# Day 1 - Intro

## intro to browser

### what language does the browser understand

>Note : chrome is built on c++

1.HTML

2.css

3.js

## multiple types of Engine

- Rendering Engine

- Parse HTML + CSS

- Chrome: Blink

- Js Engine

- parses JavaScript

- Chrome: V8

## Shortcuts

- alt + up/down ⬆️⬇️

emoji windows + .

## Usecase For markdown

- AI(chargpt)

- Notes

- Blog

- Slides

- Docs

```Js

console.log("Hello, world")

```

| Month | Savings |

| -------- | ------- |

| January | $250 |

| February | $80 |

| March | $420 |

# convert html,css,js to reacct js

webpack+Babel

### Day\_02

# CDN

# Port

22,80,443

every website runs on 443 port

# TCP

>to avoid file transfer and conect right person.

- Acknowledgement

- web browsing

- Email

- 3 way handshake

# UDP

- live stream

- Online games

- no handshake

# OFC (optical fiber cable)

# IP packets

# HTTP timeline

- 1996 1.0 version

- 1.1

- 1.1

- 1.2

## Assignment

http 2 vs http 3.

### HTTP codes

- 400 client mistake

- 500 server

- 300 redirection

### HTTP methods

- get

- post

- put

- delete

## Types of engine+

- Rendering Engine

- JS Engine

HTML ---> HTML Parser ----> DOM

--

JS ---> JS engine --

--

CSS ---> CSS parser ---> CSSOM

### JS

- Dynamically typed language.

- variables

## Datatypes

### Primitive Datatypes

1.number

2.string

3.boolean

4.undefined

### Day 03

- DNS propecation.

## templetlitral

## Arthimetic operators

# + - / %

## relational operators (gives boolean ans)

- < > != == <= >=

## Logical (gives boolean ans)

- && ||

## prompt

## confirm

## string methods

- toLowerCase()

- toUpperCase()

- trim()

- padstart(),padend()

## property

- length

## undefined vs defined

- var x; is an undefined ,declared but not assinged any value.(undefined)

- console.log(y) ,is not decalred so defined.(defined)

## Array

- Group of same data types.

## Object

- group of different data types.

- key - value pairs.

### Day 05

# Array

- index & values.

# object

- key & value

## ES6

- Numeric\_seperators

## types of declartions

- var ,let ,const

- var

- we can redeclare a same varible name.

- var name = "app";

- var name = "app";

- let

- cannot redeclare a same variable name.

let name = "app";

let name = "app"; (error)

- const

- cannot reassign and Redeclare.

### ..spread operator

- const x = [100,23,45];

-const y=[...x] // copy by value

## Array destructuring

- const [t1, t2] = [100,200];

- const [t1, t2=80] = [100,200,300]; when t2 is undefined then it takes default value

- const [t1, t2] = [100,200];

## object destructuring

🛡️ Marvel-Themed JavaScript Assessment

---

## ⭐ Easy

<details>

<summary>1. Hero ID Formatter (⭐)</summary>

### Task

Prompt the user for a hero’s name. Convert it to lowercase and uppercase. Display both on separate lines.

```js

const name = prompt("Enter a name");

var lowerCase = name.toLowerCase();

var uppperCase = name.toUpperCase();

console.log(lowerCase);

console.log(uppperCase);

```

#### Example 1

\*\*Input\*\*: `"BlackPanther"`

\*\*Output\*\*:

```

blackpanther

BLACKPANTHER

```

#### Example 2

\*\*Input\*\*: `" ScarletWitch "`

\*\*Output\*\*:

```

scarletwitch

SCARLETWITCH

```

</details>

---

<details>

<summary>2. Suit Version Type Check (⭐)</summary>

### Task

Tony inputs the suit version as a string. Convert it to number and log the type before and after conversion.

#### Example 1

\*\*Input\*\*: `"85"`

\*\*Output\*\*:

```

string

number

```

#### Example 2

\*\*Input\*\*: `"007"`

\*\*Output\*\*:

```

string

number

```

</details>

---

<details>

<summary>3. Hulk Smash Echo (⭐)</summary>

### Task

Ask the user for a sound effect (e.g., `"Smash!"`) and echo it 3 times in a single line.

#### Example 1

\*\*Input\*\*: `"Smash!"`

\*\*Output\*\*:

```

Smash!Smash!Smash!

