```
/*
Pta.js
Programming Textarea
(C) Michael D Leidel, 2016
All Rights Reserved.
If you use this library it
must display this comment block.
-------------------------------
After the library is loaded..
activate with: Pta.listeners.initialize(textarea id);
Find functions:
Pta.findr.replaceOne(textarea id, text to find, replace text)
Pta.insClip(textarea id, text to insert)
*/

var Pta = Pta || {};
Pta.listeners = {initialize:function(sid) {
    var Oid = document.getElementById(sid);
    Pta.listeners.setTabs(Oid);
    Oid.style.tabSize = 4;
    Pta.listeners.setIndent(Oid);
    Pta.listeners.setJoinLines(Oid);
    Pta.listeners.setMatching(Oid);
    Pta.listeners.setDupChr(Oid);
    Pta.listeners.setZen(Oid);
    Pta.listeners.setLineNbr(Oid);
}, setTabs:function(TAo) {
    TAo.addEventListener("keydown", function(event) {
        if (event.keyCode === 9 && !event.shiftKey) {
            event.preventDefault();
            var p1, p2, txt, sels, sele;
            sels = TAo.selectionStart;
            sele = TAo.selectionEnd;
            txt = TAo.value;
            if (sels === sele) {
                TAo.value = txt.slice(0, sels) + "\t" + txt.slice(sele);
                TAo.selectionStart = sels + 1;
                TAo.selectionEnd = sels + 1;
                return;
            }
            var stxt = txt.slice(sels, sele);
            var nl = countNewLines(stxt);
            if (txt.charCodeAt(sele - 1) !== 10) {
                var i1, i2;
                var len = stxt.length;
                while (1) {
                    var si = TAo.selectionEnd;
                    i1 = txt.indexOf(stxt, si);
                    i2 = i1 + len;
                    if (i1 >= 0) {
                        TAo.selectionStart = i1;
                        TAo.selectionEnd = i2;
                    } else {
                        if (si > 0) {
                            TAo.selectionStart = 0;
                            TAo.selectionEnd = 0;
                            continue;
                        }
                    }
                    return;
                }
            }
            stxt = stxt.replace(/\s+$/g, "");
            stxt = stxt.replace(new RegExp("\n", "g"), "\n\t");
            p1 = txt.slice(0, sels);
            p2 = txt.slice(sele);
            TAo.value = p1 + "\t" + stxt + "\n" + p2;
            TAo.selectionStart = sels;
            TAo.selectionEnd = sele + nl;
            return;
        }
        function countNewLines(txt) {
            var inx, c = 0;
            for (inx = 0;inx < txt.length; inx+=1) {
                if (txt.charAt(inx) === "\n") {
                    c+=1;
                }
            }
            return c;
        }
    });
    TAo.addEventListener("keydown", function(event) {
        if (event.keyCode === 9 && event.shiftKey) {
            event.preventDefault();
            var p1, p2, txt, sels, sele;
            sels = TAo.selectionStart;
            sele = TAo.selectionEnd;
            txt = TAo.value;
            if (sels === sele) {
                p1 = txt.slice(0, sels - 1);
                p2 = txt.slice(sels);
                TAo.value = p1 + p2;
                TAo.selectionStart = sels - 1;
                TAo.selectionEnd = sels - 1;
                return;
            }
            var stxt = txt.slice(sels, sele);
            var nl = countNewLines(stxt);
            if (stxt.charCodeAt(0) > 32) {
                return;
            }
            stxt = stxt.replace(new RegExp("\n\t", "g"), "\n");
            p1 = txt.slice(0, sels);
            p2 = txt.slice(sele);
            TAo.value = p1 + stxt.slice(1) + p2;
            TAo.selectionStart = sels;
```

```
                TAo.selectionEnd = sele - nl;
                return;
            }
            function countNewLines(txt) {
                var inx, c = 0;
                for (inx = 0; inx < txt.length; inx+=1) {
                    if (txt.charAt(inx) === "\n") {
                        c+=1;
                    }
                }
                return c;
            }
        });
    }, setJoinLines:function(TAo) {
        TAo.addEventListener("keydown", function(event) {
            if (String.fromCharCode(event.which).toLowerCase() === "t" && event.altKey) {
                event.preventDefault();
                var p1, p2, sels, sele, txt, slines;
                sels = TAo.selectionStart;
                sele = TAo.selectionEnd;
