

```
anwesend = 0
-- Anzahl der anwesenden Mitglieder
mitglieder = 0
-- Anzahl der nicht-ruhenden Mitglieder
anwesende = ""
-- Liste der anwesenden Mitglieder
entschuldigte = ""
-- Liste der entschuldigt fehlenden Mitglieder
unentschuldigte = ""
-- Liste der unentschuldigt fehlenden Mitglieder
gaeste = ""
-- Liste der Nicht-Mitglieder, die anwesend sind
ruhende = ""
-- Liste der ruhenden Mitglieder
sitzung = "Sitzung"
-- sitzung ist die Art der Sitzung. Falls leer (""), "Sitzung" wird genutzt
datum = ""
-- datum ist das Datum der Sitzung
beginn = ""
-- Beginnzeit der Sitzung
ende = ""
-- Endzeit der Sitzung
beschlussdatum = ""
-- Datum, an dem das Protokoll bestätigt wird
sitzungsleiter = ""
protokollant = ""
finanzerchen = "Der:Die Finanzer:in"
finanzerchenAdj = "seinen:ihren"
finanzerchenPronom = "er:sie"
fristen=5
```

```
protokolle = ""
-- Text für die bestätigten und vertagten Protokolle
```

```
function optText(opt, standard)
-- prints first argument if it is not empty
-- otherwise print second arg
if opt == "" then
    tex.sprint(standard);
else
    tex.sprint(opt);
end
end
```

```
function kopfzeile()
tex.sprint("Protokoll zur ");
if sitzung == "" then
    tex.sprint("Sitzung");
else
    tex.sprint(sitzung);
end
tex.sprint(" am " .. datum);
end
```

```

function titel()
  tex.sprint([[ \textbf{}}]);
  tex.print([[ \LARGE Protokoll zur ]] .. sitzung .. " am " .. datum .. [[ {} \ \ \vspace{1ex}}]);
  tex.print("Sitzungsleitung: " .. sitzungsleiter .. " , " .. "Protokollführung: " .. protokollant);
end

```

```

function printList(list, wer)
  -- list: the list of people that should be printed
  -- wer: der Titel, der über der Liste steht. Sagt, wer aufgelistet wird.
  if list ~= "" then
    tex.sprint([[ \item[] \textbf{}}] .. wer .. [[{: } \]])
    tex.sprint(list)
  end
end

```

```

function formalia()
  tex.print(sitzungsleiter .. " eröffnet die Sitzung um " .. beginn .. "~Uhr. ");
  if fristen < 2 then
    tex.print("Die Beschlussfähigkeit kann nicht festgestellt werden, da die Fristen zur Einberufung der Sitzung nicht eingehalten wurden. Es sind " .. anwesend .. " von " .. mitglieder .. " anwesend.");
  else
    if anwesend * 2 <= mitglieder then
      tex.print("Die Beschlussfähigkeit liegt mit " .. anwesend .. " von " .. mitglieder .. " Mitgliedern nicht vor.");
    else
      tex.print("Die Beschlussfähigkeit wird mit " .. anwesend .. " von " .. mitglieder .. " Mitgliedern festgestellt.");
    end
  end
  tex.print(protokolle)
end
-- body
end

```

```

function kuerzel(kuerzel, name)
  tex.print([[ \newcommand{\}] .. kuerzel .. "]{ " .. name .. [[ \xspace}}]);

  lastLetter = string.sub(name, -1, -1);
  secLastLetter = string.sub(name, -2, -2);
  if lastLetter == "s" or lastLetter == "x" or lastLetter == "ß" or
    (lastLetter == "." and (
      secLastLetter == "S" or
      secLastLetter == "X" or
      secLastLetter == "s" or
      secLastLetter == "x" )) then
    -- Name endet auf s-Laut oder Nachname ist abgekürzt mit s-Laut
    tex.print([[ \newcommand{\}] .. kuerzel .. "s]{ " .. name .. [[ \xspace}}]);
  else
    tex.print([[ \newcommand{\}] .. kuerzel .. "s]{ " .. name .. [[ \s\xspace}}]);
  end
end

```

```

function anwesendesMitglied(person, comment)
  anwesend = anwesend + 1;
  mitglieder = mitglieder + 1;

