

Tabele

În lucrările academice, tabelele sunt întâlnite la tot pasul, fiind deseori folosite pentru rezumarea rezultatelor din cercetare. Este bine deci să învățați să lucrați cu tabele, dacă vreți produceți lucrări de calitate.

Totuși, dacă există un domeniu din LaTeX cel mai puțin intuitiv, atunci acesta este. Tabelele simple nu necesită un efort prea mare, însă veți observa curând că orice lucru mai avansat poate solicita eforturi substanțiale. Așadar, vom începe cu lucrurile simple și vom înainta pas cu pas.

Soluție alternativă: S-ar putea să economisiți foarte mult timp convertind tabele din foile de calcul OpenOffice cu ajutorul unor plugin-uri open source, vezi de exemplu <http://calc2latex.sourceforge.net/>.

Mediul `tabular`

Mediul `tabular` poate fi folosit pentru redactarea de tabele cu linii orizontale și verticale opționale. LaTeX determină automat lățimea coloanelor.

Prima linie a mediului are forma: `\begin{tabular}[poz]{specificății tabel}`

Argumentul cu *specificățiile tabelului* spune sistemului LaTeX ce aliniere să folosească pentru fiecare coloană și liniile verticale pe care trebuie să le insereze.

Nu trebuie specificat numărul de coloane, deoarece acesta este dedus din numărul de argumente furnizate. Puteți de asemenea să adăugați aici și linii verticale între coloane. Puteți folosi următoarele simboluri pentru a descrie coloanele tabelului (unele necesită includerea pachetului `array`):

<code>l</code>	coloană aliniată la stânga
<code>c</code>	coloană centrată
<code>r</code>	coloană aliniată la dreapta
<code>p{lățime}</code>	coloană paragraf cu textul aliniat vertical sus
<code>m{lățime}</code>	coloană paragraf cu textul aliniat vertical la mijloc (necesită pachetul <code>array</code>)
<code>b{lățime}</code>	coloană paragraf cu textul aliniat vertical jos (necesită pachetul <code>array</code>)

	linie verticală
	linie verticală dublă

Implicit, dacă textul unei coloane este prea lat pentru pagină, LaTeX nu-l va împărți automat pe mai multe linii. Cu ajutorul parametrul `p{lățime}`, puteți defini un tip special de coloană, care va cuprinde textul ca într-un paragraf normal. Puteți pasa lățimea folosind orice unitate suportată de LaTeX, cum ar fi pt, px, mm sau cm, sau *comenzi pentru lungime*, cum este `\textwidth`. Găsiți o listă completă în anexe [Macrouri utile pentru măsurare](#).

Parametrul opțional *poz* poate fi folosit pentru a specifica poziția verticală a tabelului față de poziția textului înconjurător. În cele mai multe cazuri, nu veți avea nevoie de această opțiune. Devine relevantă numai dacă tabelul nu face parte el însuși dintr-un paragraf. Puteți folosi următoarele litere pentru opțiune:

- b jos (*bottom*)
- c centru (implicit)
- t sus (*top*)

În prima linie ați scris câte coloane vreți, alinierea lor și liniile verticale care le separă. În cadrul textului controlat de acest mediu, trebuie să scrieți textul pe care-l doriți, separând coloanele și introducând manual linii noi (pentru a trece la rândul următor). Trebuie să folosiți următoarele comenzi:

&	separator de coloane
\\	începe un rând nou (puteți specifica un spațiu adițional după \\, folosind paranteze drepte, spre exemplu <code>\\[6pt]</code>)
<code>\hline</code>	linie orizontală
<code>\cline{i-j}</code>	linie orizontală parțială începând cu coloana <i>i</i> și terminând cu coloana <i>j</i>

Notați faptul că spațiul alb inserat între aceste comenzi depinde doar de preferințele utilizatorului. Eu personal adaug spații între comenzi ca să fac codul mai ușor de citit.

