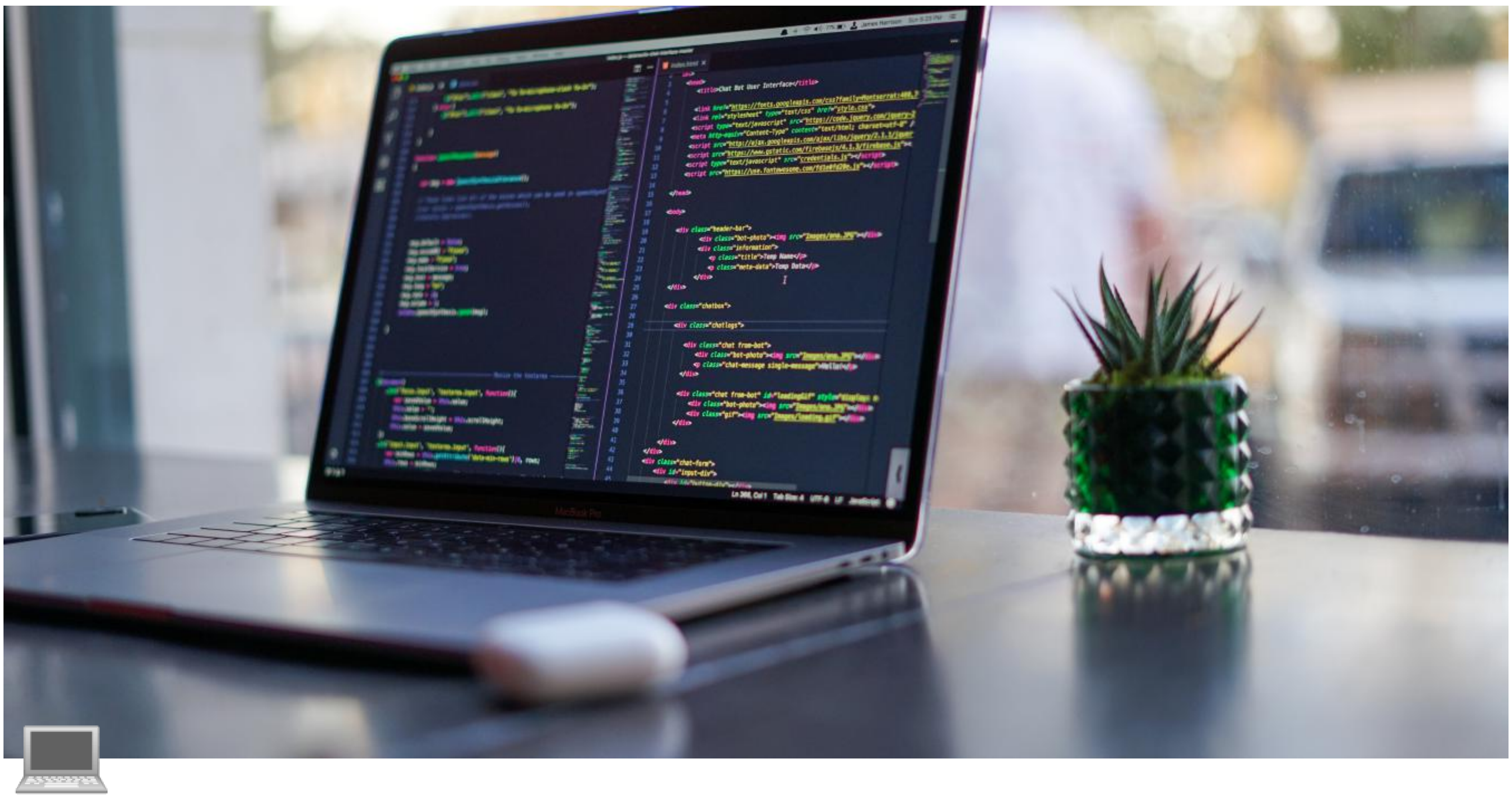


Week 2 Lecture 1

Class	BSCCS2003
Created	@September 14, 2021 12:16 PM
Materials	
Module #	10
Type	Lecture
Week #	2

Information Representation in a Machine

Markup

- Information representation
- Raw data vs Semantics
- Logical structure vs Styling
- HTML5 & CSS

Information representation

- Computers work only with "bits"
  - Binary digits: 0 and 1
- Numbers
  - Place value: binary numbers: eg. 6 = 0110
  - Two's complement: negative numbers: eg. -6 = 1010

Representing Text

- ASCII
- Unicode
- UTF-8

Information Interchange

- Communicate through machines - either between machines or between humans
- Machines only work with ***bits***
- Standard "encoding"
  - Some sequence of bits interpreted as a character

Interpretation

What is "0100 0001"?

- It is a string of bits
- Number with a value of 65 in decimal
- Character "A"
- All of the above statements are correct

It boils down to a matter of **interpretation** and **context**

ASCII

American Standard Code for Information Interchange

- 7-bits: 128 different entities
  - 'a'..'z'
  - 'A'..'Z'
  - '0'..'9'
  - Special characters- ! @ # $ % ^ & * ( ) ...
- Why 7-bits?

