

Web Dev Cheatsheet

HTML · CSS · JavaScript · Node.js · MongoDB

1

HTML — Estrutura & Semântica

1Estrutura Base (Emmet: ! + Tab)

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width,
      initial-scale=1.0">
  <title>T tulo</title>
</head>
<body>
</body>
</html>
```

1Tags Semânticas HTML5

Tag	Uso
<header>	Cabeçalho da página
<nav>	Menu de navegação
<main>	Conteúdo principal
<section>	Secção temática
<article>	Conteúdo independente
<aside>	Barra lateral
<footer>	Rodapé
<figure>	Imagem com legenda

1Emmet — Atalhos HTML

Emmet	Resultado
!	Estrutura HTML completa
div.classe	<div class="classe">
ul>li*5	ul com 5 li's
a[href=#]	
p{Texto}	<p>Texto</p>
input:email	input type='email'
nav>ul>li*4>a	Nav com 4 links
h*3{T \$}	h1, h2, h3 com texto

1Formulário

```
<form action="/enviar" method="POST">
  <label for="nome">Nome:</label>
  <input type="text" id="nome"
    name="nome" required>

  <label for="email">Email:</label>
  <input type="email" id="email"
    name="email" required>

  <button
    type="submit">Enviar</button>
</form>
```

1Imagem Responsiva

```
<picture>
  <source media="(min-width:800px)"
    srcset="grande.webp">
  
</picture>
```

Usa sempre loading="lazy" para melhor performance.

1

CSS — Estilo & Layout

1Variáveis CSS

```
:root {
  --cor-primaria: #3b82f6;
  --cor-texto: #1e293b;
  --espacamento: 1rem;
  --radius: 8px;
}
.botao {
  background: var(--cor-primaria);
  padding: var(--espacamento);
  border-radius: var(--radius);
}
```

1Flexbox

```
.container {
  display: flex;
  justify-content: center; /* eixo X */
  align-items: center; /* eixo Y */
  flex-wrap: wrap;
  gap: 1rem;
}
.item { flex: 1 1 200px; }
```

gap substitui margins entre itens flex.

1CSS Grid

```
.grid {
  display: grid;
  grid-template-columns:
    repeat(auto-fill, minmax(250px,
      1fr));
  gap: 1.5rem;
}
/* 3 colunas fixas */
.grid-3 {
  grid-template-columns: repeat(3,
    1fr);
}
```

1Media Queries (Mobile First)

```
.card { width: 100%; }
```

```
@media (min-width: 768px) {
  .card { width: 48%; }
}
@media (min-width: 1024px) {
  .card { width: 32%; }
}
```

480px	Telemóvel
768px	Tablet
1024px	Desktop
1280px	Ecrã grande

1Animações & Transições

```
.btn {
  transition: all 0.3s ease;
}
.btn:hover {
  transform: translateY(-3px);
  box-shadow: 0 10px 20px
    rgba(0,0,0,.2);
}
@keyframes fadeIn {
  from { opacity: 0; transform:
    translateY(20px); }
  to { opacity: 1; transform:
    translateY(0); }
}
.el { animation: fadeIn 0.5s ease
  forwards; }
```

1Seletores Avançados

Seletor	Significado
p:first-child	Primeiro filho
li:nth-child(2n)	Elementos pares
a:not(.ativo)	Excluir classe
input:focus	Quando focado
div > p	Filho direto
h2 + p	Irmão adjacente
[data-x="y"]	Por atributo
::before	Pseudo-elemento

1Reset Moderno

```
*, ::before, ::after {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}
```

1

JavaScript — Lógica & DOM

1Selecionar Elementos

```
const el =
  document.querySelector('#id');
const els =
  document.querySelectorAll('.cls');

// Iterar
els.forEach(el =>
  el.classList.add('ativo'));

// Manipular
el.textContent = 'Novo texto';
el.innerHTML =
  '<strong>Bold</strong>';
el.style.color = 'red';
```

1Eventos

```
btn.addEventListener('click', (e) => {
  e.preventDefault();
  console.log('Clicado!');
});

