

Abstract

This paper scrutinizes the use of different practices and methods in Client Side Scripting, enabling viewer to get the complete concept of different aspects of it. To satisfy this we have implemented a Student Grade Calculator using Javascript and CSS. It is done by implementing a series of loops and condition which is also used as a reference to the output, satisfying every need of a perfect microproject.



RSM POLY

INDEX

Sr. No.	Chapter Name	Page No.
1	Introduction	3
2	History	6
3	Code	9
4	Output	13
5	Conclusion	15
6	References	16



RSM POLY

Chapter-1

INTRODUCTION

1. Client Side Scripting:

A client-side dynamic web page is a web page whose construction is controlled by an application server processing server-side scripts. In server-side scripting, parameters determine how the assembly of every new web page proceeds, including the setting up of more client-side processing.

A client-side dynamic web page processes the web page using JavaScript running in the browser as it loads. JavaScript can interact with the page via Document Object Model, or DOM, to query page state and modify it. Even though a web page can be dynamic on the client-side, it can still be hosted on a static hosting service such as GitHub Pages or Amazon S3 as long as there isn't any server-side code included.

A dynamic web page is then reloaded by the user or by a computer program to change some variable content. The updating information could come from the server, or from changes made to that page's DOM. This may or may not truncate the browsing history or create a saved version to go back to, but a dynamic web page update using AJAX technologies will neither create a page to go back to, nor truncate the web browsing history forward of the displayed page. Using AJAX, the end user gets one dynamic page managed as a single page in the web browser while the actual web content rendered on that page can vary. The AJAX engine sits only on the browser requesting parts of its DOM, the DOM, for its client, from an application server.

DHTML is the umbrella term for technologies and methods used to create web pages that are not static web pages, though it has fallen out of common use since the popularization of AJAX, a term which is now itself rarely used. Client-side-scripting, server-side scripting, or a combination of these make for the dynamic web experience in a browser.



Fig.1. Client Side Scripting

2. JavaScript:

JavaScript, often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute JavaScript on users' devices. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O. JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.



Fig.2. JavaScript

3. HTML:

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.[2] A form of HTML, known as HTML5, is used to display video and audio, primarily using the `<canvas>` element, in collaboration with javascript.

HTML



Fig.3. HTML



Maratha Vidya Prasarak Samaj's

Rajarshi Shahu Maharaj Polytechnic, Nashik

MSBTE CODE-1002

DTE CODE:- 5247

DEPARTMENT OF COMPUTER TECHNOLOGY

ACADEMIC YEAR 2021-22



Chapter-2

History

1. Client Side Scripting:

It is difficult to be precise about "dynamic web page beginnings" or chronology because the precise concept makes sense only after the "widespread development of web pages". HTTP has been in use since 1990, HTML, as standard, since 1996. The web browser's rise in popularity started with Mosaic in 1993. It is obvious, however, that the concept of dynamically driven websites predates the Internet, and in fact HTML. For example, in 1990, before general public use of the Internet, a dynamically driven remotely accessed menu system was implemented by Susan Biddlecomb, who was Director of Computer Support of the USC Health Care system at the University of Southern California BBS on a 16 line TBBS system with TDBS add-on database.

The introduction of JavaScript (then known as LiveScript) enabled the production of client-side dynamic web pages, with JavaScript code executed in the client's browser. The letter "J" in the term AJAX originally indicated the use of JavaScript, as well as XML. With the rise of server side JavaScript processing, for example, Node.js, originally developed in 2009, JavaScript is also used to dynamically create pages on the server that are sent fully formed to clients.

Execusite introduced the first dynamic website solution for the professional marketplace in June 1997. Execusite was acquired by Website Pros (now Web.com) in January 2000. During the bust cycle of the Dot-com bubble, the original Execusite founders bought back the company from Website Pros (December 2000). Execusite was later acquired by Wolters-Kluwer in December 2001 and was re-branded as CCH Site Builder.

Netscape introduced an implementation of JavaScript for server-side scripting with Netscape Enterprise Server, first released in December, 1994 (soon after releasing JavaScript for browsers).[1][2]

Client-side scripting was later used in early 1995 by Fred DuFresne while developing the first website for Boston, MA television station WCVB. The technology is described in US patent 5835712. The patent was issued in 1998 and is now owned by Open Invention Network (OIN). In 2010 OIN named Fred DuFresne a "Distinguished Inventor" for his work on server-side scripting.

RSM POLY

2. JavaScript:

The first web browser with a graphical user interface, Mosaic, was released in 1993. Accessible to non-technical people, it played a prominent role in the rapid growth of the nascent World Wide Web. The lead developers of Mosaic then founded the Netscape corporation, which released a more polished browser, Netscape Navigator, in 1994. This quickly became the most-used.