Exemple de bază

Acest exemplu arată cum se poate crea un tabel simplu în LaTeX. Este un tabel cu 3 linii și 3 coloane, însă fără nicio linie orizontală care să separe rândurile.

```
\begin{tabular}{l c r }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

Incluzând câteva linii verticale, obținem:

```
\begin{tabular}{l | c || r | }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

Pentru a adăuga linii orizontale înainte și după tabel, procedăm în felul următor:

```
\begin{tabular}{l | c || r | }
\hline
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\hline
\end{tabular}
```

Și în cele din urmă, pentru a adăuga linii între toate rândurile, ca și pentru a centra textul din tabel (observați utilizarea mediului centru (*center*) — evident, rezultatul centrării textului nu este evident în cazul acestui exemplu simplu):

```
\begin{center}
\begin{tabular}{l | c || r | }
\hline
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\hline
\end{tabular}
\end{center}
```

```

\hline
\end{tabular}
\end{center}
\begin{tabular}{|r|}
\hline
7C0 & hexazecimal \\
3700 & octal \\
11111000000 & binar \\
\hline
1984 & zecimal \\
\hline
\end{tabular}

```

7C0	hexazecimal
3700	octal
11111000000	binar
1984	zecimal

Specificarea coloanelor cu $\gt\{\backslash\text{comandă}\}$ și $\lt\{\backslash\text{comandă}\}$

Cu ajutorul pachetului `array` puteți modifica specificațiile coloanelor. Acest lucru se face prin intermediul argumentului mediului `tabular` folosind $\gt\{\backslash\text{comandă}\}$ pentru comenzile executate chiar *înainte* de fiecare coloană și $\lt\{\backslash\text{comandă}\}$ pentru comenzile executate imediat *după* fiecare coloană. Spre exemplu, pentru a obține o coloană în modul matematic, scrieți:

```

\begin{tabular}{>{\$}c<{\$}}. Un alt exemplu este schimbarea fontului:
\begin{tabular}{>{\small}c} pentru a afișa coloana într-un font mai mic.

```

Argumentul specificațiilor \gt și \lt trebuie să fie corect echilibrat în ceea ce privește caracterele $\{$ și $\}$. Asta înseamnă că $\gt\{\backslash\text{bfseries}\}$ este valid, în vreme ce $\gt\{\backslash\text{textbf}\}$ nu va merge, iar $\gt\{\backslash\text{textbf}\}$ nu este valid. Dacă aveți nevoie să folosiți textul tabelului ca argument (spre exemplu, utilizând comanda `\textbf` pentru a produce text aldin), trebuie să folosiți comenzile `\bgroup` și `\egroup`: $\gt\{\backslash\text{textbf}\backslash\text{bgroup}\}c\lt\{\backslash\text{egroup}\}$ produce efectul dorit. Acest procedeu se poate aplica numai pentru câteva comenzi de bază din LaTeX. Pentru alte comenzi, cum este `\underline` pentru a scrie text subliniat, trebuie să păstrați temporar textul din coloană într-o căsuță cu ajutorul comenzii `\lrbox`. Mai întâi, trebuie să definiți o astfel de căsuță cu comanda `\newsavebox{\boxname}`, iar apoi puteți defini:

```

>\begin{lrbox}{\boxname}}%
!%
<\end{lrbox}}%
\underline{\unhbox\boxname}}%
}

```

Astfel, se salvează textul într-o căsuță, iar apoi se ia din căsuță cu comanda `\unhbox` (în felul acesta se distruge căsuța; dacă mai e necesară, folosiți în schimb comanda `\unhcopy`) și se pasează comenzii `\underline`. (Pentru LaTeX2e, ar fi bine să folosiți `\usebox{\boxname}` în loc de `\unhbox\boxname.`)

Același truc realizat cu `\raisebox` în loc de `\underline` poate forța toate liniile dintr-un tabel să aibă aceeași înălțime, în locul înălțimii naturale variate ce poate apare, spre exemplu, când în text sunt termeni matematici sau exponenți.

