Google Developer Student Clubs

Web Development BootCamp

Web Development?

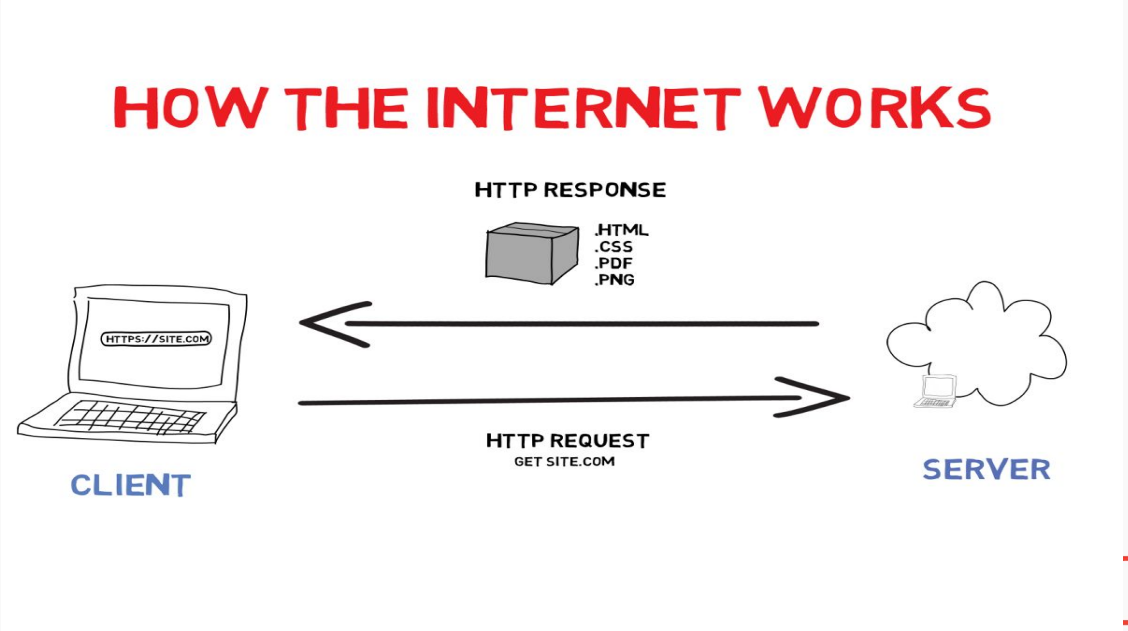
It is the process of building websites and web applications using code to create their design, functionality, and interactivity.

History of Web

Tim Berners-Lee, a British scientist, invented the World Wide Web (WWW) in 1989, while working at CERN. The Web was originally conceived and developed to meet the demand for automated information-sharing between scientists in universities and institutes around the world.

http://info.cern.ch/hypertext/WWW/TheProject.html

How Internet Works?



Frontend

The part of a website or application with which users interact directly.
Includes- HTML,CSS,JavaScript

Backend

The part users don't see, including servers, databases, and application logic.
— Python, Ruby, PHP,Javascript

Full stack

Developers proficient in both front end and back end development.

Google Developer Student Clubs

What do we learn???

a. HTML(HYPERTEXT MARKUP LANGUAGE)
b. CSS (CASCADING STYLE SHEETS)
c. BOOTSTRAP
d. JAVASCRIPT

OVERVIEW

HTML

BOOTSTRAP

Day-1

Day-2

BOOTCAMP END

PROJECT SUBMISSION
(TOP 2 BEST PROJECTS WILL BE AWARDED WITH GOODIES)

CSS

JAVASCRIPT

HTML

(HyperText Markup Language)

HTML

Hyper Text Markup Language

It is like a set of instructions for webpages. It tell browsers how to stuff like text, images and links. It uses tags, which are like labels, to organize everything neatly.

It’s basically the language websites use to display content .

Importance of HTML

- Structure - It provides the structure for web pages, organizing content logically.
- Cross platform Compatibility - It works consistently across different browsers and devices.
- Accessibility - It supports features making web content accessible to all users.
- Integration - Easily integrate with CSS and JavaScript for design and interactivity.

What is HTML element?

