

Document 1 — JavaScript Basics for Beginners

Title: JavaScript Basics for Beginners

Author: [Your Name]

Date: 28 October 2025

Table of contents

1. What is JavaScript?
 2. Setting up a simple environment
 3. Core language concepts
 4. DOM basics
 5. Small project: To-Do list (code)
 6. Next steps & resources
-

1. What is JavaScript?

JavaScript (JS) is a high-level scripting language used to build interactive behavior on web pages, run servers (Node.js), and power desktop and mobile apps. It's event-driven, prototype-based, and works alongside HTML and CSS to create modern web experiences.

2. Setting up a simple environment

You only need three things: - A modern browser (Chrome, Firefox, Edge) - A text editor (VS Code is recommended) - Optional: Node.js (for running JavaScript outside the browser)

Quick test: open the browser console (F12 → Console) and type:

```
console.log('Hello from JavaScript!')
```

If it prints the message, JS is running.

3. Core language concepts

Variables

```
let name = 'Aman';  
const PI = 3.14;  
var legacy = true; // avoid in modern code
```

Data types

- strings, numbers, booleans, null, undefined, objects, arrays

Functions

```
function add(a, b) {  
  return a + b;  
}  
const multiply = (a, b) => a * b;
```

Control flow

if / else, for loops, while loops, switch statements.

Objects & Arrays

```
const person = { name: 'Aman', age: 28 };  
const list = [1, 2, 3];
```

Promises & async/await (brief)

```
async function fetchData(url) {  
  const res = await fetch(url);  
  return res.json();  
}
```

4. DOM basics (Browser JS)

- document.querySelector() to find elements
- element.addEventListener('click', handler) to respond to events
- element.textContent / innerHTML to change content

Example: change text when a button is clicked

```
<button id="myBtn">Click</button>
<p id="output">...</p>
<script>
  document.getElementById('myBtn').addEventListener('click', () => {
    document.getElementById('output').textContent = 'Button clicked!';
  });
</script>
```

5. Small project: To-Do list (minimal example)

```
<!-- index.html -->
<input id="task" placeholder="New task" />
<button id="add">Add</button>
<ul id="list"></ul>

<script>
const input = document.getElementById('task');
const addBtn = document.getElementById('add');
const list = document.getElementById('list');

addBtn.addEventListener('click', () => {
  const text = input.value.trim();
  if (!text) return;
  const li = document.createElement('li');
  li.textContent = text;
  li.addEventListener('click', () => li.classList.toggle('done'));
  list.appendChild(li);
  input.value = '';
});
</script>
```

6. Next steps & resources

- Learn ES6+ features (destructuring, modules)
- Practice by building small projects (calculator, notes app)
- Resources: MDN Web Docs, freeCodeCamp, JavaScript.info

Document 2 — JobSync MERN Stack Project Overview

Title: JobSync — MERN Stack Project Overview

Author: [Your Name]

Date: 28 October 2025

Table of contents

1. Project summary
 2. Features list
 3. Architecture diagram (textual)
 4. Data models (MongoDB schemas)
 5. Apply System: resume upload (Cloudinary) — implementation notes
 6. Deployment checklist
 7. Lessons learned & next features
-

1. Project summary

JobSync is a job-portal web application built with the MERN stack (MongoDB, Express, React, Node). It allows companies to post jobs and candidates to search and apply with resumes.

2. Features

- User authentication (JWT)
 - Role-based access (candidate, recruiter)
 - Job search with filters (location, experience, skills)
 - Apply system: upload resume, cover letter
 - Admin dashboard for job moderation
-

3. Architecture (textual)

- Client: React app (React Router, Context/Redux)
- Server: Node.js + Express API
- Database: MongoDB (Atlas)
- File storage: Cloudinary for resume uploads
- Authentication: JWT, refresh tokens

Sequence: React → HTTP request to Express → Auth check → DB read/write → Response

4. Data models (examples)

User schema (simplified)

```
const UserSchema = new mongoose.Schema({
  name: String,
  email: { type: String, unique: true },
  passwordHash: String,
  role: { type: String, enum: ['candidate', 'recruiter', 'admin'], default:
'candidate' }
});
```

Job schema (simplified)

```
const JobSchema = new mongoose.Schema({
  title: String,
  company: String,
  location: String,
  skills: [String],
  salaryRange: { min: Number, max: Number },
  postedBy: mongoose.Types.ObjectId,
  createdAt: Date
});
```

5. Apply System (Cloudinary) — Implementation notes

- Frontend: file input, send FormData to server
- Backend: receive FormData, forward file to Cloudinary SDK, store returned URL and public_id in DB
- Security: validate file type (pdf, docx), max size 5MB

Example Express route

```
app.post('/apply', authMiddleware, upload.single('resume'), async (req, res) =>
{
  // upload buffer to Cloudinary
  const result = await cloudinary.uploader.upload_stream({ resource_type:
'raw' }, (err, data) => { ... });
});
```

6. Deployment checklist

- Configure environment variables (DB URI, Cloudinary keys)
- Use PM2 or systemd for Node process
- Host frontend on Netlify / Vercel / S3+CloudFront
- Use HTTPS, configure CORS, set security headers

7. Lessons & next features

- Add in-app PDF preview, multi-resume support, analytics dashboard, better search indexing (ElasticSearch)

Document 3 — How I Improved Focus: Deep Work Techniques

Title: How I Improved Focus — Practical Deep Work Techniques

Author: [Your Name]

Date: 28 October 2025

Table of contents

1. Why focus matters
2. Daily routine template
3. 7 practical techniques
4. Tools & apps
5. 30-day focused plan

1. Why focus matters

Deep, uninterrupted work produces higher-quality results in less time. Shallow, fragmented work increases stress and reduces output.

