```
/***
    Template: MindTouch/Controls/SeeAlso
    Version:  2.0 (firnCore)
    Datum:    2026-07-09

    USAGE:

    SeeAlso(pages, exclude, limit, parents)
        Show pages related to current page by hierarchy and by see: and
keyword: tags.

    PARAMETERS:

    (optional) pages : list (default: child pages of current page)
        List of pages to organize based on their tags.

    (optional) exclude : list (default: none)
        List of pages to exclude from the related list.

    (optional) limit : num (default: 5)
        v2.0: Harte Obergrenze der ausgegebenen Links (vorher nur
Schwelle fuer Keyword-Suche).

    (optional) parents : bool (default: true)
        Show link to parent page if no other links are shown.

    CHANGELOG:

    2.0 --- firnCore customization ---
        - Hard Cap: Ausgabe strikt auf 'limit' Seiten begrenzt
        - Prioritaets-Scoring statt Alphabet:
            see:-Tag mit Definitionsseite = 100
            see:-Tag via tag.pages        = 80
            Child-Pages / uebergebene     = 60
            Keyword-Suchtreffer           = 30
            + 10 Punkte pro gemeinsamem Content-Tag mit der aktuellen
Seite
        - Fair-Trim: Round-Robin ueber die Kategorien
          (Topics, Tutorials, Troubleshooting, Reference, Misc),
          innerhalb jeder Kategorie nach Score absteigend
        - Kategorien jetzt disjunkt (Prioritaet: topic > tutorial >
troubleshooting > reference)
        - Tutorials nach Relevanz-Score sortiert statt nach
Schwierigkeitsgrad
***/

// Redirect to custom page, if any
var template_custom = wiki.inclusions(true)[-1].path .. 'Custom';
if (wiki.pageexists(template_custom)) {
```

```
    return template(template_custom, $);
}

var pages = $0 ?? $pages;
var exclude = $1 ?? $exclude ?? [ ];
var limit = $2 ?? $limit ?? 5;
var parents = $5 ?? $parents ?? true;

//--- firnCore customization: Content-Tags der aktuellen Seite fuer
Overlap-Scoring ---
var mytags = [
    string.tolower(tag.name)
    foreach
        var tag in page.tags where (tag.type == 'text') &&
!string.contains(tag.name, ':')
];

//--- Aggregate Pages ---
//--- firnCore customization: Reihenfolge = Prioritaet (see:-Tags vor
Children vor Keyword),
//--- Dedup behaelt das erste Vorkommen, 'base' traegt den Basis-Score ---

var candidates = [ ];

// 1) see:-Tags (oder Tags ohne Prefix): Definitionsseite oder verwandte
Seiten
let candidates ..= [
    p .. { base: (def && (def.id != page.id)) ? 100 : 80 }
    foreach
        var tag in page.tags where (tag.type === 'text') &&
(string.startswith(tag.name, 'see:', true) || !string.contains(tag.name,
':')),
        var def = tag.definition,
        var p in ((def && (def.id != page.id)) ? [ def ] : tag.pages)
where p.id != page.id
];

// 2) Child-Pages (falls keine Seitenliste uebergeben wurde) bzw.
uebergebene Seiten
if(pages is nil) {
    let candidates ..= [
        wiki.getpage(id) .. { base: 60 }
        foreach
            var id in [ num.cast(xml.text(id)) foreach var id in
wiki.tree(_, 1)["//@pageid"] ]
    ];
} else {
    let candidates ..= [ p .. { base: 60 } foreach var p in pages ];
}
```

```
// 3) keyword:-Tags -> Suchtreffer als Fuellmaterial, nur wenn noch Platz
ist
if(#candidates < limit) {
    var terms = [
        '"' .. string.escape(string.substr(tag.name, 8)) .. '"'
        foreach
            var tag in page.tags where (tag.type == 'text') &&
string.startswith(tag.name, 'keyword:')
    ];
    if(#terms) {
        var searchString = string.join(terms, ' ');
        var defaultSearch = 'type:wiki AND namespace:main';

        // find main guide (tagged article:topic-guide), if any
        var pathSearch = '';
        foreach (var parent in page.parents) {
            var isTopic = #[tag foreach var tag in parent.tags where
(tag.type == 'text' && string.endswith(tag.name, 'topic-guide'))] > 0;
            if (isTopic) {
                let pathSearch = ' AND path:' .. parent.path .. '/*';
                break;
            }
        }

        // keyword must appear in title
        var titleSearch = ' AND title:' .. searchString;
        var searchQuery = defaultSearch .. titleSearch .. pathSearch;

        let candidates ..= [
            p .. { base: 30 }
            foreach
                var p in wiki.getsearch(searchString, limit -
#candidates, _, searchQuery)
        ];
    }
}

// de-duplicate pages (erstes Vorkommen = hoechste Prioritaet)
var uniquepages = [ ];
var pageids = [ ];
foreach(var p in candidates) {
    if((p.id != page.id) && (p.id not in pageids) && (p.id not in
exclude)) {
        let pageids ..= [ p.id ];
        let uniquepages ..= [ p ];
    }
}

//--- firnCore customization: Score berechnen + klassifizieren in einem
Durchgang ---
```

```
var scored = [
    p .. {
        score: p.base + (10 * #[ t foreach var t in tags where (t.type ==