An HTML element is defined by a start tag, some content, and an end tag:

```
<tagname> Content goes here...
</tagname>
```

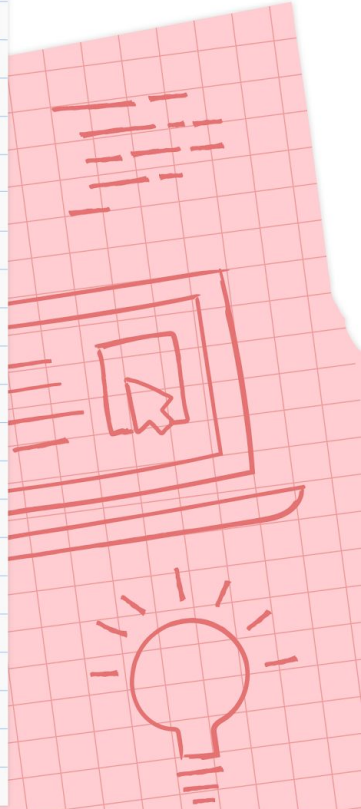
Example: <html></html>, <body></body>, <br>,<hr>

Example

```
<!Doctype html>
<html>
<head>
    <title>Webpage/title>
</head>
<body>
    <h1>First web page</h1>
    <p>...</p>
</body>
</html>
```

HTML commonly used tags..

1. **<html>**: Defines the root of an HTML document.
2. **<head>**: Contains meta-information about the document.
3. **<title>**: Sets the title of the document (displayed in the browser's title bar).
4. **<body>**: Contains the content of the document.
5. **<h1>,<h2>, <h3>, <h4>, <h5>, <h6>**: Heading tags, used to define headings of different levels.
6. **<p>**: Defines a paragraph.
7. **<a>**: Creates a hyperlink.
8. ** **: Defines a division or a section in an HTML document.
9. ****: Defines a section of text within a larger document.



Text formatting Tags

- <b> Defines bold text
- Defines emphasized text
- <i> Defines italic text
- Defines strong text
- <sub> Defines subscripted text
- <sup> Defines superscripted text
- <u> Defines underline text
- <strike> Defines strike text

Images

<img> : Defines an image

- **src** : display an image on a page,Src stands for "source"
- **alt**: Define "alternate text" for an image
- **width** : Defines the width of the image
- **height** : Defines the height of the image

List

Ordered

```
<ol type="A">
  <li>First item</li>
  <li>Second item</li>
  <li>Third item</li>
</ol>
```

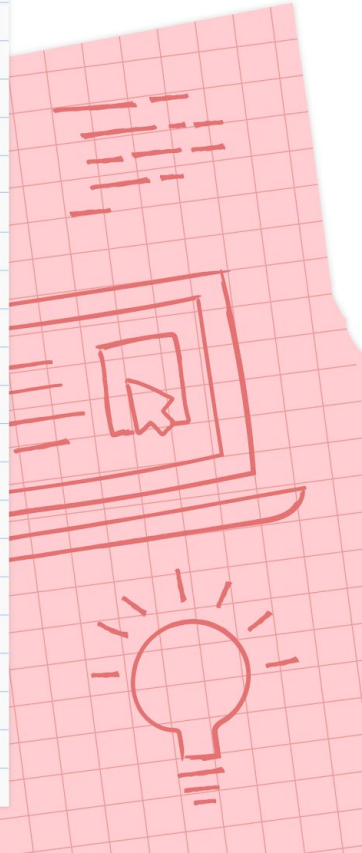
Unordered

```
<ul type="circle">
 <li>Item 1</li>
 <li>Item 2</li>
 <li>Item 3</li>
</ul>
```

Tables

<table> : used to create table
<tr> : table is divided into rows
<td >: each row is divided into data cells
<th> : Headings in a table
<Caption> : caption to the table

- **Colspan** : No of column working with will span
- **rowspan** : No of rows working with will span
- **Border** : attribute takes a number