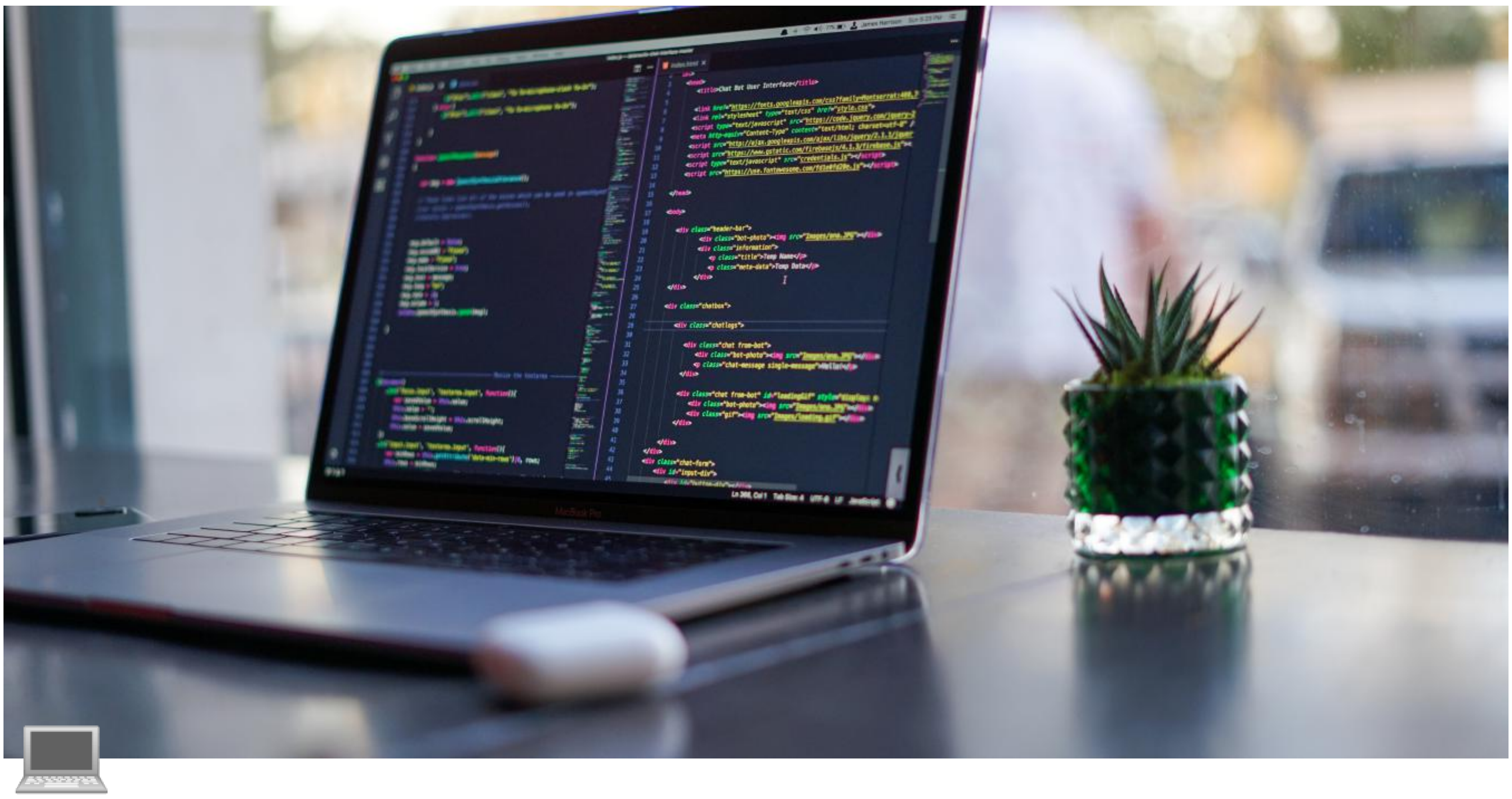
  It was a time before even 8-bit computers were standardised

  The number of bits were precious and you really didn't want to use up more than necessary bits

  That is why people tried to optimise the number of bits
- What about other characters? あ आ 아 ஐ
  - 1000s of characters needed

Unicode

- Allow codes for more script, characters
- How many?
  - All the living languages?
  - All extinct languages?
  - All future languages?
- **"Universal Character Set"** encoding - UCS
  - **UCS-2** 2 bytes per character - max 65,536 characters
  - **UCS-4** 4 bytes per character - 4B+ characters



Week 2 Lecture 2

Class	BSCCS2003
Created	@September 14, 2021 4:50 PM
Materials	
Module #	11
Type	Lecture
Week #	2

Efficiency of encoding

Efficiency

- What is the most common language on the web?
  - As of now, it is English
- Should all the characters be represented with the same number of bits?
- Example:
  - A text document with 1,000 words, which equates to approximately 5,000 characters (including spaces)
  - According to UCS-4 encoding

    32 bits × 5,000 = 160,000 bits
  - According to ASCII encoding

    8 bits × 5000 = 40,000 bits
  - According to the original 7-bit ASCII which was sufficient for English

    7 bits × 5000 = 35,000 bits
  - Minimum needed to encode just 'a' - 'z', numbers and some special characters

    could fit in 6-bits: 30,000 bits
  - Optimal coding based on frequency of occurrence of a letter
    - 'e' is the most common letter, 't', 'a', 'o', ...
    - Huffman or similar encoding: ~10,000 - 20,000 bits, possibly less

Is this problem solvable in general?

- It is impossible to encode by actual character frequency: depends on the text
  - Just use compression methods like "zip" instead
- But can encoding be a good halfway point?