// Delega o (eficiente)
document.addEventListener('click',
  (e) => {
    if (e.target.matches('.btn')) {
      // l gica aqui
    }
  });
```



```
body: JSON.stringify({ nome: 'Ana'
  });
```

1Desestruturação & Spread

```
// Arrays
const [a, b, ...resto] = [1, 2, 3, 4];

// Objetos
const { nome, idade = 18 } =
  utilizador;

// Spread
const novo = { ...obj1, ...obj2 };
const arr = [...arr1, ...arr2];

// Default params
function saudar(nome = 'Mundo') {
  return '01 , ${nome}!';
}
```

1Array — Métodos Essenciais

```
const nums = [1, 2, 3, 4, 5];

nums.map(n => n * 2); //
[2,4,6,8,10]
nums.filter(n => n > 2); // [3,4,5]
nums.reduce((a,n) => a+n, 0); // 15
nums.find(n => n > 3); // 4
nums.some(n => n > 4); // true
nums.every(n => n > 0); // true
nums.includes(3); // true
[...new Set(nums)]; // sem
duplicados
```

1LocalStorage

```
// Guardar
localStorage.setItem('user',
  JSON.stringify({ nome: 'Ana' }));

// Ler
const user = JSON.parse(
  localStorage.getItem('user'));

// Remover
localStorage.removeItem('user');
localStorage.clear();
```

Nunca guardes senhas no localStorage!

1

Node.js — Servidor & Backend

1Servidor Express Básico

```
const express = require('express');
const app = express();

app.use(express.json());

app.get('/', (req, res) => {
  res.json({ mensagem: '01 !' });
});

app.listen(3000, () => {
  console.log('http://localhost:3000');
});
```

1Estrutura de Projeto

```
projeto/
  src/
    routes/      # rotas
  da API
    models/      #
  modelos MongoDB
    controllers/ #
  l gica neg cio
    middleware/  # auth,
  etc.
    .env
    .gitignore
    package.json
    server.js
```

Instala: npm init -y → npm i express mongoose dotenv

1Variáveis de Ambiente

```
# .env
PORT=3000
MONGODB_URI=mongodb://localhost:27017/bd
JWT_SECRET=segredo_super_secreto
```

```
require('dotenv').config();
const port = process.env.PORT || 3000;
const uri = process.env.MONGODB_URI;
```

Adiciona .env ao .gitignore sempre!

1CRUD com Express Router

```
const router =
  require('express').Router();

router.get('/', listarTodos);
router.get('/:id', obterUm);
router.post('/', criar);
router.put('/:id', atualizar);
router.delete('/:id', eliminar);

module.exports = router;

// server.js
app.use('/api/utilizadores', router);
```

1

MongoDB — Base de Dados NoSQL

1Ligação com Mongoose

```
const mongoose = require('mongoose');

async function ligar() {
  try {
    await mongoose.connect(
      process.env.MONGODB_URI);
    console.log('MongoDB ligado!');
  } catch (err) {
    console.error(err);
    process.exit(1);
  }
}
ligar();
```

1Schema & Model

```
const { Schema, model } =
  require('mongoose');

const schema = new Schema({
  nome: { type: String, required:
    true },
  email: { type: String, required:
    true,
    unique: true, lowercase:
    true },
  idade: { type: Number, min: 0 },
  ativo: { type: Boolean, default:
    true }
}, { timestamps: true });

module.exports = model('User',
  schema);
```

1CRUD — Operações

```
const U = require('./models/User');

// CREATE
await U.create({ nome: 'Ana',
  email: 'ana@mail.pt'
});

// READ
await U.find({ ativo: true });
await U.findById(id);

// UPDATE
await U.findByIdAndUpdate(id,
  { nome: 'Ana Silva' }, { new: true
  });

// DELETE
await U.findByIdAndDelete(id);
```

1Queries Avançadas

```
// Filtros
await U.find({ idade: { $gte: 18 } });
await U.find({ nome: /ana/i });

