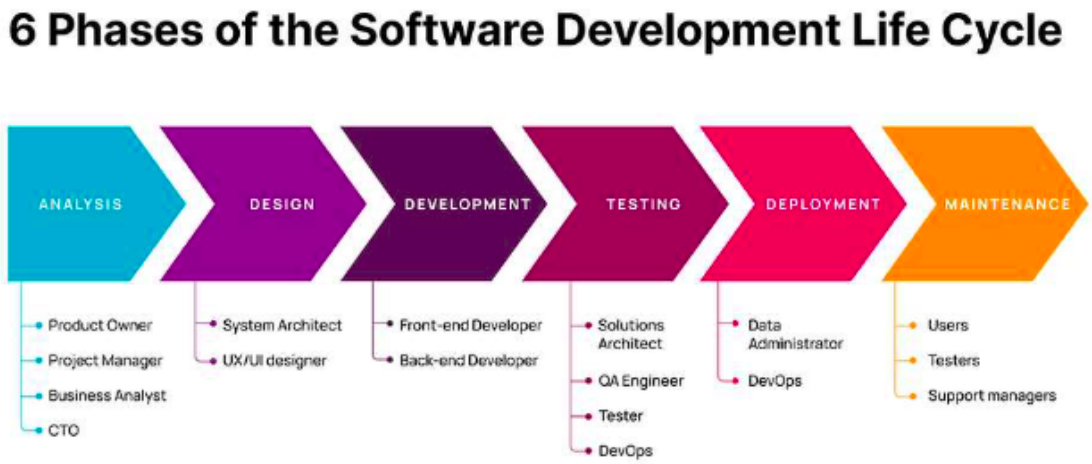
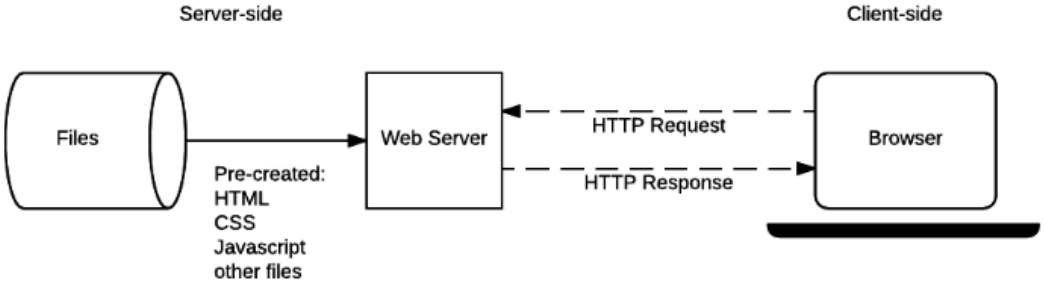
UNIT - 1

Introduction

- SDLC
    - analysis
    - design
    - dev
    - testing
    - deployment
    - maintenance



- introduction to web tech
    - web
        - collection of machines connected via the Internet using the HTTP application layer protocol to communicate via web pages
    - web client
        - machine requesting info
    - web browser
        - end user software that serves as an interface for the user to access web pages(renders web pages)
    - web server
        - machines that host/store and serve web pages to clients