Example:

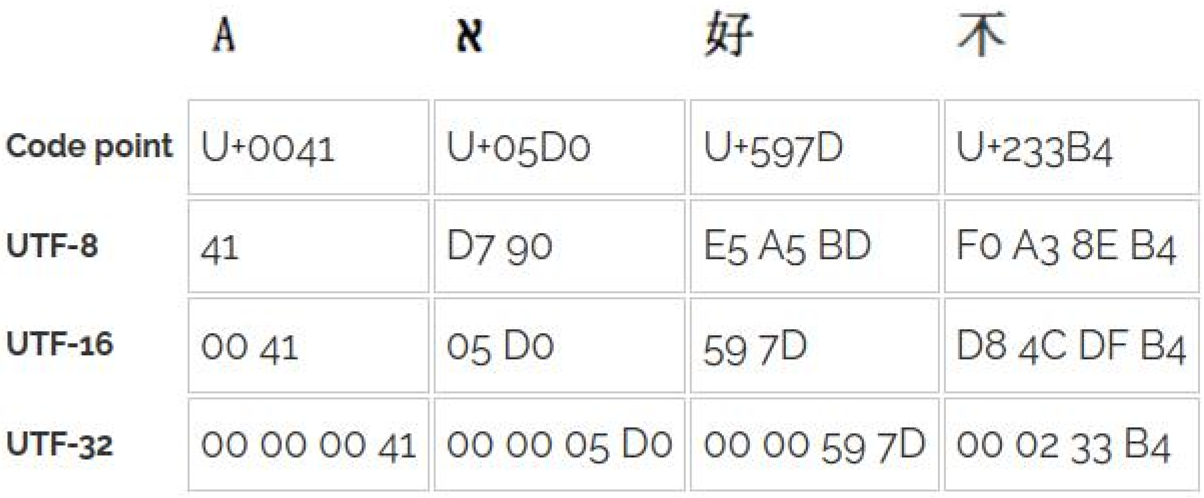
- Use 1-byte for most common alphabets
- Group others according to frequency, have "prefix" codes to indicate

Prefix Coding

1st byte	2nd byte	3rd byte	4th byte	Free bits	Max. expressible Unicode val.
0xxxxxxx				7	007F hex (127)
110xxxxx	10xxxxxx			(5+6)=11	07FF hex (2047)
1110xxxx	10xxxxxx	10xxxxxx		(4+6+6)=16	FFFF hex (65535)
11110xxx	10xxxxxx	10xxxxxx	10xxxxxx	(3+6+6+6)=21	10FFF hex (1,114,111)

- If the first bit starts with a **0**, take the remaining 7 free bits as an ASCII code