// Ordenar, limitar, paginar
await U.find()
  .sort({ nome: 1 }) // i=ASC
  -i=DESC
  .limit(10)
  .skip(20)
  .select('nome email');

// Contar
const total = await
  U.countDocuments();
```

1Manual — Ligação Passo a Passo

1. Criar projeto: npm init -y
2. Instalar: npm i express mongoose dotenv
3. MongoDB Atlas: criar cluster M0 gratuito em mongodb.com/atlas
4. .env: colar a connection string
5. server.js: require('dotenv').config() + mongoose.connect(...)
6. Testar: npm run dev + Postman/Thunder Client

1

Atalhos de Teclado & Formatações Rápidas

1VS Code — Edição de Código

Atalho	Ação
Ctrl+/ Alt+↑/↓	Comentar/descomentar linha
Alt+Shift+↑/↓	Mover linha
Alt+Shift+↑/↓	Duplicar linha
Ctrl+Shift+K	Apagar linha inteira
Ctrl+Enter	Nova linha abaixo
Ctrl+Shift+Enter	Nova linha acima
Ctrl+Z / Ctrl+Y	Desfazer / Refazer
Ctrl+X / Ctrl+C	Cortar/Copiar linha

1VS Code — Navegação & Pesquisa

Atalho	Ação
Ctrl+P	Abrir ficheiro rapidamente
Ctrl+Shift+P	Paleta de comandos
Ctrl+F	Pesquisar no ficheiro
Ctrl+H	Substituir no ficheiro
Ctrl+Shift+F	Pesquisar em todos os ficheiros
Ctrl+G	Ir para linha número
Ctrl+Tab	Alternar ficheiros abertos
Ctrl+\	Dividir editor

1VS Code — Multi-cursor & Seleção

Atalho	Ação
Ctrl+D	Selecionar próxima ocorrência
Ctrl+Shift+L	Selecionar TODAS as ocorrências
Alt+Click	Adicionar cursor extra
Ctrl+Alt+↑/↓	Cursor acima/abaixo
Shift+Alt+drag	Seleção em coluna
Ctrl+L	Selecionar linha inteira
Ctrl+Shift+[Fechar/Abrir bloco

1VS Code — Produtividade

Atalho	Ação
Ctrl+‘	Terminal integrado
Ctrl+B	Mostrar/esconder sidebar
Alt+Shift+F	Formatar documento
F2	Renomear símbolo
F12	Ir para definição
Alt+F12	Pré-visualizar definição
Ctrl+Shift+V	Pré-visualizar Markdown
Ctrl+K Z	Modo Zen (ecrã cheio)

1Emmet — HTML Rápido

Emmet	Resultado
!	HTML completo
nav>ul>li*4>a	Nav com 4 links
div.container>div.card*3	3 cards
table>tr*3>td*4	Tabela 3×4
ul>li.item\$*5	Lista item1..item5
section#hero>h1+p+a.btn	Secção hero

1Emmet — CSS Rápido

Abrev.	Propriedade
df	display: flex
dg	display: grid
jcc	justify-content: center
aic	align-items: center
m:a	margin: auto
p10	padding: 10px
w100p	width: 100%
bgc	background-color
tac	text-align: center
pos:r	position: relative

1Browser DevTools (F12)

Atalho	Ação
F12	Abrir DevTools
Ctrl+Shift+I	Abrir DevTools (alt.)
Ctrl+Shift+C	Inspecionar elemento
Ctrl+Shift+J	Consola JavaScript
Ctrl+Shift+M	Modo responsivo
Ctrl+R	Recarregar página
Ctrl+Shift+R	Recarregar sem cache
Ctrl+U	Ver código fonte

1Terminal / Linha de Comandos

Comando	Ação
Ctrl+C	Parar processo
Ctrl+L	Limpar terminal
↑ / ↓	Histórico de comandos
Tab	Auto-completar
Ctrl+A	Início da linha
Ctrl+E	Fim da linha
npm run dev	Iniciar dev server
node -e "..."	Executar JS direto