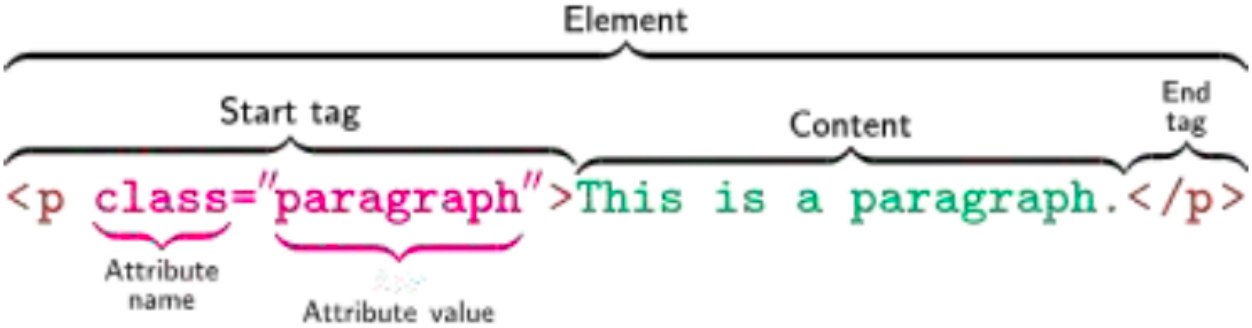
    - revisions of the web

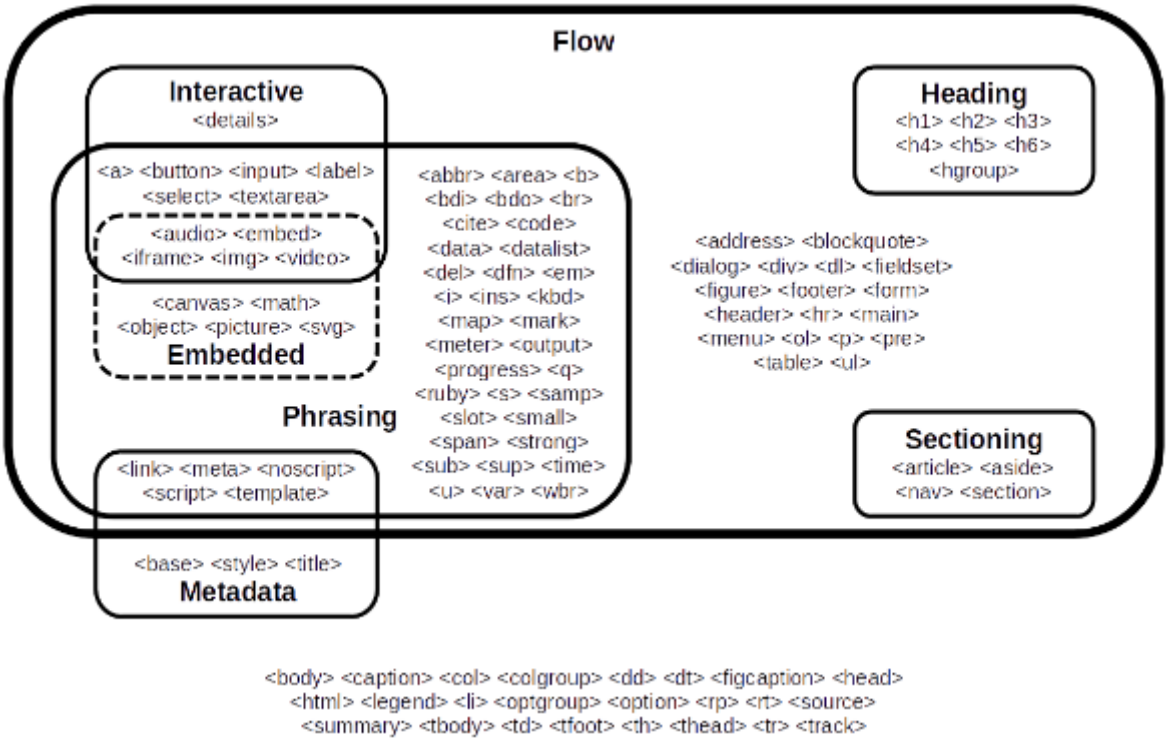
        - 1.0 - read only
        - 2.0 - social - read/write; user-generated content via platforms like YouTube, Twitter
        - 3.0 - semantic - read/write/exec; decentralized via blockchain, removing reliance on central authorities
        - 4.0 - mobile ; AI and IoT integration; hyper personlized
        - 5.0 - intelligent/emotional symbiotic; web interacts with human emotions; neural connections to the web
    - technology vs engineering
        - tech - focuses more on application and adapts to cahnges in teh industry
        - eng - focus more on design, analysis and evaluation
    - about developments
        - client side prog
            - HTML(HTML5) - content + structure
            - CSS(CSS3) - presentation; advantageous as can swap styles
            - JS - dynamic web pages; make use of events
        - server side prog
        - web services

