**Assignment – 5**

**Question 1: Difference between append() and appendChild()**

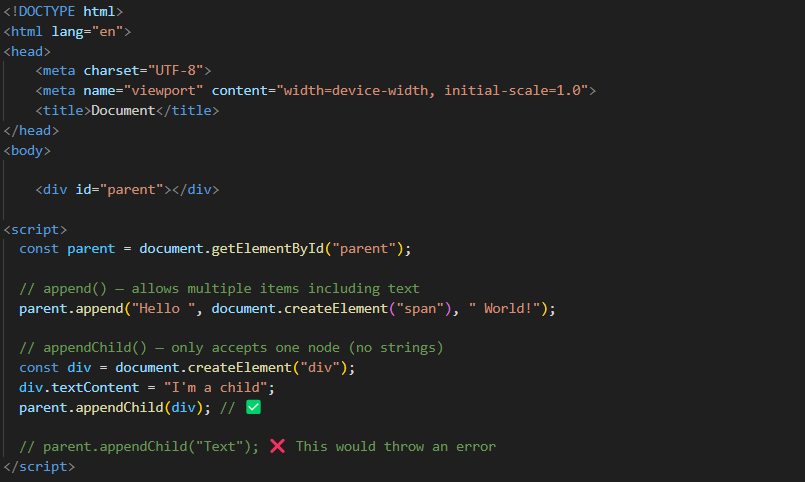
**append():**

* Can append multiple nodes or text.
* Can append strings.
* Not supported in older IE versions.

**appendChild():**

* Can append only one node (no strings).
* Returns the appended node.
* Works in all major browsers.

**Code:**

****

**Output:**

****

**Question 2: Difference between event.target and event.currentTarget**

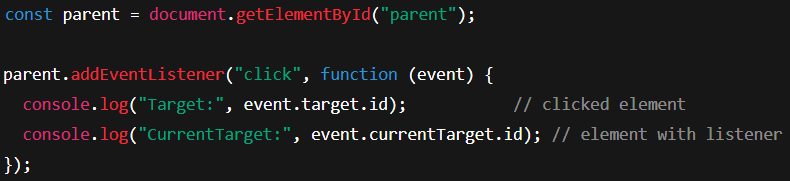
**event.target:**

The actual element that triggered the event.

**event.currentTarget:**

The element the event listener is attached to.

**Code :**

****

**Output:**

**When Click on the Button**

****

**When Click on the outer <div> (not on the button)**

****

**Question 3: How to perform debouncing by Set Interval?**

(Not recommended method – setTimeout() is preferred, but this shows concept.)

Used to mimic debounce behaviour.

Works by setting an interval that runs once, then clears itself.

Function is executed only after the user pauses input for a fixed delay.

**Code:**

****

**Key Points:**

Function runs only once after delay if no new input.

Clears the interval after executing the function.

Less accurate than setTimeout because interval runs repeatedly.

Useful for understanding timer behaviour, not suitable for production.

**Question 4: Difference between <script>, async, and defer**

**1. <script> – Regular Script**

* The browser **runs the script right away** when it finds it during page loading.
* It **pauses the HTML rendering** until the script finishes downloading and running.
* This can **slow down page load time**, especially for big scripts.
* Since it's run immediately, **script order matters** based on how they’re placed.
* Best for small, essential scripts that need to run upfront.

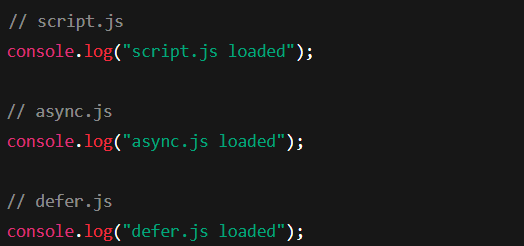
**2. async – Load & Run Independently**

* Script starts downloading **while the page is still being built**.
* As soon as the download completes, it **runs immediately**, **even if HTML isn’t done yet**.
* It doesn’t block the page, but if you use multiple async scripts, they might **not run in the same order**.
* Ideal for things like **ads, trackers, or external tools** that don’t rely on the rest of the page.

**3. defer – Wait Until Page is Ready**

* Script begins downloading during page load but **waits to run until the HTML is fully processed**.
* Multiple defer scripts are **executed in order**, just like they appear in the code.
* It doesn’t interrupt the page loading and **ensures the DOM is ready** before scripts run.
* Great choice for scripts that **interact with page content** (like DOM manipulation).

**Code:**



**Output:**

****