During these formative years of the Web, web pages could only be static, lacking the capability for dynamic behavior after the page was loaded in the browser. There was a desire in the burgeoning web development scene to remove this limitation, so in 1995, Netscape decided to add a scripting language to Navigator. They pursued two routes to achieve this: collaborating with Sun Microsystems to embed the Java programming language, while also hiring Brendan Eich to embed the Scheme language.

Netscape management soon decided that the best option was for Eich to devise a new language, with syntax similar to Java and less like Scheme or other extant scripting languages. Although the new language and its interpreter implementation were called LiveScript when first shipped as part of a Navigator beta in September 1995, the name was changed to JavaScript for the official release in December.

The choice of the JavaScript name has caused confusion, implying that it is directly related to Java. At the time, the dot-com boom had begun and Java was the hot new language, so Eich considered the JavaScript name a marketing ploy by Netscape.



Fig.4.

James Gosling, the creator of JavaScript

3. HTML:

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991. It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGMLguid, an in-house Standard Generalized Markup Language (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4.

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents. However, the SGML concept of generalized markup is based on elements (nested annotated ranges with attributes) rather than merely print effects, with also the separation of structure and markup; HTML has been progressively moved in this direction with CSS.

Berners-Lee considered HTML to be an application of SGML. It was formally defined as such by the Internet Engineering Task Force (IETF) with the mid-1993 publication of the first proposal for an HTML specification, the "Hypertext Markup Language (HTML)" Internet Draft by Berners-Lee and Dan Connolly, which included an SGML Document type definition to define the grammar.

After the HTML and HTML+ drafts expired in early 1994, the IETF created an HTML Working Group, which in 1995 completed "HTML 2.0", the first HTML specification intended to be treated as a standard against which future implementations should be based.



Fig.4. Tim Berners-Lee, the creator of HTML



Maratha Vidya Prasarak Samaj's

Rajarshi Shahu Maharaj Polytechnic, Nashik

MSBTE CODE-1002

DTE CODE:- 5247

DEPARTMENT OF COMPUTER TECHNOLOGY

ACADEMIC YEAR 2021-22



Chapter-3

Code

1. Index.html

```
<!DOCTYPE html>
<html>
<head>
<title>Student Grade Calculator</title>
<!-- link for font -->
<link
href=
"https://fonts.googleapis.com/css?family=Righteous&display=swap"
rel="stylesheet"
/>
<link rel="stylesheet" href="style.css" />
</head>
<body>
<!-- main html -->
<div class="container">
<h1><center>Student grade calculator</center></h1>
<div class="screen-body-item">
<div class="app">
<div class="form-group">
<!-- option for taking the input -->
<center><input
type="text"
class="form-control"
placeholder="AJP"
id="AJP"
/></center>
</div>
<div class="form-group">
<center><input
type="text"
class="form-control"
placeholder="OSY"
id="OSY"
/></center>
</div>
<div class="form-group">
<center><input
type="text"
class="form-control"
```



Maratha Vidya Prasarak Samaj's

Rajarshi Shahu Maharaj Polytechnic, Nashik

MSBTE CODE-1002

DTE CODE:- 5247

DEPARTMENT OF COMPUTER TECHNOLOGY

ACADEMIC YEAR 2021-22



```
placeholder="CSS"
id="CSS"
/></center>
</div>
<div class="form-group">
<center><input
type="text"
class="form-control"
placeholder="STE"
id="STE"
/></center>
</div>
<div>
<center><input
type="button"
value="Show Percentage"
class="form-button"
onclick="calculate()"
/></center>
</div>
</div>
</div>
<!-- for showing the result-->
<div class="form-group showdata">
<p id="showdata"></p>
</div>
</div>
<!--adding external javascript file-->
<script src="script.js"></script>
</body>
</html>
```

