HTML, CSS, JS

Skyler Badge, Shourya Agarwal & Navneel Mandal

HTML

Introduction

HTML stands for Hyper Text Markup Language

- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

HTML Tags

<TagName> Stuff
….. More Stuff
</TagName>

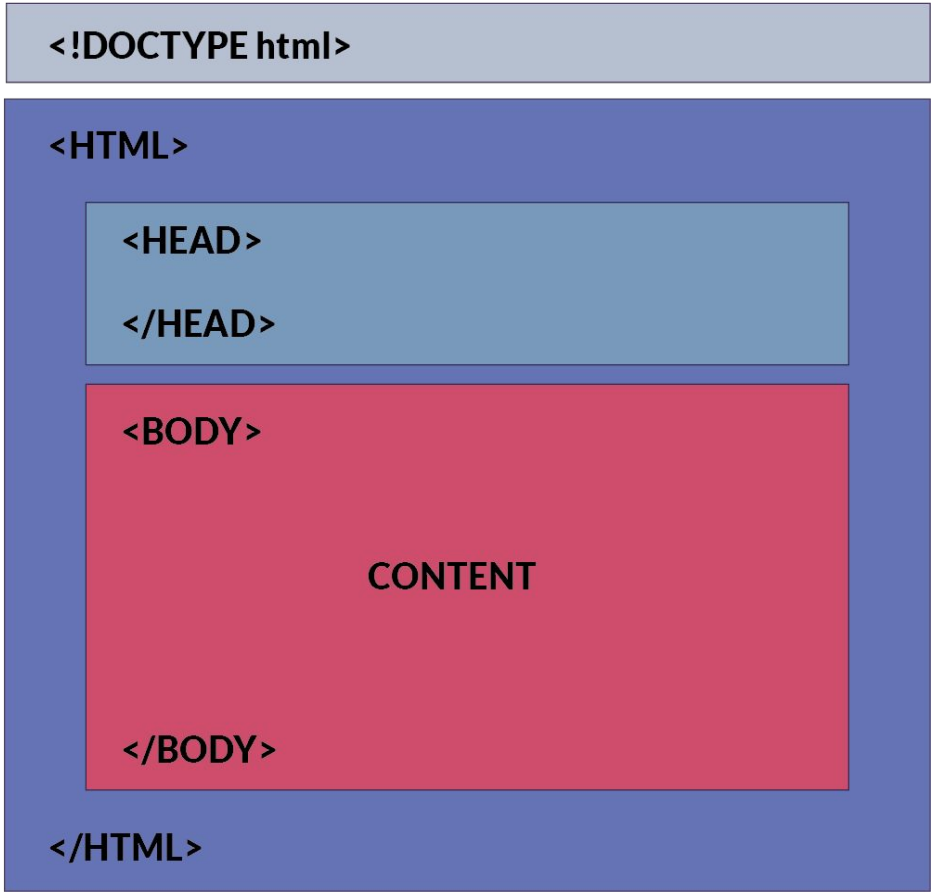
OR

<TagName> Stuff….. More Stuff</TagName>

- NoT CaSe SenSiTivE
- Formatting is not preserved

Page Structure

- The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.
- The `<head>` element contains meta information about the document (data about data)
- The `<body>` element contains the visible page content



Where to write the code?

- You can write it in any text editor like the default **notepad** available in windows
- After writing the code save the file with an extension .html
- Click on the file and open with any browser

Alternatively,

You can use an IDE like netbeans, visual studio , etc.

Basic Tags

Paragraph Tag: <p>

Defines a paragraph

```
<!DOCTYPE html>

<html>
  <head>
  </head>
  <body>

    <p>Hello World!!</p>

  </body>
</html>
```


localhost:8383/DevLecture/index
localhost:8383/DevLecture/index.html

Hello World!!