Iată un exemplu ce prezintă utilizarea `p{...}` și `>{\centering}`:

```
\begin{tabular} {>{\centering}p{3.5cm}>{\centering}p{3.5cm}}
  Geometrie & Algebră
\tabularnewline
  \hline
  Puncte & Adunare
\tabularnewline
  Sfere & Înmulțire
\end{tabular}
```

Notați utilizarea comenzii `\tabularnewline` în locul `\\` pentru a evita apariția unei erori `Misplaced \noalign.`

Împărțirea pe linii a textului din tabele

Algoritmii LaTeX pentru formatarea tabelelor au câteva deficiențe. Una dintre ele este faptul că nu vor împărți automat pe linii textul de pe coloane, chiar dacă se depășește lățimea paginii. Pentru coloane despre care știți că vor conține text ce va trebui aranjat pe coloane de o lățime dată, se recomandă să folosiți atributul p și să specificați lățimea dorită a coloanei (deși s-ar putea să fie nevoie să faceți câteva încercări până când veți obține rezultatul dorit). Folosiți atributul m pentru ca liniile de text din coloane să fie aliniate în mijlocul căsuței ce conține textul (numărul maxim de linii pe care va fi aranjat textul de pe fiecare coloană va determina înălțimea acestor căsuțe) și b pentru alinierea în partea inferioară a acesteia.

Iată un exemplu concret. Codul ce urmează creează două tabele cu același conținut; singura diferență este că ultima coloană a celui de-al doilea tabel

are o lățime definită de 5 centimetri, în vreme ce pentru primul tabel nu are fixată o dimensiune. Compilând acest cod:

```
\documentclass{article}

\usepackage[utf8x]{inputenc} % pentru caractere românești
\usepackage{array} % pentru \extrarowheight

\begin{document}
\setlength{\extrarowheight}{2pt} % adaugă spațiu vertical suplimentar
liniilor din tabele
```

Fără specificarea lățimii pentru ultima coloană:

```
\begin{center}
\begin{tabular}{|l|l|l|l|}
\hline
Ziua & Temp Min & Temp Max & Rezumat \\ \hline
Luni & 11C & 22C & Zi senină cu mult soare.
Totuși, briza puternică va reduce temperatura. \\ \hline
Marți & 9C & 19C & Noros cu ploaie, de-a lungul multor regiuni nordice.
Timp frumos în cea mai mare parte a Scoției și Irlandei de Nord,
însă ploaia va ajunge în nordul extrem. \\ \hline
Miercuri & 10C & 21C & Dimineața va continua să plouă.
Vremea se va îmbunătăți la începutul după-amiezii și
va continua să se îmbunătățească de-a lungul serii. \\ \hline
\end{tabular}
\end{center}
```

Cu lățimea specificată:

```
\begin{center}
\begin{tabular}{|l|l|l|l|p{5cm}|}
\hline
Ziua & Temp Min & Temp Max & Rezumat \\ \hline
Luni & 11C & 22C & Zi senină cu mult soare.
Totuși, briza puternică va reduce temperatura. \\ \hline
\end{tabular}
\end{center}
```

\hrule

Marți & 9C & 19C & Noros cu ploaie, de-a lungul multor regiuni nordice.
Timp frumos în cea mai mare parte a Scoției și Irlandei de Nord,
însă ploaia va ajunge în nordul extrem. \[0.5mm]

\hrule

Miercuri & 10C & 21C & Dimineața va continua să plouă.
Vremea se va îmbunătăți la începutul după-amiezii și
va continua să se îmbunătățească de-a lungul serii. \[0.5mm]

\hrule

\end{tabular}

\end{center}

\end{document}

Veți obține următorul rezultat:

Fără specificarea lățimii pentru ultima coloană:

Ziua	Temp Min	Temp Max	Rezumat
Luni	11C	22C	Zi senină cu mult soare. Totuși, briza
Marți	9C	19C	Noros cu ploaie, de-a lungul multor re
Miercuri	10C	21C	Dimineața va continua să plouă. Vren

Cu lățimea specificată:

Ziua	Temp Min	Temp Max	Rezumat
Luni	11C	22C	Zi senină cu mult soare. Totuși, briza puternică va reduce temperatura.
Marți	9C	19C	Noros cu ploaie, de-a lungul multor regiuni nordice. Timp frumos în cea mai mare parte a Scoției și Irlandei de Nord, însă ploaia va ajunge în nordul extrem.
Miercuri	10C	21C	Dimineața va continua să plouă. Vremea se va îmbunătăți la începutul după-amiezii și va continua să se îmbunătățească de-a lungul serii.

