/\*

\* Get a DOM element from a class name:

\*/

export const getNode = (className: string): HTMLElement => {

const selector = `.${className}`;

return <HTMLElement>document.querySelector(selector);

};

export const stringToNode = (html: string): HTMLElement => {

let wrapper: HTMLElement = document.createElement('div');

wrapper.innerHTML = html.trim();

return <HTMLElement>wrapper.firstChild;

};

export const insertAfter = (newNode: Node, referenceNode: Node) => {

let nextNode = referenceNode.nextSibling;

let parentNode = referenceNode.parentNode;

parentNode.insertBefore(newNode, nextNode);

};

export const removeNode = (node: Node) => {

node.parentElement.removeChild(node);

};

export const throwErr = (message: string) => {

// Remove multiple spaces:

message = message.replace(/ +(?= )/g,'');

message = message.trim();

throw `SweetAlert: ${message}`;

};

/\*

\* Match plain objects ({}) but NOT null

\*/

export const isPlainObject = (value: any): boolean => {

if (Object.prototype.toString.call(value) !== '[object Object]') {

return false;

} else {

var prototype = Object.getPrototypeOf(value);

return prototype === null || prototype === Object.prototype;

}

};

/\*

\* Take a number and return a version with ordinal suffix

\* Example: 1 => 1st

\*/

export const ordinalSuffixOf = (num: number): string => {

let j = num % 10;

let k = num % 100;

if (j === 1 && k !== 11) {

return `${num}st`;

}

if (j === 2 && k !== 12) {

return `${num}nd`;

}

if (j === 3 && k !== 13) {

return `${num}rd`;

}

return `${num}th`;

};