Basic Tags

Heading Tag:

<h1> , <h2> , ..., <h5>, <h6>

Defines a heading

```
<html>
  <head>
  </head>
  <body>
    <h1>Hello World!!</h1>
    <h2>Hello World!!</h2>
    <h3>Hello World!!</h3>
    <h4>Hello World!!</h4>
    <h5>Hello World!!</h5>
    <h6>Hello World!!</h6>
  </body>
</html>
```

Hello World!!

Hello World!!

Hello World!!

Hello World!!

Hello World!!

Hello World!!

Basic Tags

Line break:

Introduces a line break

DOES NOT HAVE A CLOSING TAG

```
<!DOCTYPE html>

<html>
  <head>
  </head>
  <body>

    <p>Hello <br> World!!</p>

  </body>
</html>
```

Hello
World!!

Tag Attributes

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

Basic Tags

Image tag: <img>

Attributes : src

Specifies the source of the image

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

      <img src ="iitd.png" alt="Image cannot be displayed">

   </body>
</html>
```


भारतीय प्रौद्योगिकी संस्थान दिल्ली
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Some More Tags

- Anchor tag:

  This can be used to create hyperlinks
  Eg Click me

- /

  This is used to define blocks in html. We can give specific attributes to these blocks

Head Tag

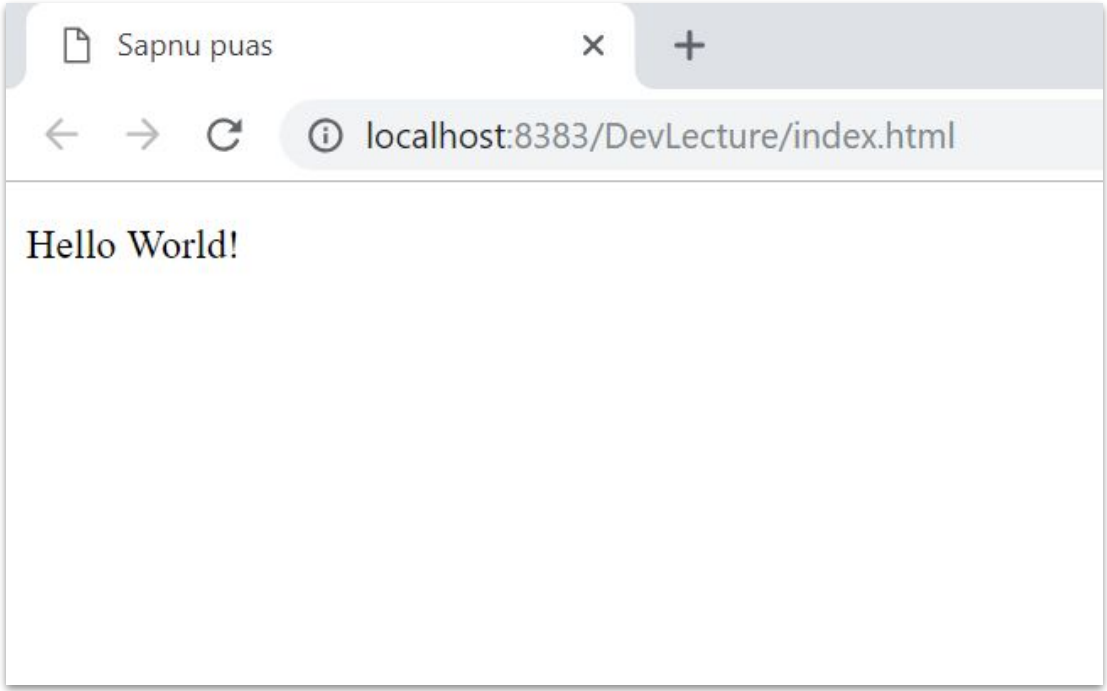
- <title>

  For specifying the name of the webpage as it should appear in the title bar.

- <link>

  To link external style sheets

```
<html>
  <head>
    <title>
      Sapnu puas
    </title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```