Example

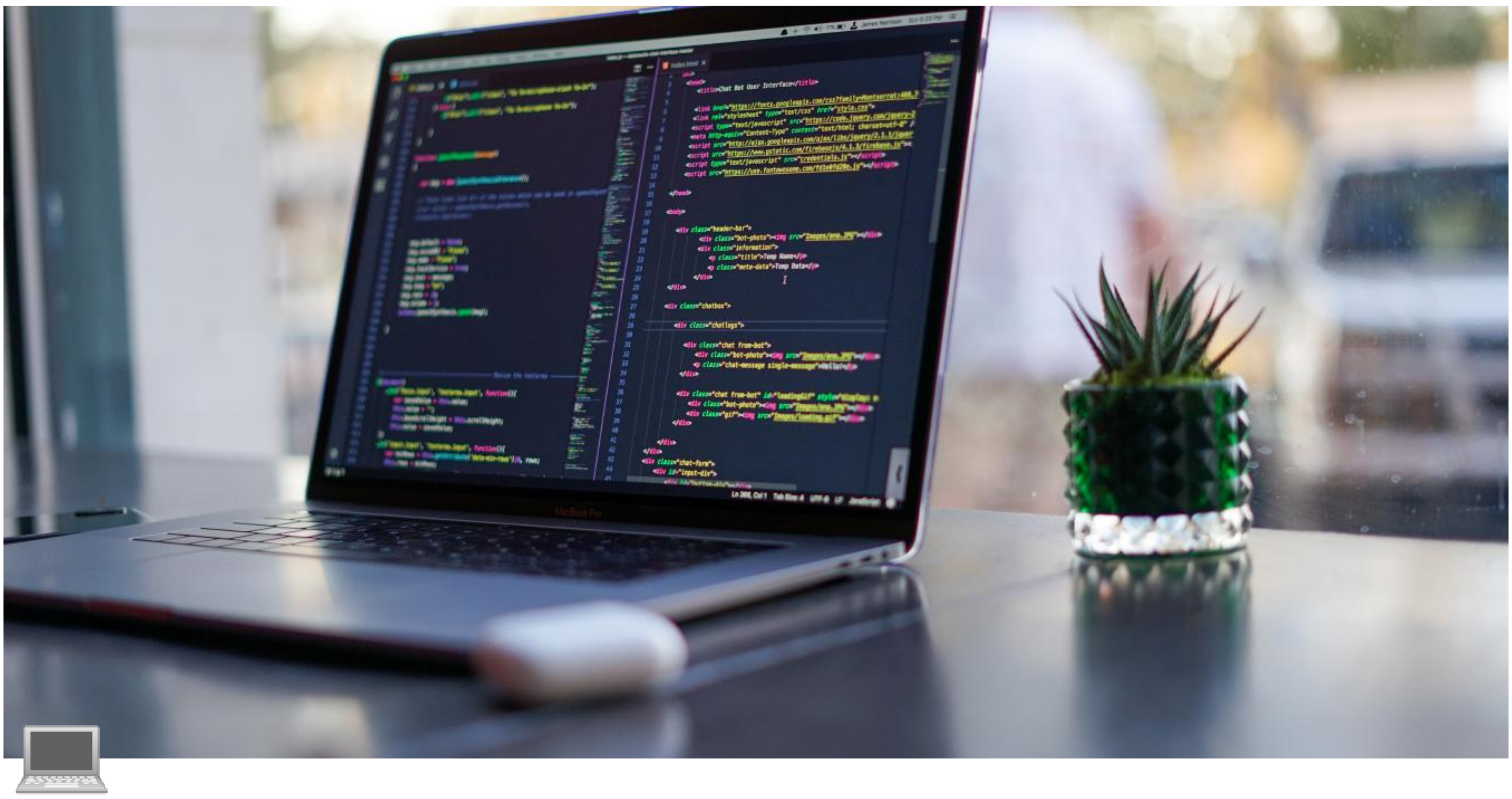


	A	א	好	𣎴
Code point	U+0041	U+05D0	U+597D	U+233B4
UTF-8	41	D7 90	E5 A5 BD	F0 A3 8E B4
UTF-16	00 41	05 D0	59 7D	D8 4C DF B4
UTF-32	00 00 00 41	00 00 05 D0	00 00 59 7D	00 02 33 B4

Source: https://www.w3.org/International/articles/definitions-characters

UTF-8

- Stands for Unicode Transformation Format
- Uses 8-bits for most common characters: ASCII subset
  - All ASCII documents are automatically UTF-8 compatible
- All other characters can be encoded based on prefix string
- More difficult for text processors:
  - first check prefix
  - linked list through chain of prefixes possible
  - Still more efficient for majority of documents
- Most common encoding in use today



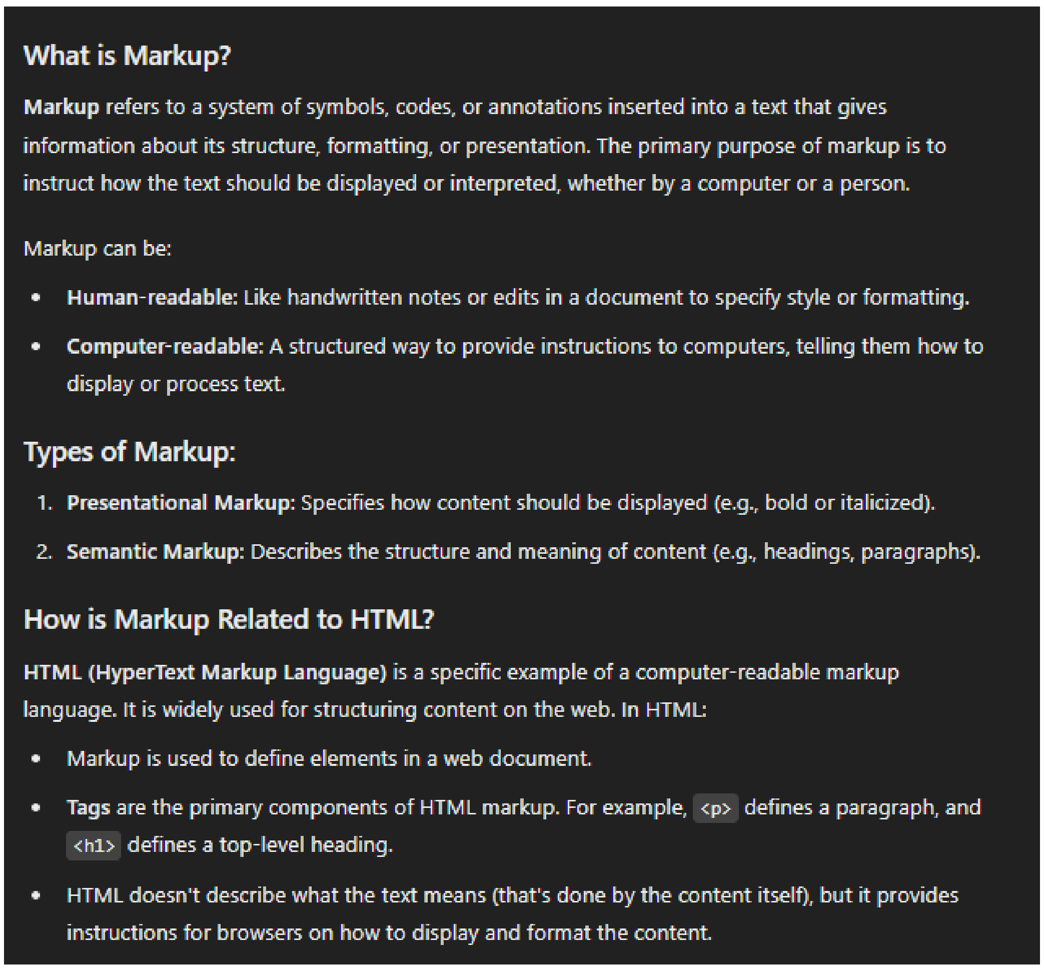
Week 2 Lecture 3

Class	BSCCS2003
Created	@September 14, 2021 6:39 PM
Materials	
Module #	12
Type	Lecture
Week #	2

What is Markup?