Observați că primul tabel este decupat: outputul depășește lățimea paginii.

Alinierea stânga-dreapta a textului în tabele

În rare ocazii, ar putea fi necesar să măriți fiecare linie dintr-un tabel până la lățimea naturală a celei mai lungi linii, spre exemplu când același text trebuie afișat în două limbi diferite, iar utilizatorul vrea să le prezinte unul lângă celălalt, cu liniile sincronizate. Un mediu tabelar ajută la împărțirea textului pe linii, însă nu aliniază textul stânga-dreapta, ceea ce lasă margini drepte neregulate. Pachetul `eqparbox` include comanda `\eqmakebox`, care este la fel ca `\makebox`, doar că, în locul unui argument *width*, ia o etichetă. În

timpul compilării, ține evidența tuturor căsuțelor `\eqmakebox` cu aceeași etichetă, memorând lățimea maximă a textului, astfel încât să poată mări toate căsuțele până la această valoare. În combinație cu pachetul `array`, puteți defini un specificator de coloană ce aliniază la stânga-dreapta textul de pe toate liniile: (Vezi documentația pachetului [eqparbox](#) pentru mai multe detalii.)

```
\newsavebox {\tstretchbox}  
\newcolumnntype{S}[1]{%  
>{\begin{lrbox} {\tstretchbox}}%  
l%  
<{\end{lrbox}}%  
  \eqmakebox[#1][s]{\unhcopy\tstretchbox}}%  
}
```

Alte medii în tabele

Dacă folosiți anumite medii LaTeX în celulele tabelelor, cum ar fi `verbatim` or [enumerate](#)

```
\begin{tabular} { | c | c | }  
\hline  
\begin{verbatim}  
cod  
\end{verbatim}  
& descriere  
\hline  
\end{tabular}
```

puteți întâlni erori de genul

```
! LaTeX Error: Something's wrong--perhaps a missing \item.
```

Pentru a rezolva această problemă, schimbați [specificatorul coloanei](#) la paragraf (`p`, `m` sau `b`).

```
\begin{tabular} { | m{5cm} | c | }
```

Definirea mai multor coloane odată

Puteți defini mai multe coloane identice odată cu ajutorul sintaxei

```
*{num}{str}.
```

Acest lucru este deosebit de util când tabelul are multe coloane.

Iată un tabel cu 6 coloane centrate, flancate de o singură coloană de fiecare parte:

```
\begin{tabular}{l*{6}{c}r}
Team          & P & W & D & L & F & A & Pts \\
\hline
Manchester United & 6 & 4 & 0 & 2 & 10 & 5 & 12 \\
Celtic          & 6 & 3 & 0 & 3 & 8 & 9 & 9 \\
Benfica         & 6 & 2 & 1 & 3 & 7 & 8 & 7 \\
FC Copenhagen   & 6 & 2 & 1 & 2 & 5 & 8 & 7 \\
\end{tabular}
```

Team	P	W	D	L	F	A	Pts
Manchester United	6	4	0	2	10	5	12
Celtic	6	3	0	3	8	9	9
Benfica	6	2	1	3	7	8	7
FC Copenhagen	6	2	1	2	5	8	7

@-expresii

Separatorul de coloană poate fi specificat cu construcția @{...}.

În mod tipic, aceasta ia ca argument un text, iar când se anexează la o coloană, va insera automat acest text în fiecare celulă de pe coloană înainte de datele efective pentru acea celulă. Această comandă șterge spațiul dintre coloane și îl înlocuiește cu ce se găsește între acolade. Pentru a adăuga spațiu, folosiți @{\hspace{lățime}}.

Evident, aceste explicații nu sunt foarte clare, astfel că vor fi necesare câteva exemple pentru clarificare. Câteodată, e de dorit ca în tabelele științifice numerele să fie aliniate după punctul zecimal. Puteți face asta în felul următor:

```

\begin{tabular}{r@{.}l}
  3 & 14159 \\
  16 & 2 \\
  123 & 456 \\
\end{tabular}

```

3.14159
16.2
123.456

Notați că abordarea de mai sus nu va merge bine dacă antetul coloanei este mai lung decât oricare dintre numere. Pentru a centra coloana după punctul zecimal, folosiți pachetul `dcolumn`, care oferă un nou specificator de coloană pentru datele flotante.