CSS

Introduction

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

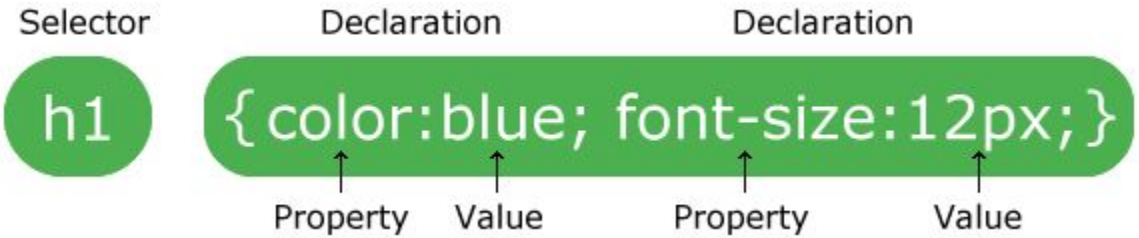
How to link the style sheet?

Use the link tag in the head of an html file.

Create a style.css file in the same folder.

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Syntax



Eg.

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


Sapnu puas
localhost:8383/DevLecture/index.html
Hello World!

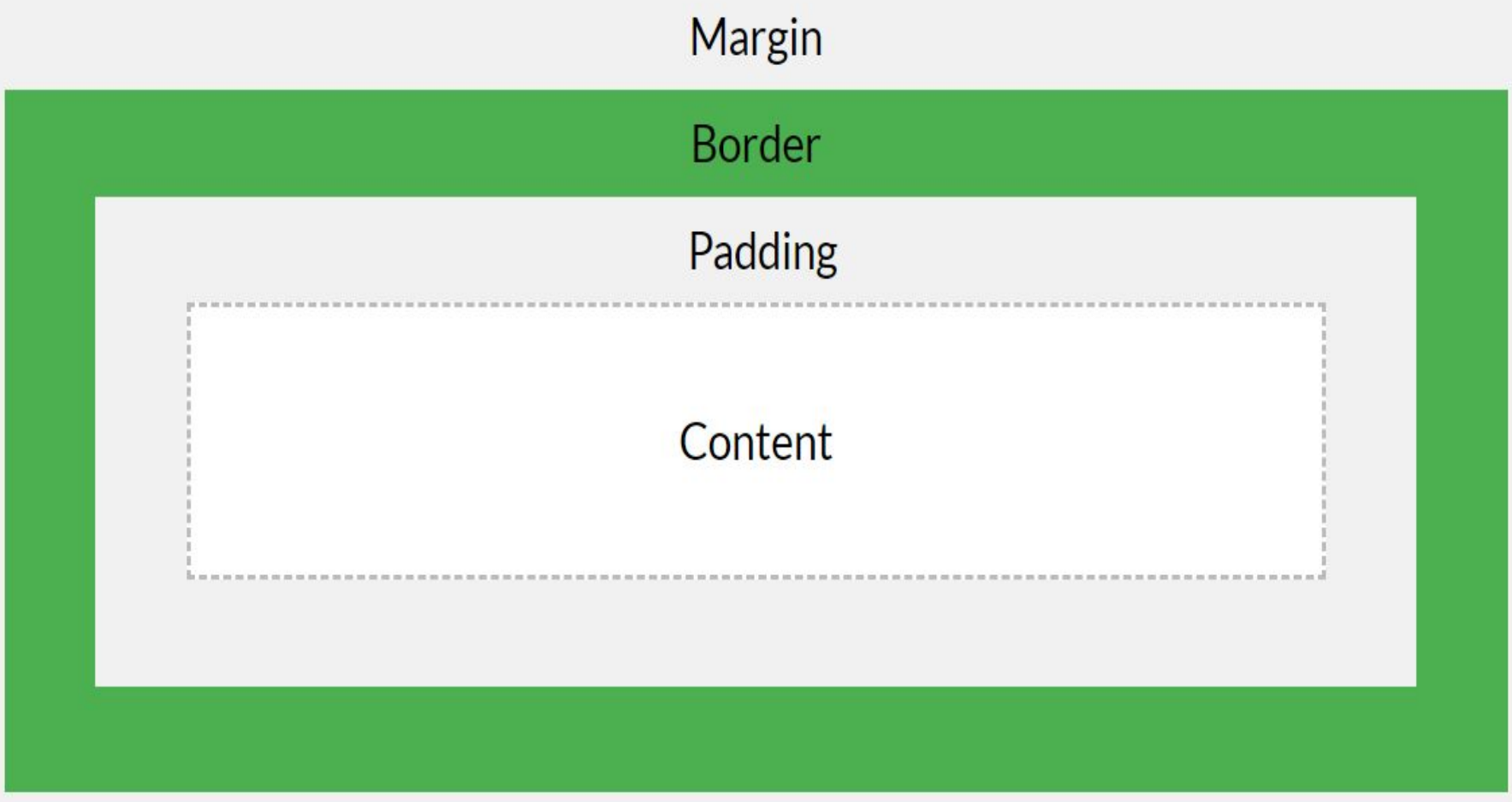
More Examples

```
<html>
  <head>
    <title>Sapnu puas</title>
    <link rel="stylesheet" type="text/css"
href="style.css">
  </head>
  <body>
    <div id = "one" ><h1>I GO LEFT!</h1></div>
    <div id = "two" ><h1>I GO RIGHT!</h1></div>
  </body>
</html>
```

```
#one {
   color: red;
   background-color: yellow;
   float:left
}

#two{
   color: white;
   background-color: black;
   text-align: right;
   float: right;
}
```