Topics

- Content v/s Meaning
- Types of Markup
- (X)HTML



What is Markup?

Markup refers to a system of symbols, codes, or annotations inserted into a text that gives information about its structure, formatting, or presentation. The primary purpose of markup is to instruct how the text should be displayed or interpreted, whether by a computer or a person.

Markup can be:

- **Human-readable:** Like handwritten notes or edits in a document to specify style or formatting.
- **Computer-readable:** A structured way to provide instructions to computers, telling them how to display or process text.

Types of Markup:

1. **Presentational Markup:** Specifies how content should be displayed (e.g., bold or italicized).
2. **Semantic Markup:** Describes the structure and meaning of content (e.g., headings, paragraphs).

How is Markup Related to HTML?

HTML (HyperText Markup Language) is a specific example of a computer-readable markup language. It is widely used for structuring content on the web. In HTML:

- Markup is used to define elements in a web document.
- **Tags** are the primary components of HTML markup. For example, `<p>` defines a paragraph, and `<h1>` defines a top-level heading.
- HTML doesn't describe what the text means (that's done by the content itself), but it provides instructions for browsers on how to display and format the content.

Raw content

```
Markup What is markup? Markup is a way of using cues or codes in the regular flow to indicate how text should be displayed. Ma
rkup is very useful to make the display of text clear and easy to understand
```

Types of Markup

- Presentational
    - **WYSIWYG:** directly format output and display
        - What You See Is What You Get
    - Embed codes not part of regular text, specific to the editor
- Procedural
    - Details on how to display something
        - Change the font to large, bold
        - skip 2 lines, indent 4 columns

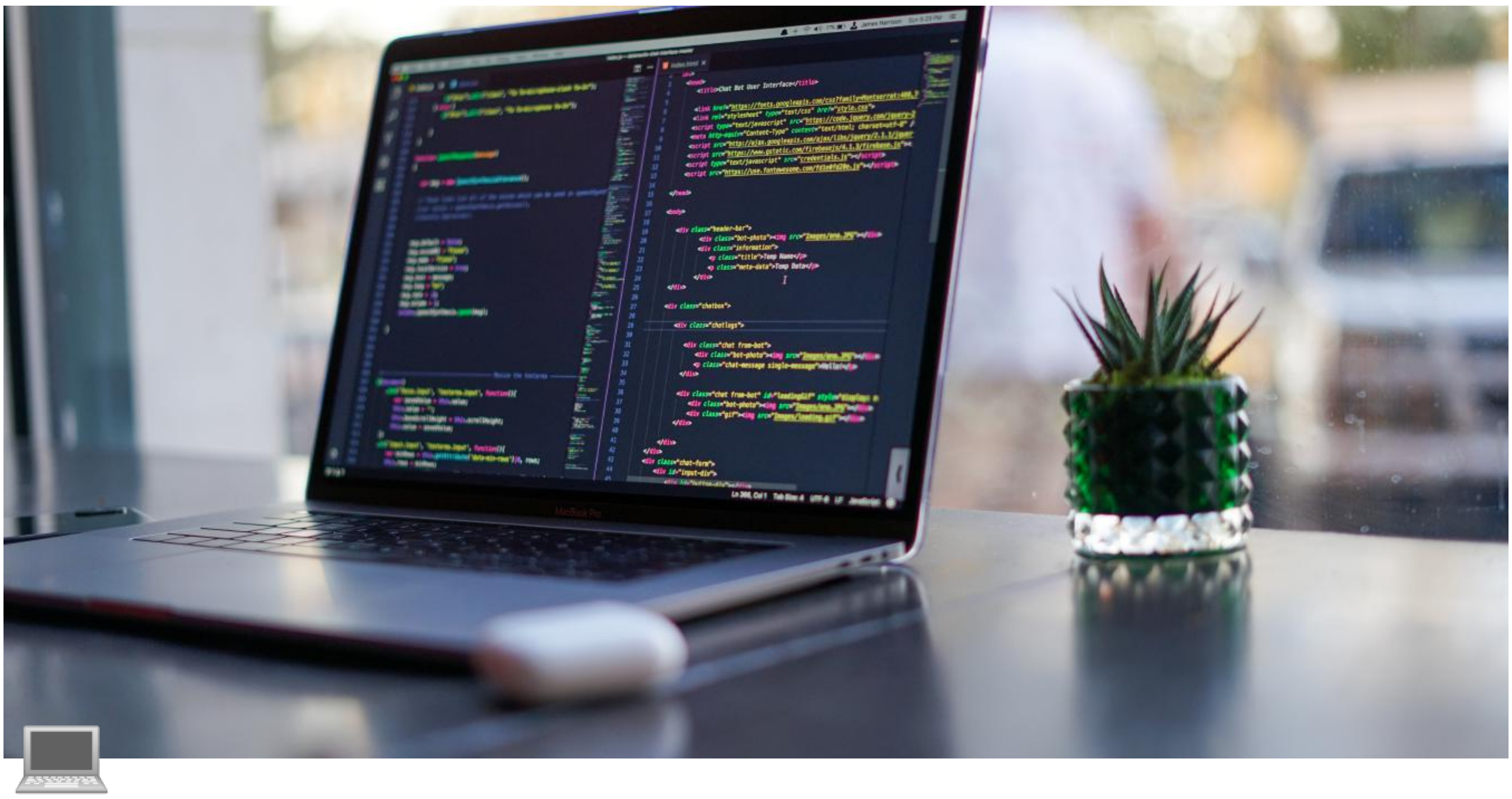
- Descriptive
  - This is a `<title>`, this is a `<heading>`, this is a `<paragraph>`