Suprimarea spațiilor realizată de `@`-expresii le face foarte utile pentru manipularea spațiului orizontal dintre coloane. Dat fiind un tabel de bază, variind specificațiile coloanelor, obținem:

```

\begin{tabular}{|l|l|}
\hline
date & date \\
\hline
date & date \\
\hline
\end{tabular}

```

`{|l|l|}`

date	date
date	date

`{|@{}l|l@{}|}`

date	date
date	date

`{|@{}l@{}|l@{}|}`

date	date
date	date

`{|@{}l@{}|@{}l@{}|}`

date	date
date	date

Întinderea liniilor și coloanelor

Pentru a încheia acest tutorial, vom lucra puțin la generarea de tabele mai complexe. Deloc surprinzător, comenzile necesare trebuie incluse în datele din tabel.

Linii care se întind pe mai multe coloane

Comanda pentru asta arată în felul următor:

`\multicolumn{număr_coloane}{aliniere}{conținut}`. *număr_coloane* este numărul de coloane succesive care vor fi îmbinate pentru a fi reprezentate ca o singură coloană; *aliniere* este l, c, r sau o opțiune de genul `p{5.0cm}`. Iar *conținut* reprezintă pur și simplu datele efective care vor fi conținute în celulă. Iată un exemplu simplu:

```
\begin{tabular}{|l|l|}
\hline
\multicolumn{2}{|c|}{Componenta echipei} \\
\hline
PT & Paul Robinson \\
FS & Lucus Radebe \\
FC & Michael Duberry \\
FC & Dominic Matteo \\
FD & Didier Domi \\
MC & David Batty \\
MC & Eirik Bakke \\
MC & Jody Morris \\
IN & Jamie McMaster \\
OG & Alan Smith \\
OG & Mark Viduka \\
\hline
\end{tabular}
```

Componenta echipei	
PT	Paul Robinson
FS	Lucus Radebe
FC	Michael Duberry
FC	Dominic Matteo
FD	Didier Domi
MC	David Batty
MC	Eirik Bakke
MC	Jody Morris
IN	Jamie McMaster
OG	Alan Smith
OG	Mark Viduka

Coloane care se întind pe mai multe linii

Primul lucru pe care trebuie să-l faci este să adaugi

`\usepackage{multirow}` în preambul. Asta oferă comanda necesară pentru a întinde o celulă pe mai multe linii:

`\multirow{număr_linii}{lățime}{conținut}`. Sensul argumentelor este destul de ușor de dedus (* pentru *lățime* înseamnă lățimea naturală a conținutului).

...
`\usepackage{multirow}`
 ...

```

\begin{tabular}{|l|l|l|}
\hline
\multicolumn{3}{|c|}{Compo
nența echipei} \\
\hline
Portar & PT & Paul Robinson \\
\hline
\multirow{4}{*}{Fundași} & FS & Lucus Radebe \\
& FC & Michael Dube \\
& FC & Dominic Matt \\
& FD & Didier Domi \\
\hline
\multirow{3}{*}{Mijlocași} & MC & David Batty \\
& MC & Eirik Bakke \\
& MC & Jody Morris \\
\hline
Înaintași & IN & Jamie McMas
ter \\
\hline
\multirow{2}{*}{Om de gol} & OG & Alan Smith \\
& OG & Mark Viduka \\
\hline
\end{tabular}

```

Compo nența echipei		
Portar	PT	Paul Robinson
Fundași	FS	Lucus Radebe
	FC	Michael Dube
	FC	Dominic Matt
	FD	Didier Domi
Mijlocași	MC	David Batty
	MC	Eirik Bakke
	MC	Jody Morris
Înaintași	IN	Jamie McMas ter
Om de gol	OG	Alan Smith
	OG	Mark Viduka

Lucrul principal de notat când folosiți comanda `\multirow` este că trebuie inserată o intrare nulă pentru fiecare celulă corespunzătoare de pe fiecare linie succesivă de înlocuit.