```

```
if (anwesend == 1) then
  anwesende = person;
else
  anwesende = anwesende .. ", " .. person;
end
if (comment == "") then
else
  anwesende = anwesende .. "~(" .. comment .. ")";
end
end
```

```
function entschuldigtesMitglied(person, comment)
  mitglieder = mitglieder + 1;
  if entschuldigte == "" then
    entschuldigte = person;
  else
    entschuldigte = entschuldigte .. ", " .. person;
  end
  if (comment == "") then
  else
    entschuldigte = entschuldigte .. "~(" .. comment .. ")";
  -- body
  end
end
```

```
function unentschuldigtesMitglied(person, comment)
  mitglieder = mitglieder + 1;
  if unentschuldigte == "" then
    unentschuldigte = person;
  else
    unentschuldigte = unentschuldigte .. ", " .. person;
  end
  if (comment == "") then
  else
    unentschuldigte = unentschuldigte .. "~(" .. comment .. ")";
  end
end
```

```
function verspaetetesMitglied(person, comment)
  mitglieder = mitglieder + 1;
  if anwesende == "" then
    anwesende = person;
  else
    anwesende = anwesende .. ", " .. person;
  end
  if (comment == "") then
  else
    anwesende = anwesende .. "~(" .. comment .. ")";
  end
end
```

```
function ruhendesMitglied(person, comment)
  if ruhende == "" then
```

```

    ruhende = person;
else
    ruhende = ruhende .. ", " .. person;
end
if (comment == "") then
else
    ruhende = ruhende .. "~(" .. comment .. ")";
end
end

```

```

function Gast(person, comment)
if gaeste == "" then
    gaeste = person;
else
    gaeste = gaeste .. ", " .. person;
end
if comment == "" then
else
    gaeste = gaeste .. "~(" .. comment .. ")";
end
-- body
end

```

```

function prVerfuegungsrahmen()
-- schreibt das, was in Finanzerchen steht
tex.sprint(finanzerchen .. " gibt bekannt, dass " .. finanzerchenPronom .. " " .. finanzerchenAdj .. "
Verfuegungsrahmen genutzt hat, ");
end

```

```

function Protokoll(art, datum, wie)
-- art: Art der Sitzung, wird vor dem Datum eingefuegt
if wie == "oG" or wie == "ohne Gegenrede" then
    protokolle = protokolle .. "Das Protokoll " .. art .. " vom " .. datum .. " wird ohne Gegenrede bestaetigt. ";
elseif wie == "Disk" or wie == "Diskussion" or wie == "disk" or wie == "discussion" then
    protokolle = protokolle .. datum;
elseif wie == "vertagt" or wie == "Vertagt" then
    protokolle = protokolle .. "Das Protokoll " .. art .. " vom " .. datum .. " wird vertagt. ";
else
    protokolle = protokolle .. [[\fehler{Ein Fehler ist aufgetreten.} "]]
end
end

```

```

function ProtokollAbstimmung(art, datum, dafuer, dagegen, enthaltung)
    protokolle = protokolle .. "Das Protokoll " .. art .. " vom " .. datum .. " wird mit " .. dafuer .. "/" .. dagegen .. "/" ..
enthaltung .. " Stimmen ";
if dagegen >= dafuer then
    protokolle = protokolle .. "nicht bestaetigt. ";
elseif dafuer < enthaltung then
    protokolle=protokolle .. "mit einer Enthaltungsmehrheit nicht bestaetigt. ";
elseif dagegen == 0 then
    protokolle = protokolle .. "einstimmig bestaetigt. ";
else
    protokolle = protokolle .. "bestaetigt. ";
end

```

```
end
end
```

```
-- schreibt das Ergebnis einer Abstimmung in Abhängigkeit der abgegebenen Stimmen
```

```
function abstimmung(ja, nein, enthaltung)
  if ja <= nein then
    tex.sprint("wird mit " .. ja .. "/" .. nein .. "/" .. enthaltung .. " Stimmen abgelehnt.");
  elseif ja < enthaltung then
    tex.sprint("ist mit " .. ja .. "/" .. nein .. "/" .. enthaltung .. " Stimmen mit einer Enthaltungsmehrheit gescheitert.");
  elseif nein == 0 and enthaltung == 0 then
    tex.sprint("wird mit " .. ja .. "/" .. nein .. "/" .. enthaltung .. " Stimmen einstimmig angenommen.");
  else
    tex.sprint("wird mit " .. ja .. "/" .. nein .. "/" .. enthaltung .. " Stimmen angenommen.");
  end
end
```

```
-- schreibt nichts, wenn Argument leer, sonst
```

```
-- entsprechend der angehängten Kalkulation in Tabelle \ref{kalk:referenz}
```

```
function Kalkulationsreferenz(referenz)
  if referenz ~= "" then
    tex.sprint([[entsprechend der angehängten Kalkulation in Tabelle \ref{kalk:}] .. referenz .. "]);
  end
end
```

```
-- schreibt das, was in geld steht
```

```
function prGeld()
  tex.sprint(string.format("%5.2f", geld));
end
```

```
-- schreibt das, was in summe steht
```

```
function prSumme()
  tex.sprint(string.format("%5.2f", summe));
end
```

```
-- schreibt das, was im Argument steht
```

```
function prZahl(zahl)
  tex.sprint(string.format("%.2f", zahl));
end
```