                txt = TAo.value;
                slines = txt.slice(sels, sele);
                if (slines === "") {
                    alert("No Lines Selected to Join");
                    return;
                }
                p1 = txt.slice(0, sels);
                p2 = txt.slice(sele);
                slines = slines.replace(/\n/g, "");
                txt = p1 + slines + p2;
                TAo.value = txt;
                TAo.selectionEnd = sels;
                return;
            }
        });
    }, setIndent:function(TAo) {
        TAo.addEventListener("keydown", function(event) {
            if (event.keyCode === 13 && !event.ctrlKey) {
                event.preventDefault();
                var inx, stx, chx, v = TAo.value, s = TAo.selectionStart, e = TAo.selectionEnd;
                for (inx = s - 1; inx > 0; inx-=1) {
                    if (v.charCodeAt(inx) === 10) {
                        break;
                    }
                }
                stx = String.fromCharCode(10);
                if (inx > 0) {
                    inx+=1;
                }
                chx = v.charCodeAt(inx);
                while (chx === 32 || chx === 9) {
                    stx += String.fromCharCode(chx);
                    inx+=1;
                    chx = v.charCodeAt(inx);
                }
                TAo.value = v.slice(0, s) + stx + v.slice(e);
                TAo.selectionStart  = s + stx.length;
                TAo.selectionEnd = s + stx.length;
                return;
            }
        });
    }, setMatching:function(TAo) {
        TAo.addEventListener("keydown", function(event) {
            if (String.fromCharCode(event.which).toLowerCase() === "q" && event.altKey) {
                event.preventDefault();
                matchBrace();
            }
        });
        TAo.addEventListener("click", function(event) {
            if (event.ctrlKey) {
                event.preventDefault();
                matchBrace();
            }
        });

        function matchBrace() {
            var sta = TAo.value;
            var cpos = TAo.selectionStart;
            var ch = sta.charAt(cpos);
            if ("{(".indexOf(ch) < 0) {
                cpos-=1;
                ch = sta.charAt(cpos);
                if ("})".indexOf(ch) < 0) {
                    return;
                }
            }
            if (ch === "}") {
                doMatchBk("}", "{", cpos);
            } else {
                if (ch === "{") {
                    doMatchFw("{", "}", cpos);
                } else {
                    if (ch === ")") {
                        doMatchBk(")", "(", cpos);
                    } else {
                        if (ch === "(") {
                            doMatchFw("(", ")", cpos);
                        }
                    }
                }
            }
        }
        function doMatchBk(cb1, cb2, cpos) {
            var inx, cbx, cnt = 0;
            var sta = TAo.value;
            for (inx = cpos - 1; inx >= 0; inx-=1) {
```

```
            cbx = sta.charAt(inx);
            if (cbx === cb2) {
                if (cnt === 0) {
                    TAo.selectionStart = inx;
                    break;
                } else {
                    cnt-=1;
                }
            } else {
                if (cbx === cb1) {
                    cnt+=1;
                }
            }
        }
        if (inx < 0) {
            alert("Match not found");
        }
    }
    function doMatchFw(cb1, cb2, cpos) {
        var inx, cbx, len, cnt = 0;
        var sta = TAo.value;
        len = sta.length;
        for (inx = cpos + 1; inx < len; inx+=1) {
            cbx = sta.charAt(inx);
            if (cbx === cb2) {
                if (cnt === 0) {
                    TAo.selectionEnd = inx + 1;
                    break;
                } else {
                    cnt-=1;
                }
            } else {
                if (cbx === cb1) {
                    cnt+=1;
                }
            }
        }
        if (inx >= len) {
            alert("Match not found");
        }
    }
}, setDupChr:function(TAo) {
    TAo.addEventListener("keydown", function(event) {
        if (String.fromCharCode(event.which).toLowerCase() === "d" && event.altKey) {
            event.preventDefault();
            var inx, chx, v = TAo.value, s = TAo.selectionStart, e = TAo.selectionEnd;
            var ct = 0;