Dacă nu sunt date pentru o celulă, nu tipăriți nimic, însă tot vă trebuie "&" pentru a le separa de datele coloanei următoare. Cititorul atent trebuie să fi dedus deja că pentru un tabel cu *n* coloane, trebuie să fie întotdeauna *n* - 1 "&" pe fiecare linie. Singura excepție este când folosiți `\multicolumn` și `\multirow` pentru a crea celule care se întind pe mai multe coloane sau linii.

Celule care se întind în ambele direcții simultan

Iată un exemplu ceva mai complex despre celule care se întind în ambele direcții simultan, cu marginile trasate corect:

```
\usepackage{multirow}

\begin{tabular}{cc|c|c|c|l}
\cline{3-6}
&&\multicolumn{4}{|c|}{Numere
prime} \\ \cline{3-6}
&&2 & 3 & 5 & 7 \\ \cline{1-6}
\multicolumn{1}{|c|}{\multirow{2}{*}
}{Puteri}} &
\multicolumn{1}{|c|}{504} & 3 & 2 & 0 & 1 \\
0 & 1 & \\ \cline{2-6}
\multicolumn{1}{|c|}{} &
&
\multicolumn{1}{|c|}{540} & 2 & 3 & 1 & 0 \\
1 & 0 & \\ \cline{1-6}
\multicolumn{1}{|c|}{\multirow{2}{*}
}{Puteri}} &
\multicolumn{1}{|c|}{cmmdc} & 2 & 2 & 0 & 0 \\
&
\multicolumn{1}{|c|}{cmmmc} & 3 & 3 & 1 & 1 \\
2 & 0 & 0 & min \\ \cline{2-6}
\multicolumn{1}{|c|}{} &
&
\multicolumn{1}{|c|}{cmmmc} & 3 & 3 & 1 & 1 \\
3 & 1 & 1 & max \\ \cline{1-6}
\end{tabular}
```

		Numere prim			
		2	3	5	7
Puteri	504	3	2	0	1
	540	2	3	1	0
Puteri	cmmdc	2	2	0	0
	cmmmc	3	3	1	1

Comanda `\multicolumn{1}{|c|}{...}` este folosită numai pentru a trasa margini verticale la stânga și la dreapta celulei. Chiar și atunci când este combinată cu `\multirow{2}{*}{...}`, tot trasează margini verticale ce se întind numai de-a lungul primei linii. Pentru a trasa linii verticale care se întind pe mai multe linii, folosim comanda `\multicolumn{1}{|c|}{...}` care trebuie să includă liniile unite de comanda `multirow`. Notați faptul că nu puteți folosi direct `\hline` pentru a trasa linii orizontale, întrucât nu vrem ca linia să fie trasată peste textul ce se întinde de-a lungul mai multor linii. Folosim în schimb comanda `\cline{2-6}` pentru a nu trasa linia de-a lungul primei coloane, ce conține textul "Puteri".

Iată încă un exemplu ce exploatează aceleași idei pentru a realiza binecunoscuta dublă dihotomie "2x2":

```

\begin{tabular}{r|c|c}
\multicolumn{1}{r}{} & & \\
& & \\
& & \\
& & \\
\multicolumn{1}{c}{} & {interactiv} & {neinteractiv} \\
\cline{2-3} & & \\
intrări multiple & Bibliotecă & Univ \\
unu-la-unu & Carte & Tu
\cline{2-3} & & \\
unu-la-unu & Carte & \\
Tutorial & & \\
\cline{2-3} & & \\
\end{tabular}

```

Redimensionarea tabelelor

Comanda `\resizebox{lățime}{înălțime}{obiect}` poate fi folosită cu `tabular` pentru a specifica înălțimea și lățimea unui tabel. Exemplul următor arată cum se poate redimensiona un tabel la 8cm lățime, menținând rata inițială lățime/înălțime.

```

\resizebox{8cm}{!}{
\begin{tabular}...
\end{tabular}
}

```

Ca o alternativă, puteți folosi `\scalebox{rată}{obiect}` în același fel, însă cu rate în loc de dimensiuni fixe:

```

\scalebox{0.7}{
\begin{tabular}...
\end{tabular}
}

```

Atât `\resizebox`, cât și `\scalebox` necesită pachetul [graphicx](#).

