

```
/* User Variables */
var srcpath = ($0 ?? args.path ..'/*');
var showcomplete = ($1 ?? args.showcomplete ?? 'true');
var onlymine = ($2 ?? args.onlymine ?? 'true');
var excludepages = ($3 ?? args.exclude ?? [""]);

/* System Variables DO NOT EDIT below this line! */
if(srcpath == '/*') {let srcpath = 'type: wiki AND title: "Projekte"'}
else { let srcpath = 'type: wiki AND title: "Projekte" AND path: '
..string.replace(srcpath, 'User:', 'User\\:')}

var IsLate='false';
var BackColor = 'white';
var todaysdate = date.format(date.changetimezone(date.now,
user.timezone), "dd/MM/yyyy");
var listpage = 'true';

var globalexcludepages = ["XXX/XXX/Projekte",
                           "User:rberinger/Projekte"];

<table width="100%" class="sortable, table, myTable" border="0"
cellpadding="1" cellspacing="1" ><tbody>
  <th>
    <b>'Bereich'</b>
  </th>
  <th>
    <b>'Mitarbeiter'</b>
  </th>
  <th>
    <b>'Task'</b>
  </th>
  <th>
    <b>'Details'</b>
  </th>
  <th>
    <b>'Start'</b>
  </th>
  <th>
    <b>'Deadline'</b>
  </th>
  <th>
    <b>'Prio'</b>
  </th>
  <th>
    <b>'Status'</b>
  </th>

  foreach(var t in wiki.getsearch('Projekte', 100, 'title', srcpath )) {
    let listpage = 'true';
```

```

    foreach(var e in excludepages) {
        if(e == t.path) {
            let listpage = 'false';
        }
    }

    foreach(var e in globalexcludepages) {
        if(e == t.path) {
            let listpage = 'false';
        }
    }

    if(listpage == 'true') {
        if(#t.subpages > 0) {
            foreach(var p in t.subpages) {
                var utext =
xml.text(wiki.page(p.path), "/*[@class='username']");
                var stext =
xml.text(wiki.page(p.path), "/*[@class='summary']");
                var sdtext =
xml.text(wiki.page(p.path), "/*[@class='startdate']");
                var edtext =
xml.text(wiki.page(p.path), "/*[@class='enddate']");
                var prtext =
xml.text(wiki.page(p.path), "/*[@class='priority']");
                var sttext =
xml.text(wiki.page(p.path), "/*[@class='status']");
                if(date.IsAfter(todaysdate, edtext)) {
                    let IsLate = 'true';
                    let BackColor = '#FFCC00';
                } else {
                    if(date.IsSameDay(todaysdate, edtext)) {
                        let IsLate = 'false';
                        let BackColor = '#FFCC00';
                    } else {
                        let IsLate = 'false';
                        let BackColor = 'white';
                    }
                }
            }
        }

        let listitem = 'true';
        if(onlymine == 'true') {
            if(!string.contains(string.ToLower(utext), user.name)) {
                let listitem = 'false'
            }
        }

        if(listitem == 'true') {
            if(string.ToUpper(string.Trim(sttext)) == 'C') {

```

```

if(showcomplete == 'true') {
  <tr BGCOLOR="Mediumspringgreen">
    <td>
      <strike>
        web.link(p.parent.parent.uri,p.parent.parent.title);
      </strike>
    </td>
    <td>
      <strike>
        utext;
      </strike>
    </td>
    <td>
      <strike>
        web.link(p.uri, p.title);
      </strike>
    </td>
    <td>
      <strike>
        stext;
      </strike>
    </td>
    <td>
      <strike>
        sdtext;
      </strike>
    </td>
    <td>
      <strike>
        edtext;
      </strike>
    </td>
    <td>
      <strike>
        prtext;
      </strike>
    </td>
    <td>
      <strike>
        sttext;
      </strike>
    </td>
  </tr>
}
} else {
  <tr BGCOLOR=(BackColor)>
    <td>
      web.link(p.parent.parent.uri,p.parent.parent.title);
    </td>
    <td>

```

```
        utext;
    </td>
    <td>
        web.link(p.uri, p.title);
    </td>
    <td>
        stext;
    </td>
    <td>
        sdtext;
    </td>
    <td>
        edtext;
    </td>
    <td>
        prtext;
    </td>
    <td>
        stttext;
    </td>
</tr>
}
}
}
    }
}
}
</tbody></table>
```