```

#### Example 2

\*\*Input\*\*: `"Boom "`

\*\*Output\*\*:

```

Boom Boom Boom

```

</details>

---

## ⭐⭐⭐ Medium

<details>

<summary>4. Stark Credits Calculator (⭐⭐⭐)</summary>

### Task

Prompt for two separate Stark wallet balances (as strings). Convert both to numbers and display the total credits.

### Answer

```js

const balance1 = prompt("Enter a no:");

const balance2 = prompt("Enter a no;")

const total = parseInt(balance1) + parseInt(balance2) ;

console.log(`Your total balance is ${total} credits.`);

```

#### Example 1

\*\*Input\*\*: `"1000"`, `"5000"`

\*\*Output\*\*:

```

Your total balance is 6000 credits.

```

#### Example 2

\*\*Input\*\*: `"250"`, `"250"`

\*\*Output\*\*:

```

Your total balance is 500 credits.

```

</details>

---

<details>

<summary>5. Shield Temperature Scanner (⭐⭐⭐)</summary>

### Task

Prompt for the shield’s surface temperature in Celsius.

\* > 120 → "🔥 Overheated!"

\* < -10 → "❄️ Frozen!"

\* Else → "🛡️ Stable."

```js

const celsius = +prompt("Enter a celsius");

var ans;

if(celsius >= 120){

ans = "🔥 Overheated!";

}else if(celsius <= 10){

ans = "❄️ Frozen!";

}else{

ans = "🛡️ Stable.";

}

console.log(ans);

```

#### Example 1

\*\*Input\*\*: `130`

\*\*Output\*\*:

```

🔥 Overheated!

```

#### Example 2

\*\*Input\*\*: `25`

\*\*Output\*\*:

```

🛡️ Stable.

```

</details>

---

<details>

<summary>6. Infinity Stone Energy Rank (⭐⭐⭐)</summary>

### Task

Prompt for the power level (0–100) of an infinity stone.

Rank:

\* 90+ → "Legendary 💎"

\* 60–89 → "Potent ⚡"

\* <60 → "Weak 🪨"

```js

var ans;

var pow = +prompt("Power level");

if (pow >= 90) {

ans = "Legendary 💎";

} else if (pow >= 60) {

ans = "Potent ⚡";

} else {

ans = "Weak 🪨";

}

console.log(ans);

```

#### Example 1

\*\*Input\*\*: `95`

\*\*Output\*\*:

```

Legendary 💎

```

#### Example 2

\*\*Input\*\*: `58`

\*\*Output\*\*:

```

Weak 🪨

```

</details>

---

<details>

<summary>7. Stark Email Generator (⭐⭐⭐)</summary>

### Task

Prompt for a name. Trim and convert it to lowercase. Output the Stark Industries email.

```js

var name = prompt("Enter Name: ").trim().toLowerCase();

var format = name +"@starkindustries.com";

console.log(format);

```

#### Example 1

\*\*Input\*\*: `" Peter "`

\*\*Output\*\*:

```

peter@starkindustries.com

```

#### Example 2

\*\*Input\*\*: `"WARMachine"`

\*\*Output\*\*:

```

warmachine@starkindustries.com

```

</details>

---

<details>

<summary>8. Jarvis Access Gate (⭐⭐⭐)</summary>

### Task

Prompt for an access code (number).

\* 1000–1999 → “Access granted”

\* Exactly 1700 → “⚠️ Override mode activated”

\* Else → “Access denied”

```js

const code = +prompt("Enter a Code");

var ans;

if(code == 1700){

ans = "⚠️ Override mode activated";

} else if(code >= 1000 && code <= 1999){

ans = "Access granted";

}else{

ans = "Access denied";

}

console.log(ans);

```

#### Example 1

\*\*Input\*\*: `1024`

\*\*Output\*\*:

```

Access granted

```

#### Example 2

\*\*Input\*\*: `1700`

\*\*Output\*\*:

```

⚠️ Override mode activated

```

</details>

---

## ⭐⭐⭐⭐ Hard

<details>

<summary>9. Speed Duel – Quicksilver vs Hawkeye (⭐⭐⭐⭐)</summary>

### Task

Prompt for their speeds. Log who is faster and by how much (in km/h).

```js

const person1 = prompt("Enter a first name:");

const person1Speed = +prompt("Enter speed");

const person2 = prompt("Enter a second name:");

const person2Speed = +prompt("Enter speed");

var ans;

var diff = Math.abs(person1Speed - person2Speed);

if(person1Speed > person2Speed){

ans = `${person1} is faster than ${person2} by ${diff} km/h.`;

}else if(person2Speed > person1Speed){

ans = `${person2} is faster than ${person1} by ${diff} km/h.`;

}else{

ans = `${person1} and ${person2} are same speed.`;

}

console.log(ans);

```

#### Example 1

\*\*Input\*\*: `180`, `90`

\*\*Output\*\*:

```

Quicksilver is faster than Hawkeye by 90 km/h.