Pentru a redimensiona spațiul dintre coloane (LaTeX va alege implicit coloane foarte strânse), puteți modifica separarea coloanelor:
`\setlength{\tabcolsep}{5pt}`. Valoarea implicită este 6pt.

Tabele cu text scris pe lateral

Tabelele pot fi aranjate pe lateral într-un document cu ajutorul pachetului `rotating` și a mediului `sidewaystable` care să includă tabelul. (NOTĂ: cele mai multe programe de vizualizare pentru DVI nu suportă afișarea textului rotit. Converteți documentul într-un PDF ca să vedeți rezultatul. Cele mai multe, dacă nu toate, programele de vizualizare a PDF-urilor suportă textul rotit.)

```
\usepackage{rotating}

\begin{sidewaystable}
\begin{tabular}
...
\end{tabular}
\end{sidewaystable}
```

Pentru a roti textul înăuntrul tabelului:

```
\usepackage{rotating}

\begin{tabular}{...}
\begin{sideways}
text rotit ...
\end{sideways}
\end{tabular}
```

Când doriți să așezați tabelul rotit în exact același loc în care apare în fișierul sursă (*.tex*), puteți folosi pachetul `rotfloat`. Atunci puteți folosi `\begin{sidewaystable}[H]` la fel ca pentru orice alt tabel. Opțiunea 'H' nu poate fi folosită fără acest pachet.

Culori de bază alternate în tabele

Pachetul `xcolor` oferă comenzile necesare pentru a realiza tabele cu culori de bază alternate, când sunt încărcate cu opțiunea `table`. Comanda

`\rowcolors{<începutul liniei>}{<culoare impară>}{<culoare pară>}` trebuie specificată chiar înainte de a începe mediul `tabular`.

```
\documentclass{article}
```

```
\usepackage[table]{xcolor}
```

```
\begin{document}
```

```
\begin{center}
```

```
\rowcolors{1}{green}{pink}
```

```
\begin{tabular}{lll}
```

```
impar & impar & impar \\
```

```
par & par & par \\
```

```
impar & impar & impar \\
```

```
par & par & par \\
```

```
\end{tabular}
```

```
\end{center}
```

```
\end{document}
```

impar	impar	impar
par	par	par
impar	impar	impar
par	par	par

Comanda `\hiderowcolors` dezactivează colorarea liniei specificate.

Colorarea poate fi reactivată în interiorul tabelului cu comanda

```
\showrowcolors.
```

Colorarea celulelor individuale

La fel ca mai sus, colorarea celulelor individuale folosește pachetul

[xcolor](#).

```
\usepackage[table]{xcolor}
```

```
\cellcolor[gray]{0.9} % gri deschis ("gray" este aici schema de colorare alb-negru)
```

```
\cellcolor{red} % roșu
```

Linii verticale parțiale

Adăugarea unei linii verticale parțiale unei celule individuale:

```

\begin{tabular}{lcr}
1 & 2 & 3 \\
4 & 5 & \multicolumn{1}{r}{6} \\
7 & 8 & 9 \\
\end{tabular}

```

1	2	3
4	5	6
7	8	9

Înlăturarea unei părți a unei linii verticale dintr-o celulă anume:

```

\begin{tabular}{|l|c|r|}
1 & 2 & 3 \\
4 & 5 & \multicolumn{1}{r}{6} \\
7 & 8 & 9 \\
\end{tabular}

```

1	2	3
4	5	6
7	8	9

Mediul `table` - atașarea de legende, etc

Deși mediul `tabular` poate crea tabele complexe, nu acoperă tot ce puteți face cu tabelele. Spre exemplu, poate veți dori o legendă pentru un tabel. Din acest motiv, dar și din altele, ar trebui să plasați mediul `tabular` într-un mediu `table`:

```

\begin{table}
\caption{Activitate la temperatura maximă}
\begin{tabular}{|r|r||c|c|c|}
...
\end{tabular}
\end{table}

```

De ce există aceste două medii diferite? Gândiți-vă în felul următor: mediul `tabular` se ocupă cu aranjarea elementelor într-o structură tabelară, în vreme ce mediul `table` reprezintă un tabel la un nivel mai conceptual. Asta explică de ce nu mediul `tabular`, ci `table` este cel care oferă posibilitatea de a atașa legende (întrucât legenda nu se afișează în structura tabelară).