            for (inx = s - 1; inx > 0; inx-=1) {
                ct+=1;
                if (v.charCodeAt(inx) === 10) {
                    break;
                }
            }
            while (inx > 0) {
                inx-=1;
                if (v.charCodeAt(inx) === 10) {
                    break;
                }
            }
            if (v.charCodeAt(inx) !== 10) {
                ct-=1;
            }
            inx += ct;
            chx = v.charAt(inx);
            TAo.value = v.slice(0, s) + chx + v.slice(e);
            TAo.selectionStart = s + 1;
            TAo.selectionEnd = s + 1;
            return;
        }
    });
}, setZen:function(TAo) {
    var lastWrap = "";
    TAo.addEventListener("keydown", function(event) {
        if (String.fromCharCode(event.which).toLowerCase() === "z" && event.altKey) {
        zentag();
            event.preventDefault();
            return;
        }
    });
    TAo.addEventListener("keydown", function(event) {
        if (String.fromCharCode(event.which).toLowerCase() === "a" && event.altKey) {
            zentage(0);
            event.preventDefault();
            return;
        }
    });
    TAo.addEventListener("keydown", function(event) {
        if (String.fromCharCode(event.which).toLowerCase() === "w" && event.altKey) {
            zentage(1);
            event.preventDefault();
            return;
        }
    });

    function zentage(m) {
        var p1, p2, stag, t1, t2, sels, sele, txt, stxt;
        sels = TAo.selectionStart;
        sele = TAo.selectionEnd;
        txt = TAo.value;
        stxt = txt.slice(sels, sele);
        p1 = txt.slice(0, sels);
        p2 = txt.slice(sele);
        if (sels === sele) {
            alert("nothing selected");
            return;
        }
```

```
if (m === 0) {
    stag = prompt("Enter tag abbreviation\n or command");
    lastWrap = stag;
} else {
    stag = lastWrap;
}
if (stag === null) {
    return;
}
if (stag === "") {
    return;
}
if (stag === "/*") {
    t1 = "/* ";
    t2 = " */";
} else {
    if (stag === "<!") {
        t1 = "\x3c!-- ";
        t2 = " --\x3e";
    } else {
        if (stag === "ucase") {
            t1 = "";
            t2 = "";
            stxt = stxt.toUpperCase();
        } else {
            if (stag === "lcase") {
                t1 = "";
                t2 = "";
                stxt = stxt.toLowerCase();
            } else {
                if (stag === "\"") {
                    t1 = "\"";
                    t2 = "\"";
                } else {
                    if (stag === "'") {
                        t1 = "'";
                        t2 = "'";
                    } else {
                        if (stag === "{") {
                            t1 = "{";
                            t2 = "}";
                        } else {
                            t1 = "<" + stag + ">";
                            t2 = "</" + stag + ">";
                        }
                    }
                }
            }
        }
    }
}
txt = p1 + t1 + stxt + t2 + p2;
TAo.value = txt;
```

```
    TAo.selectionEnd = txt.length - p2.length;
    TAo.selectionStart = txt.length - p2.length;
    return;
}

function zentag() {
    var inx, p1, p2, sels, sele, txt, stag, cpos;
    sels = TAo.selectionStart;
    sele = TAo.selectionEnd;
    txt = TAo.value;
    stag = txt.slice(sels, sele);
    cpos = 0;   // for ^ position
    if (stag === "") {
        while (sele < txt.length) {
            if (txt.charCodeAt(sele) < 49) {
                break;
            }
            sele +=1;
        }
        while (sels > 0) {
            sels -= 1;
            if (txt.charCodeAt(sels) < 49) {
                sels += 1;
                break;
            }
        }
        stag = txt.slice(sels, sele);
    }
    if (stag === "") {
        alert("No Key Word to Match");
        return;
    }

    p1 = txt.slice(0, sels);
    p2 = txt.slice(sele);
    for (inx = 0; inx < atags.length; inx+=1) {
        if (stag === atags[inx].tag) {
            txt = p1 + atags[inx].tagx + p2;
            // try to set cursor at ^
            cpos = txt.indexOf("^", sels);
            if (cpos === -1) {
                TAo.value = txt;
                TAo.selectionEnd = txt.length - p2.length;
                TAo.selectionStart = txt.length - p2.length;
            } else {
                    p1 = txt.slice(0, cpos);
                    p2 = txt.slice(cpos+1);