```

#### Example 2

\*\*Input\*\*: `150`, `160`

\*\*Output\*\*:

```

Hawkeye is faster than Quicksilver by 10 km/h.

```

</details>

---

<details>

<summary>10. S.H.I.E.L.D. Agent Validation (⭐⭐⭐⭐)</summary>

### Task

Prompt for a name. Normalize the input (trim + lowercase).

Validate against allowed agents: `"natasha"`, `"clint"`, `"nick"`.

\* If match → "🛡️ Access granted"

\* Else → "⛔ Unauthorized"

```js

const agents = prompt("Enter a name").trim().toLowerCase();

if(agents == "natasha" || agents == "clint" || agents == "nick"){

console.log("🛡️ Access granted");

}else{

console.log("⛔ Unauthorized");

}

```

#### Example 1

\*\*Input\*\*: `" NiCk "`

\*\*Output\*\*:

```

🛡️ Access granted

```

#### Example 2

\*\*Input\*\*: `"tony"`

\*\*Output\*\*:

```

⛔ Unauthorized

```

</details>

Web Development Assessment – MCQs

How did Internet Explorer win the first browser war?

A) Better performance

B) Free and pre-bundled with Windows

C) Open-source community

D) Strong privacy features

Q1 B

In a three-layer architecture, which part is responsible for handling logic and server-side processing?

A) Frontend

B) Backend

C) Database

D) Browser Engine

Q2 B

How is the world physically connected to allow global internet access?

A) Satellite links

B) Optical wireless routers

C) Submarine OFC cables

D) Wi-Fi satellites

Q3 C

Which HTTP version runs on QUIC protocol for faster performance?

A) HTTP 1.1

B) HTTP 2.0

C) HTTP 3.0

D) HTTP 0.9

Q4 B

Which category of HTTP status codes represents client errors?

A) 1xx

B) 2xx

C) 3xx

D) 4xx

Q5 D

Which of the following best describes what happens when you type a URL in the browser and hit Enter?

A) It directly opens the saved page from the cache

B) The browser engine renders the page without contacting any server

C) DNS resolves the domain to an IP, TCP connects, then HTTP request is made

D) The rendering engine creates a pixel map first and then calls the server

Q6 C

Which is NOT part of the browser's internal architecture?

A) UI Backend

B) Rendering Engine

C) CSS Engine

D) Networking

Q7 D

The JavaScript engine “V8” was developed by:

A) Mozilla

B) Apple

C) Microsoft

D) Google

Q8 A

In browser architecture, the Rendering Engine is responsible for:

A) Parsing JavaScript

B) Rendering CSS and HTML to pixels

C) Managing cookies

D) Handling DNS queries

Q9 B

In the CSS rule p.error { color: red; }, what is p.error?

A) Property

B) Value

C) Selector

D) Declaration

Q10 C

What role does DNS play in web browsing?

A) Encrypts HTTP traffic

B) Resolves IP to MAC address

C) Translates domain names to IP addresses

D) Stores user session cookies

Q11 C

Which rendering engine powers Chromium-based browsers like Chrome and Brave?

A) Trident

B) Gecko

C) Blink

D) EdgeHTML

Q12 C

Which statement about MAC addresses is true?

A) They change every time a device connects to a new network

B) They are assigned by ISPs

C) They are fixed at the hardware level

D) They are encrypted using HTTPS

Q13 C

Why was the V8 engine a game-changer for JavaScript performance?

A) It was built in C#

B) It used JIT (Just-In-Time) compilation for speed

C) It replaced CSS rendering

D) It integrated SQL parsing

Q14 A

Which of the following is NOT a valid HTTP method?

A) GET

B) FETCH

C) POST

D) DELETE

Q15 B

jsCopy to Clipboard

// Do this:

// This always returns a boolean value

const isObject = (obj) => !!obj && typeof obj === "object";

// Or this:

const isObject = (obj) => Boolean(obj) && typeof obj === "object";

// Or this:

const isObject = (obj) => obj !== null && typeof obj === "object";

// Instead of this:

// This may return falsy values that are not equal to false

const isObject = (obj) => obj && typeof obj === "object";