

JavaScript + Web Development Study Guide

■ VARIABLES & BASIC TYPES

- string: `let name = 'Marjorie'`
- number: `let age = 19`
- boolean: `let isStudent = true`
- `const` = cannot be reassigned, `let` = assignable

■ ARRAYS

- `let fruits = ['apple', 'banana']`
- Access: `fruits[0]`
- Length: `fruits.length`

■ OBJECTS

- `let user = { name: 'Bloom', age: 19 }`
- Access: `user.name` or `user['name']`

■ FUNCTIONS

- Regular function: `function add(a, b) { return a + b }`
- Arrow function: `const greet = (name) => `Hello, ${name}``

■ CONDITIONALS & LOGIC

- if/else, switch statements
- Comparison: `===`, `!==`, `>`, `<`
- Logical: `&&`, `||`, `!`

■ LOOPS & ARRAY METHODS

- for, while
- `Array.map()`: transform each item
- `Array.filter()`: keep some items
- `Array.find()`: find first match

■ DESTRUCTURING

- `const { name, age } = user`
- `const [first, second] = fruits`

■ TEMPLATE LITERALS

- `let msg = `Hello ${name}, age ${age}``

■ MODULES (IMPORT/EXPORT)

- `export function add(a, b) { return a + b }`
- `import { add } from './math.js'`

■ REACT + NEXT.JS WEB DEV

■ COMPONENTS

- A component is just a function returning JSX:

```
function Navbar() { return <nav>Bloom</nav> }
```

■ NEXT.JS LINK

- Use Link instead of <a> for client-side navigation:

```
import Link from 'next/link'  
<Link href='/about'>About</Link>
```

■ PROPS

- Props = inputs to a component (plain objects):

```
function Button({ label, onClick }) {  
  return <button onClick={onClick}>{label}</button>  
}
```

■ CHILDREN

- Special prop for nested JSX:

```
function Card({ children }) {  
  return <div className='card'>{children}</div>  
}  
<Card><h2>Title</h2><p>Body</p></Card>
```

■ STATE

- useState remembers values inside components:

```
import { useState } from 'react'  
const [count, setCount] = useState(0)  
<button onClick={() => setCount(count + 1)}>Count: {count}</button>
```

■ EVENTS

- Event handlers in React:

```
function SearchBar() {  
  const handleChange = (e) => {  
    console.log(e.target.value)  
  }  
  return <input onChange={handleChange} />  
}
```

■ RENDERING LISTS WITH map()

- map arrays to JSX, always give a key:

```
const items = ['apple','banana']  
<ul>  
  {items.map((fruit, i) => <li key={i}>{fruit}</li>)}  
</ul>
```

■ ASYNC DATA FETCH

- Fetch data with async/await:

```
async function getArticles() {
```

```
const res = await fetch('https://example.com/rss')
const data = await res.json()
console.log(data)
}
```

■ KEY TAKEAWAYS

- JavaScript powers React and Next.js.
- Props and state give interactivity.
- `map()` is essential for rendering lists of data.
- `fetch()` loads external data for dynamic pages.
- React components = plain JS functions returning JSX.