                    txt = p1 + p2;
                    TAo.value = txt;
                    TAo.selectionEnd = cpos;
                    TAo.selectionStart = cpos;
                }
            return;
            }
        }

        // specific tag not found, use the word by the cursor

        txt = p1 + "<" + stag + "></" + stag + ">" + p2;
        cpos = txt.indexOf("><", sels);
        cpos+=1;
        TAo.value = txt;
        TAo.selectionEnd = cpos;
        TAo.selectionStart = cpos;
        return;
    }
}, setLineNbr:function(TAo) {
    var gSave4LnNbr = "";
    var sout;
    TAo.addEventListener("keydown", function(event) {
        if (String.fromCharCode(event.which).toLowerCase() === "n" && event.altKey) {
            event.preventDefault();
            if (gSave4LnNbr === "") {
                sout = "1\t\t";
                var lc = 2, inx, cc, slc;
                var txt = TAo.value;
                gSave4LnNbr = txt;
                for (inx = 0; inx < txt.length - 1; inx+=1) {
                    cc = txt.charAt(inx);
                    if (cc === "\n") {
                        slc = lc.toString();
                        while (slc.length < 4) {
                            slc += " ";
                        }
                        sout += cc + slc + "\t\t";
                        lc+=1;
                    } else {
                        sout += cc;
                    }
                }
                TAo.value = sout;
                TAo.readOnly=true;
            } else {
                TAo.readOnly=false;
                TAo.value = gSave4LnNbr;
                gSave4LnNbr = "";
            }
        }
    });
}
};  // END Pta.listeners

Pta.findr = {findText:function(Oid, targ) {
    if (arguments.length !== 2) {
        alert("missing arguments for findText");
        return;
        }
        var TAo = document.getElementById(Oid);
        var targ_id = document.getElementById(targ);
        var i1, i2;
        var stxt = TAo.value;
        var len = targ_id.value.length;
        var si = TAo.selectionEnd;
        i1 = stxt.indexOf(targ_id.value, si);
        i2 = i1 + len;
        if (i1 >= 0) {  // match found
            TAo.selectionStart = i1;
            TAo.selectionEnd = i2;
            TAo.focus();
            return;
        } else {
            if (si > 0) {   // back to top
                TAo.selectionStart = 0;
                TAo.selectionEnd = 0;
                Pta.findr.findText(Oid, targ);
            } else {
                return;
            }
        }
    }, replaceOne:function(Oid, targ, chng) {
        if (arguments.length !== 3) {
            alert("missing arguments for findText");
            return;
        }
        var TAo = document.getElementById(Oid);
        var chng_id = document.getElementById(chng);
        Pta.findr.findText(Oid, targ);
        var p1, p2, txt, sels, sele;
        txt = TAo.value;
        sels = TAo.selectionStart;
        sele = TAo.selectionEnd;
        if (sels === sele) {
            alert("No More to replace");
            return;
        }
        p1 = txt.slice(0, sels);
        p2 = txt.slice(sele);
        txt = p1 + chng_id.value + p2;
        TAo.value = txt;
        TAo.selectionStart = txt.length - p2.length;
        TAo.selectionEnd =  txt.length - p2.length;
        TAo.focus();
    }, replaceAll:function(Oid, targ, chng) {
        if (arguments.length !== 3) {
            alert("missing arguments for findText");
            return;
        }
        var TAo = document.getElementById(Oid);
        var targ_id = document.getElementById(targ);
    var chng_id = document.getElementById(chng);
    var txt = TAo.value;
    txt = txt.replace(new RegExp(targ_id.value, "g"), chng_id.value);
    TAo.value = txt;
    TAo.focus();
}, adjustScroll:function(Oid, lh) {
    var inx, c = 0, p = 0, txt;
    var TAo = document.getElementById(Oid);
    txt = TAo.value;
    p = TAo.selectionEnd;
    for (inx = 0;inx < p; inx+=1) {
        if (txt.charAt(inx) === "\n") {
            c+=1;
        }
    }
    TAo.scrollTop = c * lh;

}}; // END of Pta.findr

Pta.insClip =   function (Oid, itext) {
    var TAo = document.getElementById(Oid);
    var tav = TAo.value;
    var strPos = TAo.selectionStart;
    var front = tav.slice(0, strPos);
    var back = tav.slice(strPos);
    TAo.value = front + itext + back;
    TAo.selectionEnd = strPos + itext.length;
    TAo.focus();
};
```