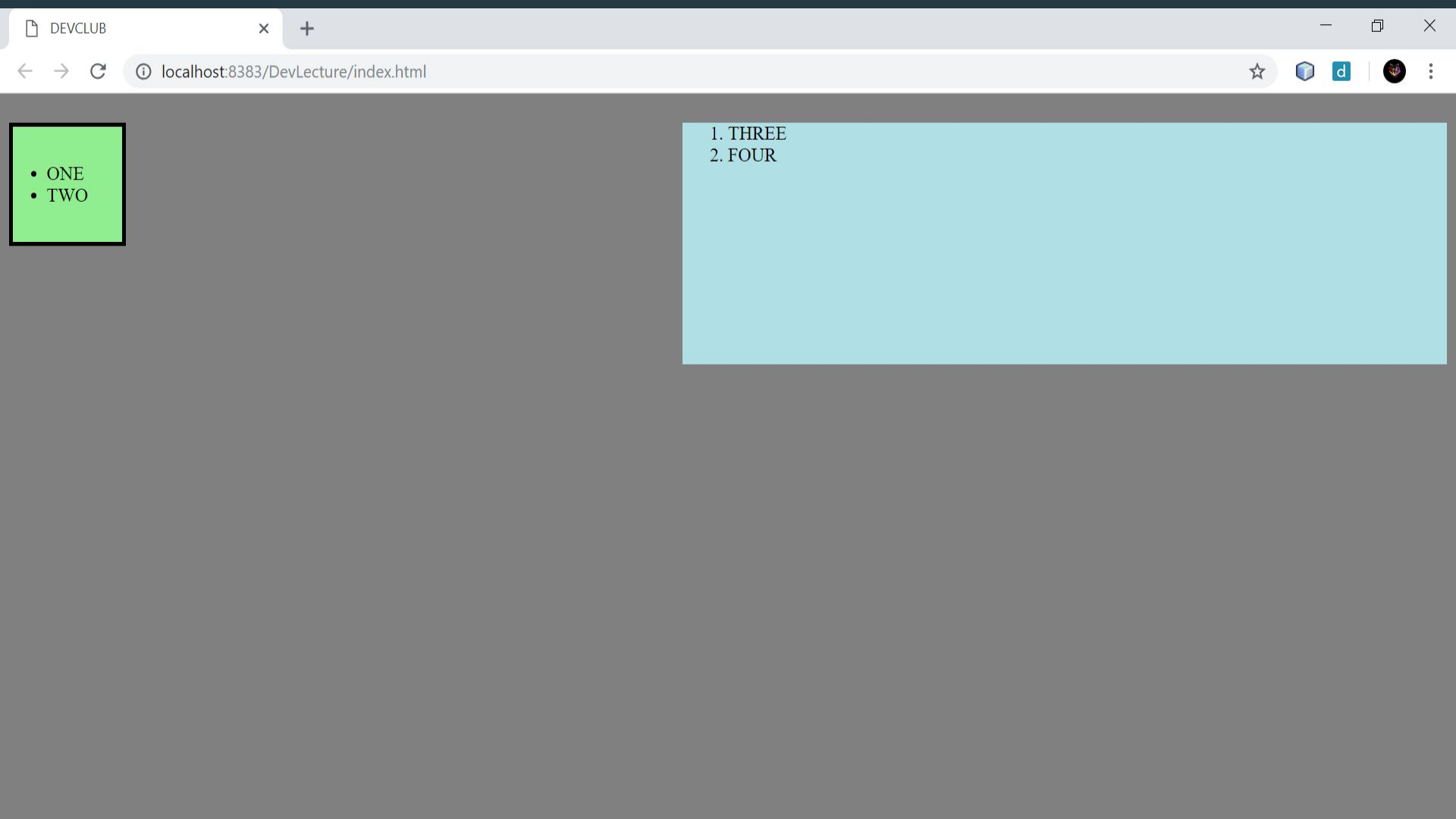

Sapnu puas
localhost:8383/DevLecture/index.html
I GO LEFT!
I GO RIGHT!


Margin
Border
Padding
Content

```
<body>
    <ul id = "one">
      <li>ONE</li>
      <li>TWO</li>
    </ul>
    <ol id = "two">
      <li>THREE</li>
      <li>FOUR</li>
    </ol>

</body>
```