'text') && (string.tolower(t.name) in mytags) ]),
        topic: #[ t foreach var t in tags where
string.startswith(t.value, 'article:topic') ] > 0,
        tutorial: #[ t foreach var t in tags where
string.startswith(t.value, 'article:task-tutorial') ] > 0,
        troubleshooting: #[ t foreach var t in tags where
string.startswith(t.value, 'article:task-troubleshooting') ] > 0,
        reference: #[ t foreach var t in tags where
string.startswith(t.value, 'article:reference') ] > 0
    } foreach
        var p in uniquepages,
        var tags = p.tags
];

//--- firnCore customization: disjunkte Kategorien, je nach Score
absteigend sortiert ---
var topics = list.reverse(list.sort([ p foreach var p in scored where
p.topic ], 'score'));
var tutorials = list.reverse(list.sort([ p foreach var p in scored where
!p.topic && p.tutorial ], 'score'));
var troubleshootings = list.reverse(list.sort([ p foreach var p in scored
where !p.topic && !p.tutorial && p.troubleshooting ], 'score'));
var references = list.reverse(list.sort([ p foreach var p in scored where
!p.topic && !p.tutorial && !p.troubleshooting && p.reference ], 'score'));
var misc = list.reverse(list.sort([ p foreach var p in scored where
!p.topic && !p.tutorial && !p.troubleshooting && !p.reference ],
'score'));

//--- firnCore customization: Fair-Trim via Round-Robin ueber Kategorien,
Hard Cap auf limit ---
var buckets = [ topics, tutorials, troubleshootings, references, misc ];
var selectedids = [ ];
if(limit >= 1) {
    foreach(var round in num.series(0, limit - 1)) {
        foreach(var bucket in buckets) {
            if((#selectedids < limit) && (round < #bucket)) {
                let selectedids ..= [ bucket[round].id ];
            }
        }
    }
}

var topics_sel = [ p foreach var p in topics where p.id in selectedids ];
var tutorials_sel = [ p foreach var p in tutorials where p.id in
selectedids ];
var troubleshootings_sel = [ p foreach var p in troubleshootings where
p.id in selectedids ];
```

```
var references_sel = [ p foreach var p in references where p.id in
selectedids ];
var misc_sel = [ p foreach var p in misc where p.id in selectedids ];
var total = #topics_sel + #tutorials_sel + #troubleshootings_sel +
#references_sel + #misc_sel;

<div class="noindex">
    <div class="mt-control-block-content">

        //--- Show Topic Pages ---
        if(#topics_sel > 0) {
            if(#topics_sel < total) {
                <div class="mt-control-page-subtitle">
wiki.localize("MindTouch.IDF.topics"); </div>
            }
            <dl class="mt-control-seealso">
            foreach(var p in topics_sel) {
                    <dt> web.link(p.uri, p.title) </dt>
                    <dd> wiki.text(p.path, "Overview") </dd>
            }
            </dl>
        }

        //--- Show Tutorial Pages ---
        if(#tutorials_sel > 0) {
            if(#tutorials_sel < total) {
                <div class="mt-control-page-subtitle">
wiki.localize("MindTouch.IDF.tutorial") </div>
            }
            template("MindTouch/Controls/ListPages", {
                pages: tutorials_sel,
                style: "bullets",
                sort: "custom"
            });
        }

        //--- Show Troubleshooting Pages ---
        if(#troubleshootings_sel > 0) {
            if(#troubleshootings_sel < total) {
                <div class="mt-control-page-subtitle">
wiki.localize("MindTouch.IDF.troubleshooting") </div>
            }
            template("MindTouch/Controls/ListPages", {
                pages: troubleshootings_sel,
                style: "bullets",
                sort: "custom"
            });
        }

        //--- Show Reference Pages ---
```

```
        if(#references_sel > 0) {
            if(#references_sel < total) {
                <div class="mt-control-page-subtitle">
wiki.localize("MindTouch.IDF.reference") </div>
            }
            template("MindTouch/Controls/ListPages", {
                pages: references_sel,
                style: "bullets",
                sort: "custom"
            });
        }

        //--- Show Miscellaneous Pages ---
        if(#misc_sel > 0) {
            if(#misc_sel < total) {
                <div class="mt-control-page-subtitle">
wiki.localize("MindTouch.IDF.other") </div>
            }
            template("MindTouch/Controls/ListPages", {
                pages: misc_sel,
                style: "bullets",
                sort: "custom"
            });
        }

        //--- Show Parent Page (if no other pages were shown) ---
        if(parents && (total == 0)) {
            template("MindTouch/Controls/ListPages", {
                pages: [ page.parent ],
                style: "bullets",
                sort: "custom"
            });
        }
    </div>
</div>
```