Summary of JavaScript Tag Name Manipulation

This document explains how to manipulate all elements of a specific tag type using `document.getElementsByTagName`:

1. **Purpose**: `document.getElementsByTagName(”tag”)` targets all elements of a tag type (e.g., <p>) for batch manipulation, avoiding the need for individual IDs.

2. **Collection Creation**: Returns an array-like collection, e.g., `var par = document.getE` for all paragraphs.

3. **Accessing Elements**: Use array-like indexing (e.g., `par[0]` for the first paragraph) and `par.length` for the number of elements.

4. **Reading and Setting Content**:

   - Read: `par[1].innerHTML` gets the second paragraph’s content (e.g., ”This bed is too big”).

   - Set: `par[1].innerHTML = ”This SUV is too big”` updates the content.

5. **Batch Styling**: A loop applies styles, e.g., `for (var i = 0; i < par.length; i++) { par[i].style.fontFamily = ”Verdana, Geneva, sans-serif”; }` sets the font for all paragraphs.

6. **Other Tags**: Works for other tags, e.g., `document.getElementsByTagName(”div”)` or `document.getElementsByTagName(”ul”)`.

7. **Context**: Inline event handling is implied, with professional methods to be covered later.