बहुजन हिताय बहुजन सुखाय

RSM POLY



Maratha Vidya Prasarak Samaj's

Rajarshi Shahu Maharaj Polytechnic, Nashik

MSBTE CODE-1002

DTE CODE:- 5247

DEPARTMENT OF COMPUTER TECHNOLOGY

ACADEMIC YEAR 2021-22



2. Script.js

```
// Function for calculating grades
const calculate = () => {
// Getting input from user into height variable.
let AJP = document.querySelector("#AJP").value;
let OSY = document.querySelector("#OSY").value;
let CSS = document.querySelector("#CSS").value;
let STE = document.querySelector("#STE").value;
let grades = "";
// Input is string so typecasting is necessary. */
let totalgrades =
parseFloat(AJP) +
parseFloat(OSY) +
parseFloat(CSS) +
parseFloat(STE);
// Checking the condition for the providing the
// grade to student based on percentage
let percentage = (totalgrades / 400) * 100;
if (percentage <= 100 && percentage >= 80)
{
grades = "A";
}
else if (percentage <= 79 && percentage >= 60) {
grades = "B";
}
else if (percentage <= 59 && percentage >= 40)
{
grades = "C";
} else
{
grades = "F";
}
// Checking the values are empty if empty than
// show please fill them
if (AJP == "" || OSY == ""
|| CSS == "" || STE == "") {
document.querySelector("#showdata").innerHTML
= "<center>Please enter all the fields</center>";
} else {
// Checking the condition for the fail and pass
if (percentage >= 39.5) {
document.querySelector(
"#showdata"
).innerHTML =
<center>Out of 400 your total is ${totalgrades}
and percentage is ${percentage}%. <br>
```




Maratha Vidya Prasarak Samaj's

Rajarshi Shahu Maharaj Polytechnic, Nashik

MSBTE CODE-1002

DTE CODE:- 5247

DEPARTMENT OF COMPUTER TECHNOLOGY

ACADEMIC YEAR 2021-22



```
Your grade is ${grades}. You are Pass. </center>`;
} else {
document.querySelector(
"#showdata"
).innerHTML =
`<center>Out of 400 your total is ${totalgrades}
and percentage is ${percentage}%. <br>
Your grade is ${grades}. You are Fail.</center> `;
}
}
};
```

3. Style.css

```
* {
margin: 0;
padding: 0;
box-sizing: border-box;
}
body {
align-content: center;
font-size: 12px;
}

.container {
flex: 0 1 700px;
margin: auto;
padding: 10px;
}

.screen-body-item {
flex: 1;
padding: 50px;
}
input {
margin: 10px 10px 10px;
}

.showdata {
color: black;
font-size: 1.2rem;
padding-top: 10px;
padding-bottom: 10px;
}
```

RSM POLY

Chapter-4

OUTPUT

Student grade calculator

Out of 400 your total is 400 and percentage is 100%.
Your grade is A. You are Pass.

Fig.1. Grade A

Student grade calculator

Out of 400 your total is 300 and percentage is 75%.
Your grade is B. You are Pass.

Fig.2. Grade B



Maratha Vidya Prasarak Samaj's

Rajarshi Shahu Maharaj Polytechnic, Nashik

MSBTE CODE-1002

DTE CODE:- 5247

DEPARTMENT OF COMPUTER TECHNOLOGY

ACADEMIC YEAR 2021-22



Student grade calculator

Show Percentage

Out of 400 your total is 200 and percentage is 50%.
Your grade is C. You are Pass.

Fig.3. Grade C

Student grade calculator

Show Percentage

Out of 400 your total is 120 and percentage is 30%.
Your grade is F. You are Fail.

Fig.4. Grade F

RSM POLY



Maratha Vidya Prasarak Samaj's

Rajarshi Shahu Maharaj Polytechnic, Nashik

MSBTE CODE-1002

DTE CODE:- 5247

DEPARTMENT OF COMPUTER TECHNOLOGY

ACADEMIC YEAR 2021-22



Chapter-6

CONCLUSION

Client side and server side are web development terms that describe where application code runs. Web developers will also refer to this distinction as the frontend vs. the backend, although client-side/server-side and frontend/backend aren't quite the same. In a serverless architecture, the serverless vendor hosts and assigns resources to all server-side processes, and the processes scale up as application usage increases.

Much of the Internet is based on the client-server model. In this model, user devices communicate via a network with centrally located servers to get the data they need, instead of communicating with each other. End user devices such as laptops, smartphones, and desktop computers are considered to be 'clients' of the servers, as if they were customers obtaining services from a company. Client devices send requests to the servers for webpages or applications, and the servers serve up responses.

The client-server model is used because servers are typically more powerful and more reliable than user devices. They also are constantly maintained and kept in controlled environments to make sure they're always on and available; although individual servers may go down, there are usually other servers backing them up. Meanwhile, users can turn their devices on and off, or lose or break their devices, and it should not impact Internet service for other users.

Servers can serve multiple client devices at once, and each client device sends requests to multiple servers in the course of accessing and browsing the Internet.