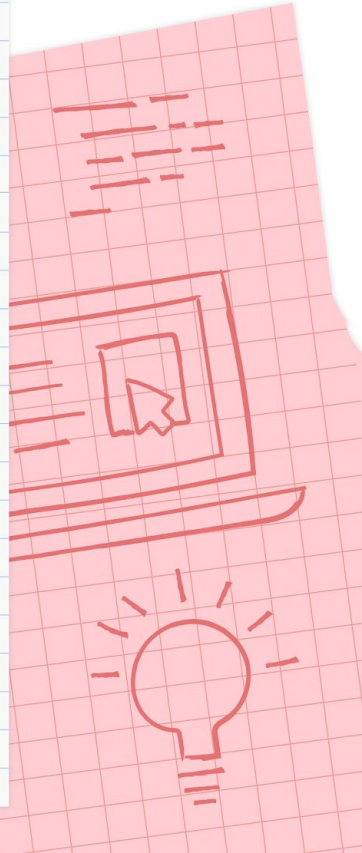


Form

<form>
<input>
<textarea>
<select>
<option>

- Text
- Password
- Email
- Number
- Date
- Radio

- Placeholder



CSS

(Cascading Style Sheets)

CSS (CASCADING STYLE SHEETS)

What is CSS ?

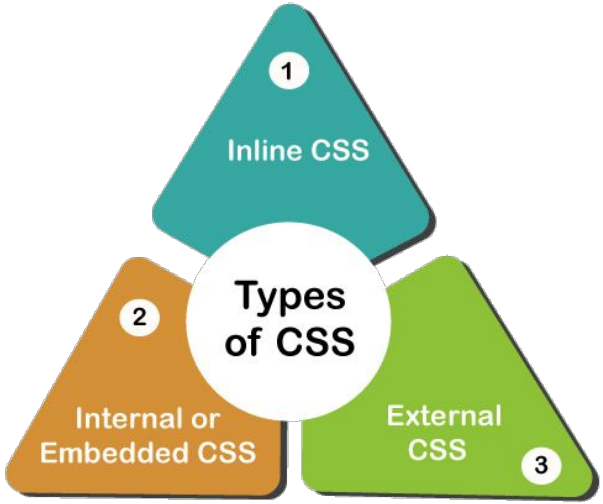
- CSS (Cascading Style Sheets) is a language used to control the presentation and layout of web pages. It defines how HTML elements are displayed on screen, including their colors, fonts, sizes, positions, and more. CSS helps make websites look visually appealing and structured, enhancing the user experience.

FOR EXAMPLE

```
h2{
    Background-color : green;
}
```

Website for more details related to WEB DEV → www.mdn.com

Types of CSS



- **Inline CSS →**

```
<p style="color: blue; font-size: 16px;">This is a paragraph with inline CSS.</p>
```

- **Internal CSS**

```
 <head>
 <style>
p {
        color: blue;
        font-size: 16px;
  }
 </style>
</head>
```

- **External CSS**

Linking → `<head><link rel="stylesheet" href="styles.css"></head>`

```
body {
  font-family: Tahoma, Arial, sans-serif;
  font-size: 13px;
  color: black;
  background: white;
  margin: 8px;
}
h1 {
  font-size: 19px;
  margin-top: 0px;
  margin-bottom: 5px;
  border-bottom: 1px solid black
}
.shaded {
  background: #d0d0ff;
}
```

CSS:

```
<body>
  <h1>First Section Heading</h1>
  <p>
    Here is the first paragraph, containing
    text that really doesn't have any use
    or meaning; it just prattles on and on,
    with no end whatsoever, no point to
    make, really no purpose for existence
    at all.
  </p>
  <div class="shaded">
    <h1>Another Section Heading</h1>
    <p>
      Another paragraph.
    </p>
  </div>
</body>
```

HTML:

Style sheet contain one or more **CSS Rules**

Selector

```
body {
  font-family: Tahoma, Arial, sans-serif;
  color: black;
  background: white;
  margin: 8px;
}
```

Declaration Block

Property

Value

SELECTORS

In CSS Selectors are used to target the HTML elements on our WEB PAGES that we want to style.

Selector

```
h1 {
  color: blue;
  background-color: yellow;
}

p {
  color: red;
}
```

Types of Selectors

1. **Universal selector**

```
*{
    Background-color : green;
}
```

2. **Element selector**

```
h2{
    Background-color : green;
}
```

3. **Class selector**

```
.myclass{
    Property : value;
}
```

4. Id selector

```
#myid{
    Property : value;
}
```

5. Descendent selector

Outer level Element

```
div p{
    Property : value;
}
```

Inner level element

CSS Pseudo Selectors

→ A pseudo-selector in CSS is a keyword that specifies a special state of the selected element(s).

hover - Apply rule when mouse is over element (e.g. tooltip)

```
p:hover, a:hover {
  background-color: yellow;
}
```

a:link, a:visited - Apply rule when link has been visited or not visited (link)

```
a:visited {
  color: green;
}
```

```
a:link {
  color: blue;
}
```

Pseudo Elements

Pseudo-elements in CSS allow you to style parts of an element without adding extra HTML. They are denoted by double colons (::) and enable you to style elements in ways that aren't possible with just classes or IDs.