HTML

- hypertext markup language
    - hypertext
        - highlighted links
        - easy navigation within or across webpages
    - markup
        - mark text(bold, italic, underline, etc.)
    - lang
        - respresnted by elements(syntax + nodes)

- elements





- tags
    - keywords enclosed in angular brackets
    - 142 tags in HTML 5.2; 115 compatible amongst all versions
    - syntax error ifthey are not closed
    - not case sensitive
- tag attributes
    - keywords present in tags
    - specific set of vals
    - additional characteristics to elements
    - not mandatory for most tags
    - found in the start tag of an element
- editing
    - HTML files edited using a text editor
    - file extension - .html or .htm
    - file name to be names based on functionality(subjected to variability based on use case); home page - index.html
    - HTML errors are not fatal and do not crash the prog
- !DOCTYPE
    - indicates to the browser that it is an HTML file
    - syntax may vary depending on the version
    - empty element - no end tag and no content
- `<html>`
    - root element

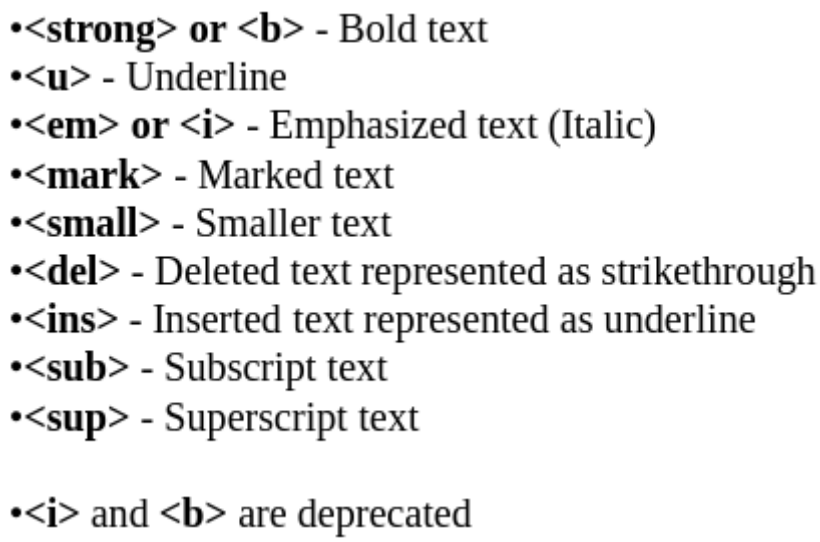
    - contains all elements except !DOCTYPE

- `<head>`
    - between `<html>` and `<body>`
    - contains metadata
    - info generally not rendered
    - metadata tags - `<title>`, `<style>`, `<meta>`, `<link>`, `<script>`, `<base>`

- `<title>`
    - sets the title of the webpage
    - one title per webpage

- `<body>`, `<p>`
    - `<body>` - document content; text, paragraph, image, hyperlinks, tables, lists, frames
    - `<!-- comments -->` - adding comments to your file
    - `<p>` - paragraph; brwoser renders new line before and after the element

- text formatting



- headers
    - defautl text foramtting
    - `<h1>` to `<h6>` - decides size

- hyperlinks
    - `<a>` anchor
    - mandatory attribute - href
    - `<a href="address">content</a>`
    - all links - underlined
    - visited - purple
    - unvisited - blue
    - active - red

- inernal linking

    - linking to a loaction within a webpage
    - set location to go to - `<a name="value">content</a>`
    - as usual - `<a href="#value">content</a>`

- images, special char, `<hr>`, `<br>`

    - `<img>` - manfatory attribute is src
    - special chars like math chars can be added in code form
    - `<br>` - line break; no closing tag
    - `<hr>` - horizontal line/rule; no closing tag