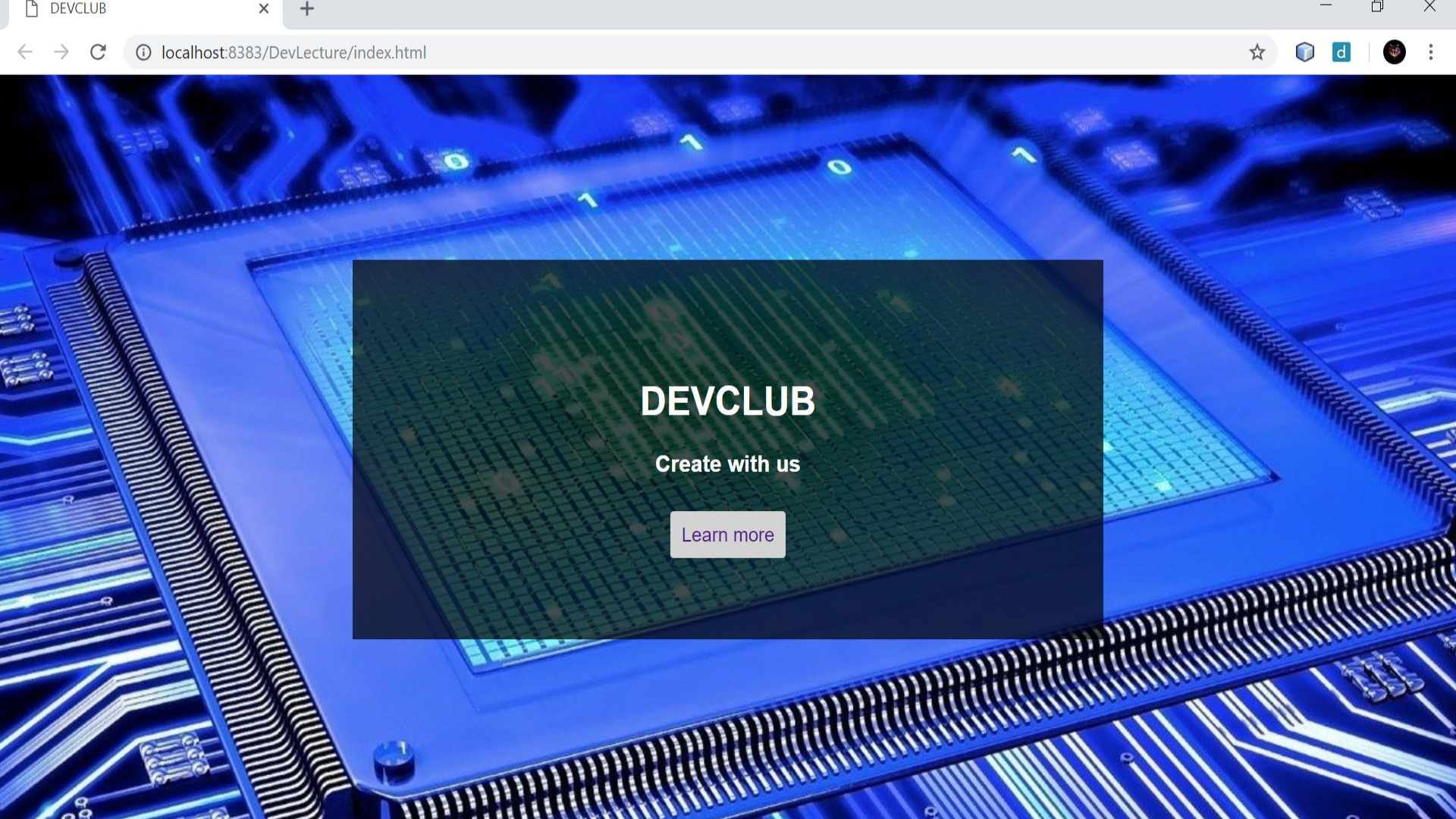
```
body {
   background-color: grey;
}
#one {
   border-style: solid;
   border-width: 3px;
   float:left;
   padding: 30px;
   background-color:lightgreen;
}
#two{
   height: 200px;
   width: 50%;
   background-color: powderblue;
   float:right;
}
```


DEVCLUB
localhost:8383/DevLecture/index.html
ONE
TWO
1. THREE
2. FOUR

```
<html>
   <head>
      <title>DEVCLUB</title>
      <link rel="stylesheet" type="text/css" href="style.css">
   </head>
   <body>
      <div id="content">
         <h1>DEVCLUB</h1>
         <h3>Create with us</h3><br>
         <a href="https://devclub.in/about.html" id="button">Learn more</a>
      </div>
   </body>
</html>
```

```
body{
    text-align: center;
    color: white;
    background: url(banner.jpg) center;
}
#content{
    font-family: Helvetica;
    padding-top: 6%;
    padding-bottom: 6%;
    margin-top: 12%;
    margin-bottom: 12%;
    background-color: rgba(0,0,0,0.7);
    max-width: 660px;
    margin-left: auto;
    margin-right: auto;
}
```

```
#button{
    padding: 10px;
    background-color: lightgray;
    border-style: solid;
    border-radius: 3px;
    border: 3px;
    text-decoration: none;
}
```


DEVCLUB
localhost:8383/DevLecture/index.html
DEVCLUB
Create with us
Learn more

JavaScript

What does JS do?

How to use JS?

1. **Internal**

Script Tag.