For Example →

```
p::first-letter {
    font-size: 150%;
    color: red;
    }
```

CSS Properties

Control many style properties of an element:

- Coloring
- Size
- Position
- Visibility
- Many more: (e.g. `p: { text-decoration: line-through; }`)

- Also used in animation

Color - Properties: `color` & `background_color`

Must ultimately turn into red, green, and blue intensities between 0 and 255:

- Predefined names: `red`, `blue`, `green`, `white`, etc. (140 standard names)
- 8-bit hexadecimal numbers for red, green, blue: `#ff0000` → 
  R G B
- 0-255 decimal intensities: `rgb(255,255,0)` → 
  R G B
- Percentage intensities: `rgb(80%,80%,100%)` → 
  R G B

Example: `h1: { color: red; }`

Some more common properties

`background-image`: image for element's background

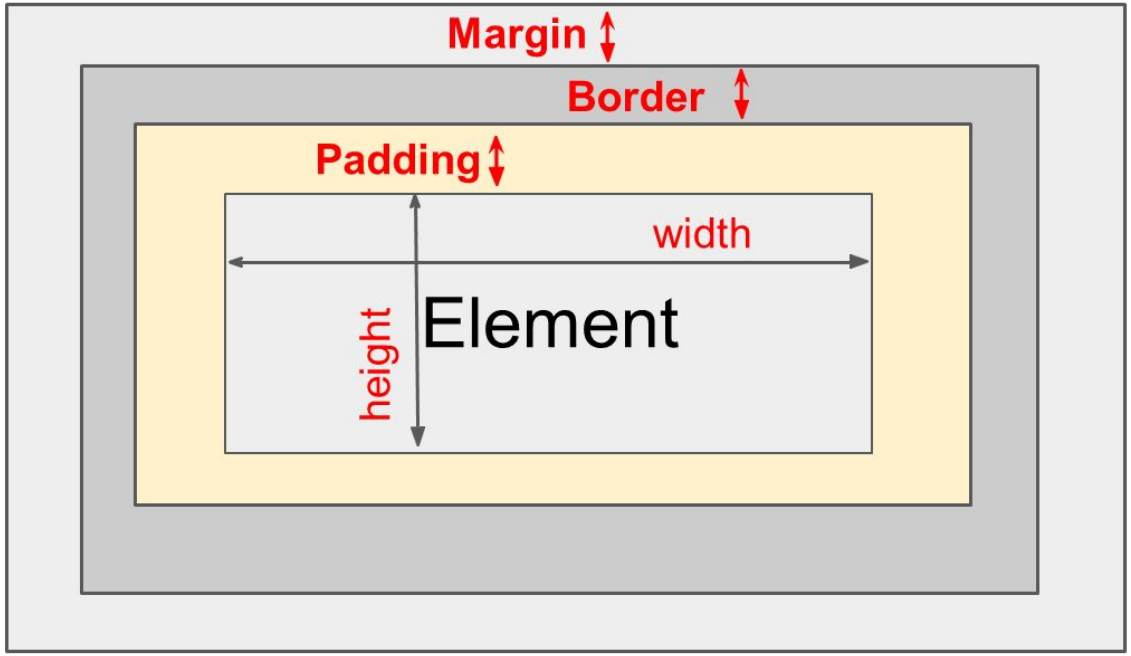
`background-repeat`: should background image be displayed in a repeating pattern (versus once only)

`font, font-family, font-size, font-weight, font-style`: font information for text

`text-align, vertical-align: Alignment: center, left, right`

`cursor` - Set the cursor when over element (e.g. `help`)

CSS Box Model



Total element width =
width +
left padding +
right padding +
left border +
right border +
left margin +
right margin

Margin & Padding
Transparent

CSS distance units

Absolute	
2px	pixels
1mm	millimeters
2cm	centimeters
0.2in	inches
3pt	printer point 1/72 inch
Relative	
2em	2 times the element's current font size
3rem	3 times the root element's current font size

Size Properties - Element, pad, margin, border

width - Override element defaults
height

padding-top
padding-right
padding-bottom
padding-left

margin-top
margin-right
margin-bottom
margin-left

border-bottom-color
border-bottom-style
border-bottom-width
border-left-color
border-left-style
border-left-width
border-right-color
border-right-style
border-right-width
etc.

```
p {
  border: 5px solid red;
}
```

Element visibility control properties

`display: none;` - Element is not displayed and takes no space in layout.