- lists

    - list - `<li>` - closing tag optional
    - unordered list - `<ul>` - bullets; can be nested
    - ordered list - `<ol>` - numbers; can be nested
    - newline after every closed list

- table

    - `<table>` - table element
        - `<caption>` - caption element
        - `<tr>` - row element
            - `<td>` - data element
        - `<colgroup>` - column group element - styling group of colums
            - `<col>` - column to be styled
        - `<thead>` - table header element
            - `<th>` - table head element
        - `<tbody>` - table body element
            - `<td>` - data element
        - `<tfoot>` - table foot element
            - `<td>` - data element

- iframes

    - group multiple HTML files
    - `<iframe>` - inline frame
    - styling - attrbutes/CSS files
    - eg.: `<iframe src="HTML file" title="Title for the HTML file"></iframe>`

CSS

- cascading style sheets
    - cascade - change one style to another
- 3 ways of specification
    - inline - HTMl tag attributes

    - internal - HTML `<style>` element in `<head>` element

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
```

- Whole – selector string
- body – selector
- background-color – property names
- background-color: powderblue; - declaration
- {} – declaration block

    - external - separate CSS file
        - add using `<link>` element - rel and href are mandatory attributes; defined in `<head>` element

```
<head>
  <link rel="stylesheet" href="sample.css">
</head>
```

- if multiple stylesheets are used, the latest style will be applied
    - styles are applied in the order of inclusion
    - more specific styles override the other specifications
    - inline and internal styles override external styles
- selector strings
    - single element type
    - multiple element types - comma separate list of element names
    - all element types - *
    - elements by ID - #id_name
    - single element type by class name - .class_name
    - specific element + class name - element_name.class_name
    - comments - /**/
- more than 200 property names

> browsers render a new line before and after a `<div>` element

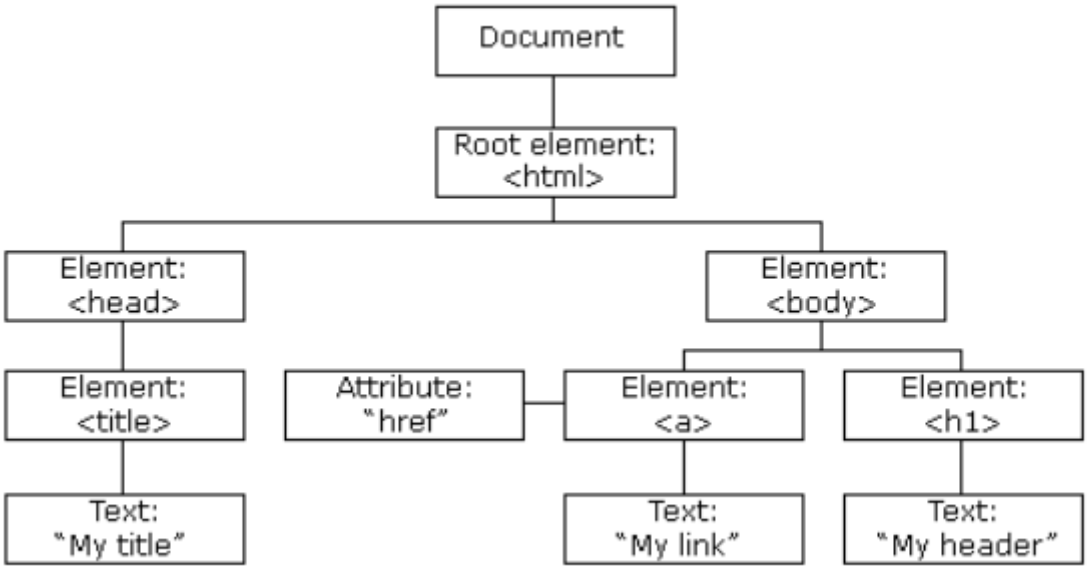
Javascript

- introduction
    - most popular lang
    - OOPS - inspired from Java
    - Netscape browser supported Java Applets; used to develop web pages in Java
    - high level language
    - follows ECMA(European Computer Manufacturer's Association) script standard
    - dynamically typed
    - interpreted/Just-In-Time compiled
- types of progamming with respect to web pages
    - inline - write the script inn the `<script>` tag
    - external - include script(saved as .js file) path in the src attribute of the `<script>` element

- 48 keywords

abstract	arguments	await*	boolean
break	byte	case	catch
char	class*	const*	continue
debugger	default	delete	do
double	else	enum*	eval
export*	extends*	false	final
finally	float	for	function
goto	if	implements	import*
in	instanceof	int	interface
let*	long	native	new
null	package	private	protected
public	return	short	static
super*	switch	synchronized	this
throw	throws	transient	true
try	typeof	var	void
volatile	while	with	yield

- DOM
- document object model



- standard for accessing documents
- parts
  - core DOM
  - XML DOM
  - HTML DOM
- HTML DOM is used to get, change, add or delete HTML elements
  - dynamic HTML

        - add new HTMl elements, attributes and CSS styles
        - change HTMl elements, attributes and CSS styles
        - remove existing HTML elements, attributes and CSS styles
        - react to HTML events
        - create new HTMl events
        - remove existing HTMl events

- HTML objects have properties and methods
- events for HTML
- finding HTML elements

Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

- changing HTML elements

Property	Description