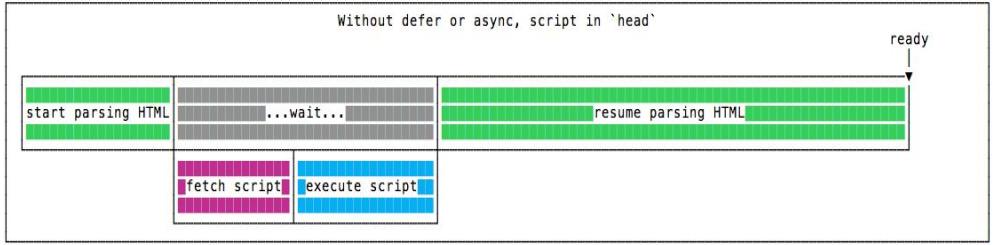
<script type = “text/javascript”></script>

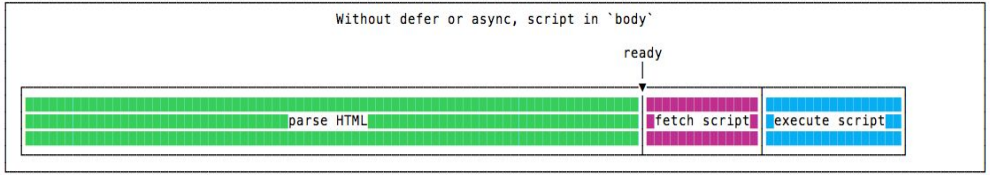
Can be placed in the head or the body.

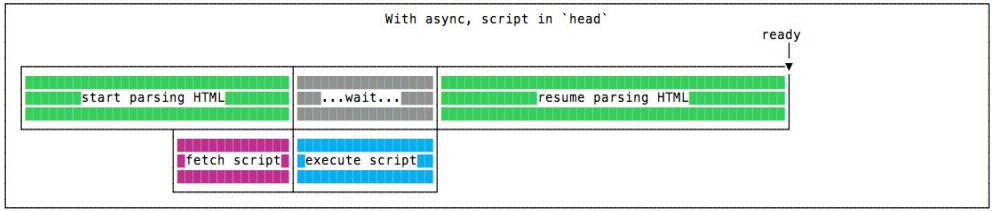
Why is JS at the bottom of the page better?

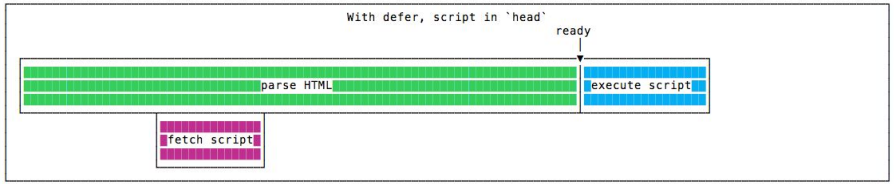
2. **External**

<script src=”extScript.js” (async/defer)> </script>









https://stackoverflow.com/questions/5329807/benefits-of-loading-js-at-the-bottom-as-opposed-to-the-top-of-the-document

External v/s Internal

Inline scripts

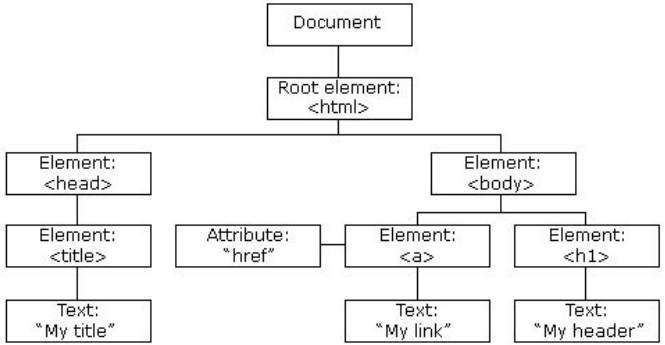
- Are loaded in the same page so is not necessary to trigger another request.
- Are executed immediately.
- The async and defer attributes have no effect

External scripts

- Gives better separation of concerns and maintainability.
- The async and defer attributes have effect so if this attributes are present the script will change the default behavior. This is not possible with inline scripts.
- Once a external script is downloaded the browser store it in the cache so if another page reference it no additional download is required.

https://flaviocopes.com/javascript-async-defer/

The HTML DOM



It has been discussed in the HTML Part…

The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:

- The HTML elements as **objects**
- The **properties** of all HTML elements
- The **methods** to access all HTML elements
- The **events** for all HTML elements

In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

Some DOM methods

1) document.getElementById(“ ”).innerHTML = “new Value”
2) document.getElementById(“ ”).attribute = *new value*
3) document.getElementById(“ ”).style.property = *new style*

Output Data

1) innerHTML
2) document.write
3) window.alert
4) Console.log

What are the differences?