`display: inline;` - Element is treated as an inline element.
`display: block;` - Element is treated as a block element.

`display: flex;` - Element is treated as a flex container.
`display: grid;` - Element is treated as a grid container.

Position Property

The position property in CSS determines how an element is positioned within its parent container. Here are the different values for the position property:

- **static (default):** The element follows the normal document flow.
- **relative:** The element is positioned relative to its default position using the top, right, bottom, and left properties.
- **fixed:** The element is positioned at a fixed location on the screen using the same properties as relative.
- **absolute**: The element is positioned relative to an ancestor element with position: absolute, also using the top, right, bottom, and left properties.
- **sticky:** It is used to make an element stick to a specific position on the screen as the user scrolls.

Intermission

Day-2

Bootstrap

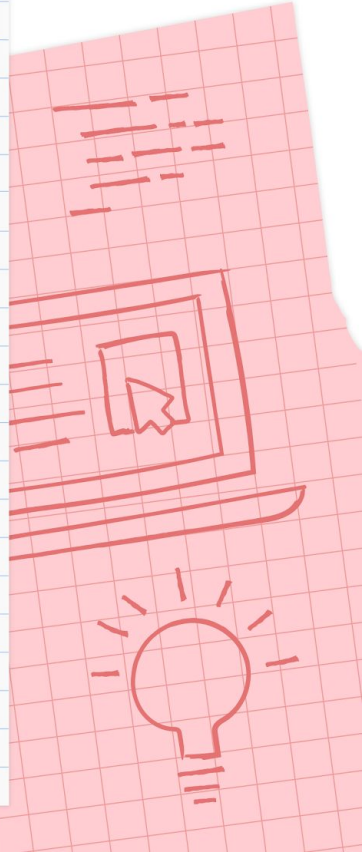
What is Bootstrap?

- Bootstrap is a popular front-end framework for building responsive and mobile-first websites.
- Developed by Twitter, Bootstrap provides a collection of tools, CSS, and JavaScript components to streamline web development.
- Its primary goal is to facilitate the creation of consistent, visually appealing, and user-friendly web interfaces.

How to use

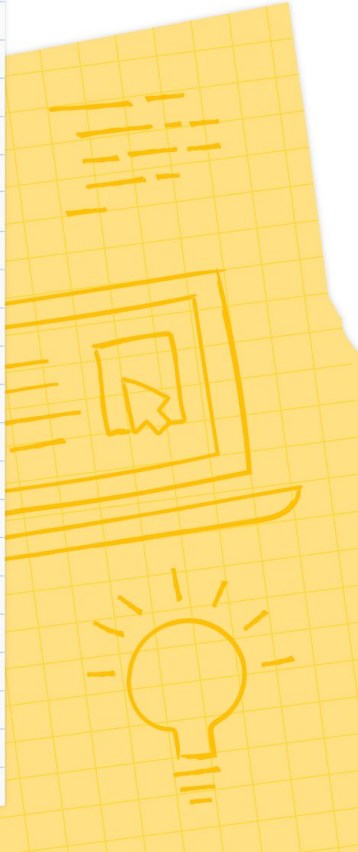
A basic understanding of HTML is required to start learning Bootstrap. Some familiarity with how CSS works (CSS Selectors and Visual Rules) would be helpful,

1. Download Bootstrap from getbootstrap.com
2. Include Bootstrap from a CDN



Components

- Accordion
- Alerts
- Badge
- Breadcrumb
- Buttons
- Button group
- Card
- Carousel
- Close button
- Collapse
- Dropdowns
- List group
- Modal
- Navs & tabs
- Navbar
- Offcanvas
- Pagination
- Popovers
- Progress
- Scrollspy
- Spinners
- Toasts
- Tooltips



Advantages of Bootstrap

1. Fewer Cross browser bugs
2. A consistent framework that supports major of all browsers and CSS compatibility fixes
3. Lightweight and customizable
4. Responsive structures and styles
5. Good documentation and community support
6. Great grid system

Portfolio

How To upload your Project on GitHub.

