# الگو:Hlist/styles.css

```
/* {{pp-protected|reason=match parent|small=yes}} */
/*
 * hlist styles are defined in core and Minerva and differ in Minerva. The
 * current definitions here (2023-01-01) are sufficient to override Minerva
 * without use of the hlist-separated class. The most problematic styles were
 * related to margin, padding, and the bullet. Check files listed at
 * [[MediaWiki talk:Common.css/to do#hlist-separated]]
 */
/*
 * TODO: When the majority of readership supports it (or some beautiful world
 * in which grade C support is above the minimum threshold), use :is()
 */
.hlist dl,
.hlist ol,
.hlist ul {
        margin: 0;
        padding: 0;
}

/* Display list items inline */
.hlist dd,
.hlist dt,
.hlist li {
        /*
         * don't trust the note that says margin doesn't work with inline
         * removing margin: 0 makes dds have margins again
         * We also want to reset margin-right in Minerva
         */
        margin: 0;
        display: inline;
}

/* Display requested top-level lists inline */
.hlist.inline,
.hlist.inline dl,
.hlist.inline ol,
.hlist.inline ul,
/* Display nested lists inline */
.hlist dl dl,
.hlist dl ol,
.hlist dl ul,
.hlist ol dl,
.hlist ol ol,
.hlist ol ul,
.hlist ul dl,
```

```
.hlist ul ol,
.hlist ul ul {
	display: inline;
}

/* Hide empty list items */
.hlist .mw-empty-li {
	display: none;
}

/* TODO: :not() can maybe be used here to remove the later rule. naive test
 * seems to work. more testing needed. like so:
 *.hlist dt:not(:last-child)::after {
 *	content: ": ";
 *}
 *.hlist dd:not(:last-child)::after,
 *.hlist li:not(:last-child)::after {
 *	content: " · ";
 *	font-weight: bold;
 *}
 */
/* Generate interpuncts */
.hlist dt::after {
	content: ": ";
}

.hlist dd::after,
.hlist li::after {
	content: " · ";
	font-weight: bold;
}

.hlist dd:last-child::after,
.hlist dt:last-child::after,
.hlist li:last-child::after {
	content: none;
}

/* Add parentheses around nested lists */
.hlist dd dd:first-child::before,
.hlist dd dt:first-child::before,
.hlist dd li:first-child::before,
.hlist dt dd:first-child::before,
.hlist dt dt:first-child::before,
.hlist dt li:first-child::before,
.hlist li dd:first-child::before,
.hlist li dt:first-child::before,
.hlist li li:first-child::before {
	content: " (";
	font-weight: normal;
}
```

```
.hlist dd dd:last-child::after,
.hlist dd dt:last-child::after,
.hlist dd li:last-child::after,
.hlist dt dd:last-child::after,
.hlist dt dt:last-child::after,
.hlist dt li:last-child::after,
.hlist li dd:last-child::after,
.hlist li dt:last-child::after,
.hlist li li:last-child::after {
        content: ")";
        font-weight: normal;
}

/* Put ordinals in front of ordered list items */
.hlist ol {
        counter-reset: listitem;
}

.hlist ol > li {
        counter-increment: listitem;
}

.hlist ol > li::before {
        content: " " counter(listitem) "\a0";
}

.hlist dd ol > li:first-child::before,
.hlist dt ol > li:first-child::before,
.hlist li ol > li:first-child::before {
        content: " (" counter(listitem) "\a0";
}
```