बहुजन हिताय बहुजन सुखाय

RSM POLY



Maratha Vidya Prasarak Samaj's

Rajarshi Shahu Maharaj Polytechnic, Nashik

MSBTE CODE-1002

DTE CODE:- 5247

DEPARTMENT OF COMPUTER TECHNOLOGY

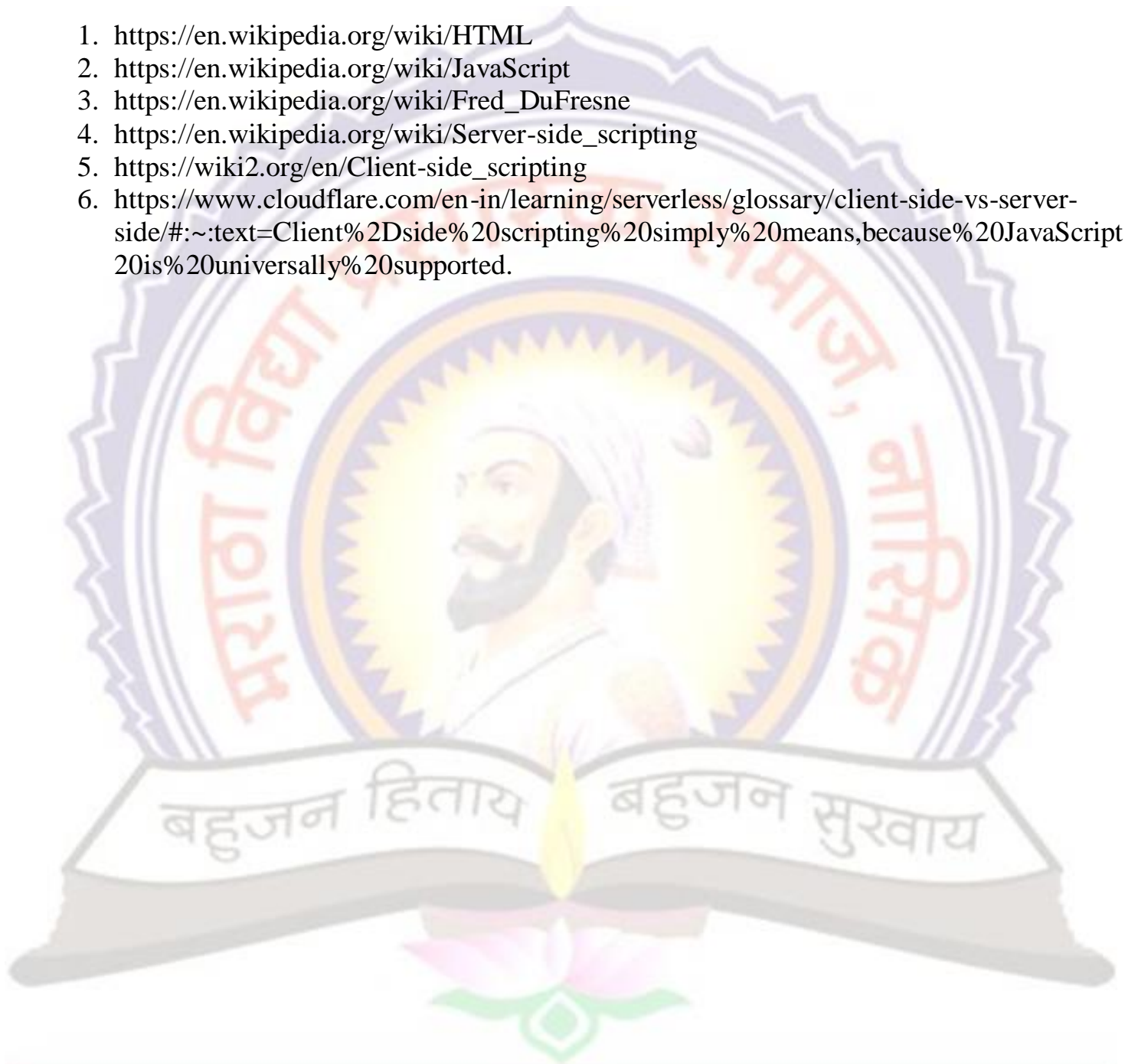
ACADEMIC YEAR 2021-22



Chapter-7

References

1. <https://en.wikipedia.org/wiki/HTML>
2. <https://en.wikipedia.org/wiki/JavaScript>
3. https://en.wikipedia.org/wiki/Fred_DuFresne
4. https://en.wikipedia.org/wiki/Server-side_scripting
5. https://wiki2.org/en/Client-side_scripting
6. <https://www.cloudflare.com/en-in/learning/serverless/glossary/client-side-vs-server-side/#:~:text=Client%2Dside%20scripting%20simply%20means,because%20JavaScript%20is%20universally%20supported.>



RSM POLY