Project Submission Link

https://forms.gle/7Y5Gdio23mmPZF669

JS

JavaScript

JAVASCRIPT

JavaScript often abbreviated as JS, is a programming language and core technology of the Web, alongside HTML and CSS. 99% of websites use JavaScript on the client side for webpage behavior. Web browsers have a dedicated JavaScript engine that executes the client code.

JS

VARIABLES

Name of a storage location.

```
a =10;
Age =23;
Name = Tony;
```

DATA TYPES

a. Number - 1, 2, 3, 45, 10, 2.3
b. Boolean - true, false
c. String - Tony, John
d. Undefined - undefined
e. Null- null

Numbers In JS

1. Integers (-45 , 50)
2. Floating Numbers with Decimal (4.6 ,-8.9)

Linking in JS File

<script src = “app.js”></script>

Console.log() ⟶To write a message on console

```
console.log(“Hello world”);
console.log(1234);
console.log(2+2);
console.log(“Tony”,”Stark” ,123);
```

Operations In JS

Operands

a + b

Operators

I. Modulo (remainder operator '%')
II. Exponentiation (power operator 2**3 = 8)
III. Addition
IV. Subtraction
V. Multiplication
VI. Division

Operators Precedence

This is a general order of solving an Expression.

()
**** → **Exponentiation**
***,/,%**
+,-

let ,const & var Keywords

Syntax of Declaring Variables.

```
let age = 23;
let cgpa = 8.4;
```

const Keyword (Values of constants that can't be changed with re-assignment & they can't be declared)

```
const year = 2025;
const pi = 3.14;
```

Var Keyword

Many modern JavaScript style guides and best practices recommend using `let` over `var` due to its more predictable behavior and better scoping rules.

Old syntax of writing variables.

```
var age = 23;
var cgpa = 8.4;
Var num1 =20;
var num2 = 30;
var sum = num1+num2;
```

Way of writing identifiers

1. **camelCase -> fullName**
2. snake_case -> full_name
3. PascalCase -> FullName

Google Developer Student Clubs

Strings in JS

```
let name = “Tony”;
let role = `Student`;
let gender = ‘Male’;
```

Strings are text or sequence of characters.

To find the length of any string use → **name.length**

Google Developer Student Clubs

Null & Undefined values in JS

Undefined → A variable in JS that has not been assigned any value.

```
let a ;
>> undefined
```

NULL → The NULL value represents the intentional absence of any object value.

```
let a = null;
a;
>> undefined
```

Increment & Decrement Operator

1. **Post-increment:** `variable++` or `variable = variable + 1`
   - This increases the value of the variable after its current value is used in the expression.
2. **Pre-increment:** `++variable`
   - This increases the value of the variable before it is used in the expression.

```
int x = 5;
int y = x++; // y is assigned the value of x (5), then x is incremented (x becomes 6)

int a = 10;
int b = ++a; // a is incremented first (a becomes 11), then b is assigned the value of a (11)
```

1. **Post-decrement:** `variable--` or `variable = variable - 1`
   - This decreases the value of the variable after its current value is used in the expression.
2. **Pre-decrement:** `--variable`
   - This decreases the value of the variable before it is used in the expression.

```
int x = 8;
int y = x--; // y is assigned the value of x (8), then x is decremented (x becomes 7)

int a = 15;
int b = --a; // a is decremented first (a becomes 14), then b is assigned the value of a (14)
```

Conditional Statement

1. If-else
2. Nested if-else

Logical Operators

Used to combine expressions.

&& (AND)

|| (OR)

! (NOT)

Alert & Prompt

Alert displays an alert message on the page.

```
alert("something is wrong");
```

Prompt displays a dialog box that asks user for some input.

```
prompt(" Please enter your age " )
```

Console.error → used for printing an error on console

```
console.error("This is a sample")
```

Console.warn → used for warning messages on console

```
console.warn("This is a sample")
```

String Methods

Methods → Actions that can be performed on object.

FORMAT → *stringName.method();*

1. **Trim method** → it trim whitespaces from both the ends of a string and returns a new one.

```
let message = “      hello      “;
>> message.trim();
>>hello;
```

2. **toUpperCase()** → “Tony” → “TONY”
3. **toLowerCase()** → “sTaRk” → “stark”

Thank you!!