Examples

- Microsoft Word, Google Docs, etc
  - User interface focused on "appearance", not meaning
  - WYSIWYG: direct control over styling
  - Often leads to complex formatting and loss of inherent meaning
- LaTeX, HTML (general *ML)
  - Focus on meaning
  - More complex to write and edit, not WYSIWYG in general
  - Fun fact: The current raw document was written in LaTeX and rendered to a PDF later

Semantic Markup

- Content v/s Presentation
- Semantics
  - Meaning of the text
  - Structure or Logic of the document



💻

Week 2 Lecture 4

Class	BSCCS2003
Created	@September 14, 2021 7:21 PM
Materials	
Module #	13
Type	Lecture
Week #	2

Introduction to HTML

Topics

- HyperText Markup Language
- Generalizations
- Variants of Interest

HyperText Markup Language (HTML)

- HTML was first used by Tim Berners-Lee in the original Web at CERN (-1989)
- Considered an application of SGML (Standard Generalized Markup Language)
    - Strict definitions on structure, syntax and validity
- HTML was meant for browser interpretation
    - Very forgiving: loose validity checks
    - Best effort to display

HTML Example

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Heading.</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Tags

- `<h1></h1>` - paired tags
- Represented by angled brackets `<>`
- Closing tag with `/`
- Location specific: `<DOCTYPE>` is only at the head of the document
- Case insensitive

Nesting

Valid syntax

- `<em><strong>Hello</strong></em>`
- The output is
    - ***Hello***
- Invalid:
    - `<em><strong>Hello</em></strong>`
    - `<em><strong>Hello</em>`
    - `<em><strong>Hell<o/em></strong>`

Presentation v/s Semantics

- `<strong>Hello</strong>`
- `<b>Hello</b>`
- **Hello**

Which one is right? Which one is better?

It is not an easy question to answer, both of them may be correct

`<strong>` is putting an emphasis meanwhile `<b>` does not

Timelines

- SGML based
    - 1989 - HTML original
    - 1995 - HTML 2
    - 1997 - HTML 3, 4
- XML based
    - XHTML - 1997 - mid 2010s
- HTML 5
    - first release in 2008
    - W3C recommendation - 2014
        - World Wide Web Consortium

HTML 5

- Block element: `<div>`
- In-line element: `<span>`
- Logical elements: `<nav>`, `<footer>`
- Media: `<audio>`, `<video>`

Remove "presentation only" tags:

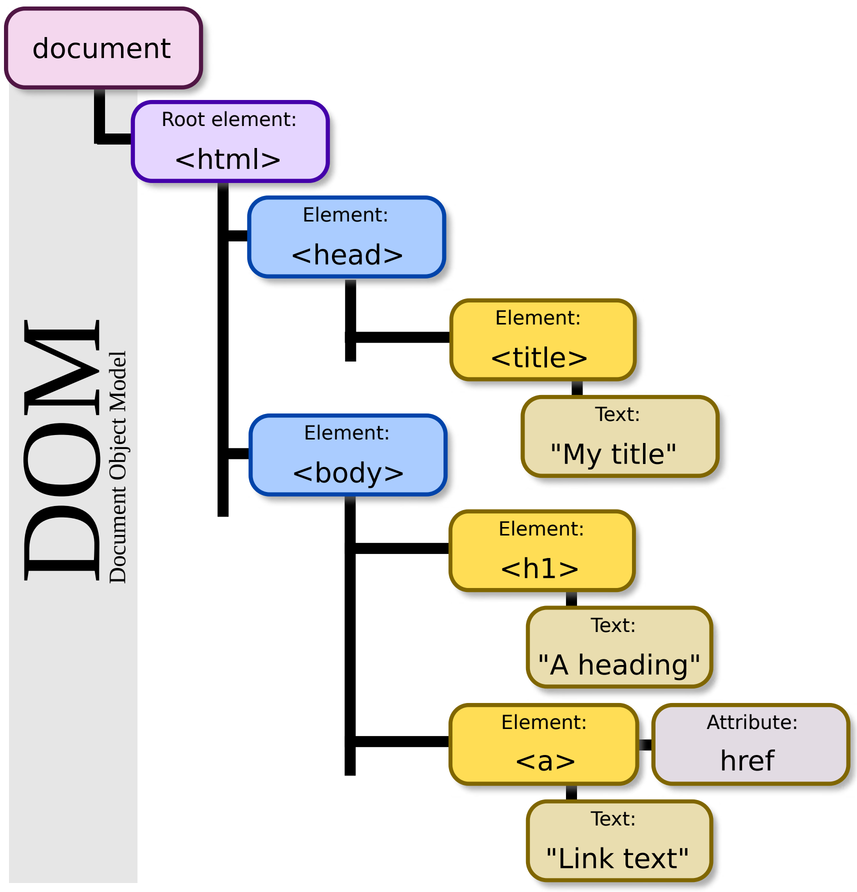
- `<center>`
- `<font>`

Document Object Model (DOM)

> The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page.