element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.style.*property = new style*	Change the style of an HTML element
Method	**Description**
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

- DOM traversal
    - directions
        - upward
            - parentElement
            - parentNode
        - downward
            - firstElementChild
            - children
            - lastElementChild
            - childNodes, firstChild, lastChild
        - sideways
            - nextElementSibling, previousElementSibling
            - nextsibling, previousSibling
    - document.querySelector() selects the first element/attribute
    - document.querySelectorAll() selects all elements/attributes, returning a NodeList

- adding/deleting elements

Method	Description
document.createElement(*element*)	Create an HTML element
document.removeChild(*element*)	Remove an HTML element
document.appendChild(*element*)	Add an HTML element
document.replaceChild(*new*, *old*)	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

- HTML event handling

Method	Description
document.getElementById(*id*).onclick = function(){*code*}	Adding event handler code to an onclick event

- finding HTML objs

Property	Description	DOM
document.anchors	Returns all <a> elements that have a name attribute	1
document.applets	Deprecated	1
document.baseURI	Returns the absolute base URI of the document	3
document.body	Returns the <body> element	1
document.cookie	Returns the document's cookie	1
document.doctype	Returns the document's doctype	3
document.documentElement	Returns the <html> element	3
document.documentMode	Returns the mode used by the browser	3
document.documentURI	Returns the URI of the document	3
document.domain	Returns the domain name of the document server	1
document.domConfig	Obsolete.	3
document.embeds	Returns all <embed> elements	3
document.forms	Returns all <form> elements	1
document.head	Returns the <head> element	3
document.images	Returns all <img> elements	1
document.implementation	Returns the DOM implementation	3
document.inputEncoding	Returns the document's encoding (character set)	3
document.lastModified	Returns the date and time the document was updated	3
document.links	Returns all <area> and <a> elements that have a href attribute	1
document.readyState	Returns the (loading) status of the document	3
document.referrer	Returns the URI of the referrer (the linking document)	1
document.scripts	Returns all <script> elements	3
document.strictErrorChecking	Returns if error checking is enforced	3
document.title	Returns the <title> element	1
document.URL	Returns the complete URL of the document	1

- HTML events
    - event listener
        - some events such as mouse click, scrolling, key press, loading, animations, etc.
    - event handler
        - JS - repsonds to an event
    - HTML - events can be given element attributes
    - JS - evenets can be added to element objs
        - element.addEventListener(event, func());
        - element.removeEventListener(event, func());

AJAX

- asynchronous JS and XML
- collection of web dev techniques to build more responsive web apps
- async nature
    - send request to server; proceeds to next request before response; response handled in the background
- features
    - update pages without reloading
    - request/respond data after page loads
    - sending data to server in the background
- utilities
    - browser's built-in XMLHttpRequest obj - send request to server
        - identifier_name = newXMLHttpRequest();
    - HTML, DOM, JS - used to display data

new XMLHttpRequest()	Creates a new XMLHttpRequest object
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(*method,url,async,user,psw*)	Specifies the request *method*: the request type GET or POST *url*: the file location *async*: true (asynchronous) or false (synchronous) *user*: optional user name *psw*: optional password