2. Daily routine (example)

- 06:00 — Wake, morning hygiene
 - 06:30 — 45-minute focused session (no phone)
 - 07:30 — Breakfast
 - 09:00 — 2-hour deep work block (highest priority)
 - 12:00 — Break / walk
 - 14:00 — 90-minute secondary block
 - 17:00 — Review tasks, plan next day
-

3. 7 practical techniques

1. **Time blocking:** Schedule specific blocks for focused work. Protect them in calendar.
 2. **Pomodoro variant:** 52/17 or 90/20 works well for deep tasks.
 3. **Single-tasking:** Keep one main objective per block.
 4. **Reduce friction:** Remove distracting tabs, use website blockers (Cold Turkey, LeechBlock).
 5. **Task batching:** Group related small tasks into a single slot.
 6. **Environmental cues:** Use consistent workspace, headphones, or a plant to signal focus.
 7. **Digital minimalism:** Turn off non-essential notifications; schedule email checks twice daily.
-

4. Tools & apps

- Notion / Obsidian for notes and project outlines
 - Todoist / Google Tasks for daily tasks
 - Focus apps: Forest (phone), Focus To-Do, Cold Turkey
-

5. 30-day focused plan (simple)

Week 1: time-block two daily deep sessions (30–45 min).

Week 2: increase one session to 90 minutes.

Week 3: measure output, refine schedule.

Week 4: add accountability (public report or partner).

Document 4 — OBS Studio: Step-by-Step Webinar Recording Guide

Title: OBS Studio — Record a Webinar in Chrome (Video + Audio)

Author: [Your Name]

Table of contents

1. Goal and assumptions
 2. Quick checklist
 3. Full setup steps (windows)
 4. Fixes for black screen
 5. Audio routing and testing
 6. Final recording settings
-

1. Goal and assumptions

Goal: Record a webinar playing in a Chrome tab (video + system audio) on Windows 10/11. Assumes OBS Studio installed and Chrome available.

2. Quick checklist

- Add Display Capture source
 - Desktop Audio set to correct device
 - Turn off Chrome hardware acceleration
 - Run OBS on integrated GPU if black screen occurs
-

3. Full setup steps

1. Open Chrome → Settings → System → turn off **Use hardware acceleration** and relaunch Chrome.
 2. In OBS → Sources → **+** → **Display Capture** → select the monitor showing the webinar.
 3. Settings → Audio → Desktop Audio → Default (or explicitly select your speakers/headphones).
 4. Add Mic (Audio Input Capture) if you want narration.
 5. Test: play webinar → check Desktop Audio meter.
 6. Start Recording → stop → check saved file.
-

4. Fixes for black screen

- Run OBS as Administrator.
 - Go to Windows Settings → Graphics Settings → add obs64.exe → set to **Power saving** (integrated GPU).
 - Use Display Capture (reliable) rather than Window Capture if DRM or GPU isolation blocks capture.
-

5. Audio routing & testing

- If you use a headphone + mic combo, ensure Windows default output matches OBS desktop audio.
 - Optional: use VoiceMeeter to create virtual inputs and route app audio separately for advanced mixing.
-

6. Final recording settings (recommended for webinars)

- Output → Recording Format: `mkv` (or `mp4` if you need direct playback)
 - Encoder: `x264` or `NVENC` if available
 - FPS: 30
 - Bitrate: 2500–4000 kbps
-

Document 5 — Stoic Habits for Daily Confidence

Title: Stoic Habits for Daily Confidence — Practical Micro-Rituals

Author: [Your Name]

Date: 28 October 2025

Table of contents

1. What is Stoicism (short)
 2. Daily micro-rituals (10 practical habits)
 3. How to practice in idle moments
 4. Weekly reflection template
-

1. What is Stoicism?

Stoicism is a practical philosophy focused on controlling what you can, accepting what you cannot, and living in accordance with reason and virtue. It's highly action-oriented and suitable for building steady confidence.

2. Daily micro-rituals (10)

1. **Morning intention (1 minute):** State one aim for the day.
2. **Premeditatio malorum (2 minutes):** Visualize small setbacks and how you'll respond.

3. **Negative visualization before hard tasks:** Briefly imagine things going wrong to reduce surprise.
 4. **Focus on process, not outcome:** Emphasize consistent effort over praise.
 5. **Gratitude snapshot (30s):** Note three small things you're grateful for.
 6. **Pause before reacting:** Count to 5 before responding to criticism.
 7. **Evening reflection (5 minutes):** What went well? What could be done better?
 8. **Physical posture check:** Stand straight for one minute to reset confidence.
 9. **Micro journaling:** Two sentences about progress.
 10. **Practice active kindness:** small helpful action for someone daily.
-

3. How to practice in idle moments

- While waiting for a meeting, run a quick breathing exercise.
 - Turn small queues (elevators, lines) into 30s moments for a calm reframe.
-

4. Weekly reflection template

- Wins this week (3):
 - Lessons (2):
 - One experiment for next week: (e.g., try working with no phone for a 90-minute deep session)
-

End of document collection.