JS statements

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

JavaScript ignores multiple spaces. You can add white space to your script to make it more readable.

JavaScript statements can be grouped together in code blocks, inside curly brackets {...}.

JavaScript statements often start with a **keyword** to identify the JavaScript action to be performed.

List of keywords: break, continue, do...while, for, function…

The JavaScript syntax defines two types of values: Fixed values and variable values.

Fixed values are called **literals**. Variable values are called **variables**.

.

JS Variables

1) Var
2) Let

Difference?

Which to use and why?

https://www.geeksforgeeks.org/difference-between-var-and-let-in-javascript/

https://stackoverflow.com/questions/762011/whats-the-difference-between-using-let-and-var-to-declare-a-variable-in-jav

JS constants, operators

Numbers are written with or without decimals

Strings are text, written within double or single quotes

JavaScript uses **arithmetic operators** (+ - * / % ++ --) to **compute** values

BDMAS follows. When operations have same precedence, left to right.

JavaScript uses an **assignment operator** (=) to **assign** values to variables (-=, +=, *= ,/=)

Not all JavaScript statements are "executed".

Code after double slashes **//** or between **/*** and ***/** is treated as a **comment**.

Comments are ignored, and will not be executed:

JS Operators contd…

Comparison

== , ===, !=, !===, >, <, >=, <=, ?

Logical

&&, ||, !

Type Operators

Typeof

Instanceof

Bitwise: **Do yourself**

JS Identifiers

In JavaScript, identifiers are used to name variables (and keywords, and functions, and labels).

The rules for legal names are much the same in most programming languages.

In JavaScript, the first character must be a letter, or an underscore (_), or a dollar sign ($).

Subsequent characters may be letters, digits, underscores, or dollar signs.

All JavaScript identifiers are **case sensitive**.

Hyphens are not allowed in JS

JS Data Types

Primitive

String, Number, Boolean, Undefined, null

Difference b/w undefined and null? Typeof null?

Derived(Non Primitive)

Object, Array, RegExp

JS is dynamic

JS Data Type

String arithmetic:

- \+ Results in concatenation

var t = “50”+ 23 + 45;

var s = 23 + 45 + “50”;

Type Conversion

String(999) or (999).toString

toExponential() or toFixed()

Similarly Number(“111”)

Number(“Wow”) = ??

parseFloat() and parseInt()

Functions

```
Function myF(p1,p2){

console.log(“wow”);

return “wow”;

}
```

JS Events

An HTML event can be something the browser does, or something a user does.

- An HTML web page has finished loading
- An HTML input field was changed
- An HTML button was clicked

```
<element event='some JavaScript'>

Object.event = function(){functionName}


Do it yourself: addEventListener??
```

Event	Description
onchange	An HTML element has been changed
Onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
Onmouseout	The user moves the mouse away from an HTML element
Onkeydown	The user pushes a keyboard key
Onload	The browser has finished loading the page

String methods

stringName.length

stringName.indexOf(“string”)

stringName.slice(startInd, endInd)

stringName.toLowerCase()

stringName.charAt(index)

stringName.split(“.”) omit? Empty string?

Loops

For loop

for/in loop

While loop

Do...while loop

What is break?

What is continue?

JS is asynchronous

What is asynchronous?

Is it desirable?

How to handle it?

Learn more about:

previousSIbling
nextSibling
firstChild
Parent

typeof(null)
typeof(undefined)

regExp

Feedback: bit.ly/devlec2