send()	Sends the request to the server Used for GET requests
send(*string*)	Sends the request to the server. Used for POST requests
setRequestHeader()	Adds a label/value pair to the header to be sent

onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

Reference Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Common HTML Elements</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

    <header>
        <h1>Welcome to My Page</h1>
        <p>A simple HTML page with commonly used elements.</p>
    </header>

    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </nav>

    <section>
        <h2>About Us</h2>
        <p>This is a paragraph inside a section. <strong>Bold text</strong>
and <em>italic text</em> are commonly used.</p>
        <img src="https://via.placeholder.com/400" alt="Sample Image">
    </section>
```

```
    <section>
        <h2>Our Services</h2>
        <ul>
            <li>Web Design</li>
            <li>Development</li>
            <li>SEO Optimization</li>
        </ul>
    </section>

    <section>
        <h2>Contact Us</h2>
        <form>
            <label for="name">Name:</label>
            <input type="text" id="name" placeholder="Enter your name"
required>

            <label for="email">Email:</label>
            <input type="email" id="email" placeholder="Enter your email"
required>

            <label for="message">Message:</label>
            <textarea id="message" placeholder="Your message"></textarea>

            <button type="submit">Send</button>
        </form>
    </section>

    <section>
        <h2>Data Table</h2>
        <table>
            <tr>
                <th>Name</th>
                <th>Age</th>
                <th>Job</th>
            </tr>
            <tr>
                <td>John Doe</td>
                <td>30</td>
                <td>Developer</td>
            </tr>
            <tr>
                <td>Jane Smith</td>
                <td>28</td>
                <td>Designer</td>
            </tr>
        </table>
    </section>

    <footer>
        <p>&copy; 2025 My Website | All Rights Reserved</p>
    </footer>
```

```
</body>
</html>
```

```
/* General Styles */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    line-height: 1.6;
    background-color: #f4f4f4;
}

/* Header */
header {
    background: #333;
    color: white;
    text-align: center;
    padding: 20px;
}

/* Navigation */
nav ul {
    list-style: none;
    padding: 0;
    background: #444;
    text-align: center;
}

nav ul li {
    display: inline;
    margin: 0 15px;
}

nav ul li a {
    color: white;
    text-decoration: none;
    font-weight: bold;
}

/* Sections */
section {
    background: white;
    margin: 20px auto;
    padding: 20px;
    max-width: 800px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

img {
    width: 100%;
```

```
    max-width: 400px;
    display: block;
    margin: 10px 0;
}

/* Lists */
ul {
    padding: 0;
}

ul li {
    background: #ddd;
    margin: 5px 0;
    padding: 10px;
    border-left: 5px solid #333;
}

/* Form */
form {
    display: flex;
    flex-direction: column;
}

label {
    margin-top: 10px;
    font-weight: bold;
}

input, textarea {
    padding: 10px;
    margin-top: 5px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

button {
    margin-top: 10px;
    background: #333;
    color: white;
    padding: 10px;
    border: none;
    cursor: pointer;
    border-radius: 5px;
}

button:hover {
    background: #555;
}

/* Table */
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 10px;
```

```
}

th, td {
    border: 1px solid #ddd;
    padding: 10px;
    text-align: left;
}

th {
    background: #333;
    color: white;
}

/* Footer */
footer {
    background: #222;
    color: white;
    text-align: center;
    padding: 15px;
    margin-top: 20px;
}
```