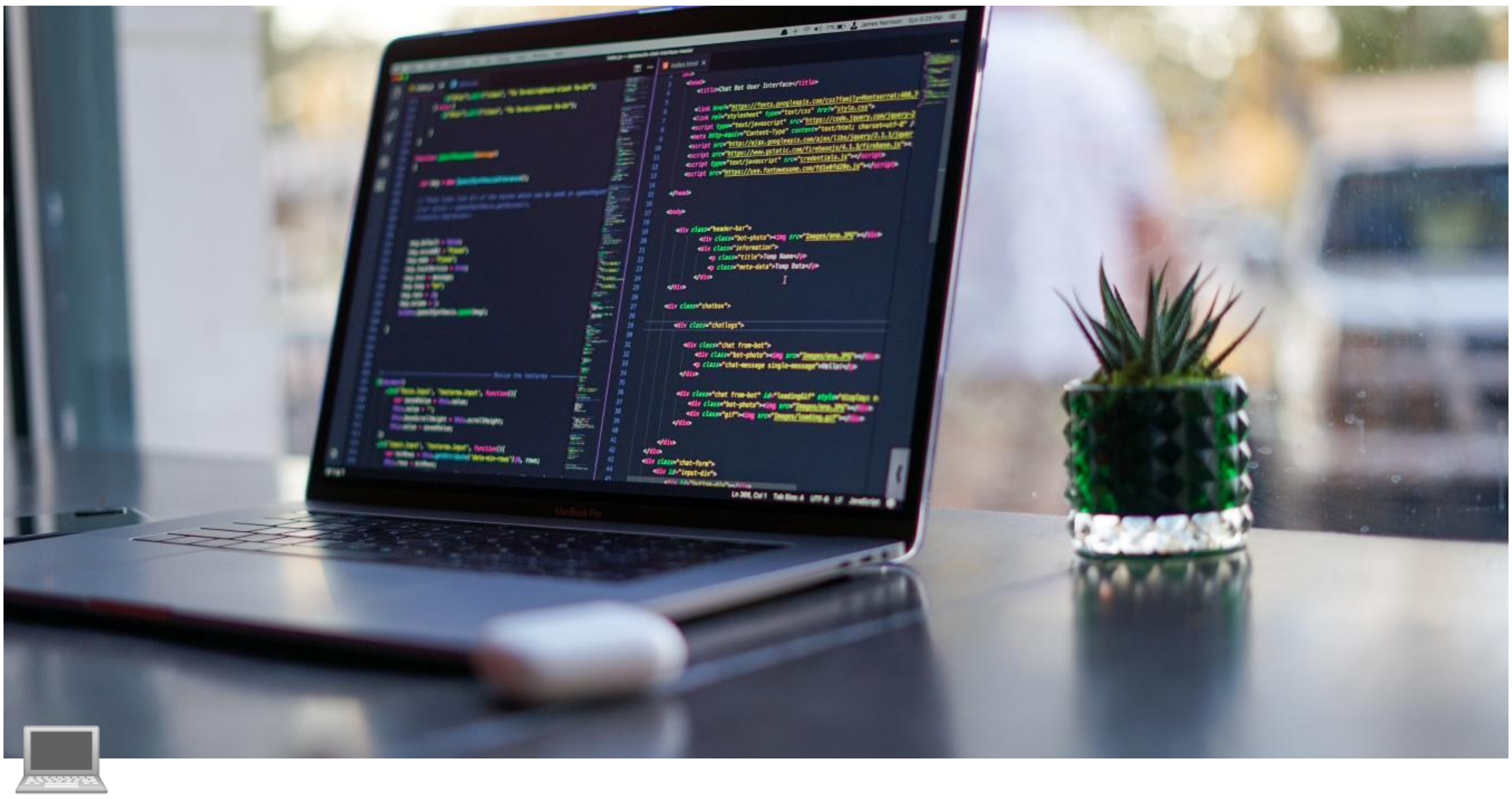
Source: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

```
<html>
<head>
  <title>My title</title>
</head>
<body>
  <h1>A heading</h1>
  <a href="https://onlinedegree.iitm.ac.in">Link text</a>
</body>
</html>
```



Source: https://en.wikipedia.org/wiki/Document_Object_Model

- It is a tree structure which represents the logical layout of the document
- Direct manipulation of this tree is possible!
- Application Programming Interfaces (APIs) are also defined on the DOM:
  - Canvas
  - Offline
  - Web Storage
  - Drag & Drop
  - ...
- In most of the cases, when we talk about manipulating the DOM, we use JavaScript
- Cascading Style Sheets (CSS) is used for styling



Week 2 Lecture 5

Class	BSCCS2003
Created	@September 15, 2021 12:59 PM
Materials	
Module #	14
Type	Lecture
Week #	2

Introduction to Styling

Topics

- Markup v/s Style
- Themes
- CSS

Markup v/s Style

```
<h1>Hello</h1>
```

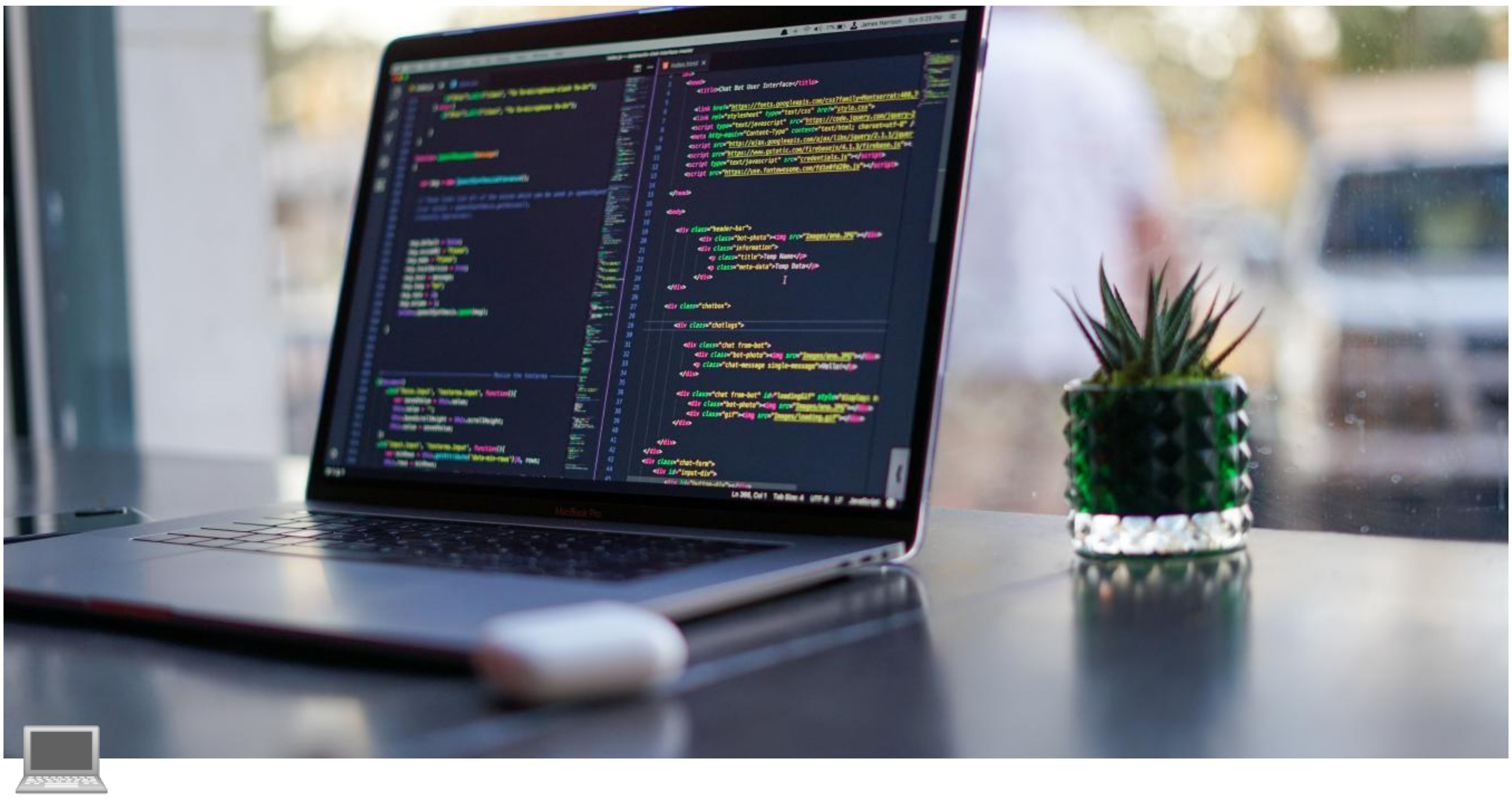
https://codepen.io/21f1003586/pen/bGRoJJX

https://codepen.io/21f1003586/pen/BaZweBX

Separation of Styling

- Style hints in separate blocks
  - Separate files included
- Themes

- Style sheets
    - Specify presentation information
- Cascading Style Sheets (CSS)
    - Allows multiple definition
    - Latest definition takes the precedence



Week 2 Lecture 6

Class	BSCCS2003
Created	@September 15, 2021 2:23 PM
Materials	
Module #	15
Type	Lecture
Week #	2

Types of CSS styling & Responsive Websites

Inline CSS

- Directly add style to the tag
- Example:

```
<h1 style="color: blue; text-align: center;">A heading</h1>
```

Internal CSS

- Embed inside the `<head>` tag
- Now, all the `<h1>` tags in the document will look the same - centrally modified

```
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
```

External CSS

- Extract common content for re-use

- Multiple CSS files can be included
- Latest definition of style takes the precedence

Responsive Design

- Mobile and Tablet devices have smaller screen
  - Different form factors
- Adapt to the screen - ***Respond***
- CSS controls the styling - HTML controls the client

Bootstrap

- Most commonly used framework for CSS
  - Originate from Twitter
  - Widely used now
- Standard styles for various components
  - Buttons
  - Forms
  - Icons
  - ...
- Mobile first: highly responsive layout

JavaScript

- It is an interpreted language brought into the browser
- Not really related to Java in any way - formally known as ECMAScript
- Why was JavaScript brought into the browser?
  - HTML is not a programming language
  - CSS is not a programming language too (There are ways to make it Turing complete, google it)
- We wanted to have "programmability" inside the browser
- It is not a part of the core presentation requirements
  - But still, it is very useful and will be considered later

Summary

- Presentation - Human interaction
- Separate content from style
  - Markup - HTML
  - Styling - CSS