Mediul `table` prezintă multe similarități cu mediul `figure`, în ce privește tratarea elementelor "flotante", etc. Spre exemplu, poate specifica plasarea pe pagină cu opțiunea `[placement]`, valorile valide sunt orice combinație de (ordinea nu este importantă):

`h` unde este declarat tabelul (aici — *here*)

t la începutul (*top*) paginii

b la sfârșitul (*bottom*) paginii

p pe o pagină dedicată de elemente flotante

suprascrie restricțiile flotante implicite. Spre exemplu, dimensiunea

! maximă permisă unui element flotant b este în mod normal foarte mică;

dacă vreți o dimensiune mai mare, aveți nevoie și de parametrul !.

Valoarea implicită este [tbp]. Dacă vreți să plasați un tabel în locul în care este declarat, nu scrieți doar [h]; dacă tabelul nu încapă (întrucât textul este aproape de sfârșitul paginii, să zicem), va fi mutat pe o pagină dedicată cu elemente flotante (ca în cazul unui element flotant p), care poate fi la ceva distanță în document. O regulă general recomandată este să folosiți întotdeauna htbp până la sfârșitul documentului, stadiu în care parametrii flotanți finali pot fi mai bine rafinați.

Mediul table este de asemenea util când vreți să aveți o listă de tabele la începutul sau la sfârșitul documentului, cu ajutorul comenzii

\listoftables; aceasta permite realizarea de referințe la tabel în felul următor:

Puteți face referire la table~\ref{tabelul_meu} pentru a vedea un exemplu.

...

```
\begin{table}
  \begin{tabular}
    ...
  \end{tabular}
  \caption{Un exemplu de tabel}
  \label{tabelul_meu}
\end{table}
```

Mediul `tabular*` - controlarea lățimii unui tabel

Acesta este în fapt o ușoară extensie a versiunii originale *tabular*, deși necesită un argument în plus (înainte de descrierile coloanelor) pentru a specifica lățimea preferată a tabelului.

```

\begin{tabular*}{0.75\textwidth}{|c|c|c|r|}
\hline
eticheta 1 & eticheta
2 & eticheta 3 &
eticheta 4 \\
\hline
articolul 1 & articolul
2 & articolul 3 &
articolul 4 \\
\hline
\end{tabular*}

```

eticheta 1	eticheta 2	eticheta 3	
articolul 1	articolul 2	articolul 3	

Totuși, asta s-ar putea să nu arate exact așa după cum era de așteptat. Coloanele tot mai sunt la lățimea lor naturală (îndeajuns de mari cât să cuprindă conținutul), în vreme ce liniile sunt îndeajuns de mari pe cât de lat a fost specificat tabelul. Dacă nu vă place acest comportament implicit, trebuie de asemenea să inserați manual încă un spațiu pentru o coloană. LaTeX are *rubber lengths*, care, spre deosebire de altele, nu sunt fixate (lungimi flexibile). LaTeX poate decide dinamic cât de mari ar trebui să fie aceste lungimi. Spre exemplu:

```

\begin{tabular*}{0.75\textwidth}{@{\extracolsep{\fill}}|c|c|c|r|}
\hline
eticheta 1 & eticheta 2 & eticheta 3 & eticheta 4 \\
\hline
articolul 1 & articolul 2 & articolul 3 & articolul 4 \\
\hline
\end{tabular*}

```

eticheta 1	eticheta 2	eticheta 3	eticheta 4
articolul 1	articolul 2	articolul 3	articolul 4

Observați construcția `@{...}` adăugată la începutul descrierii coloanei. În cadrul acesteia se găsește comanda `\extracolsep`, care necesită o lățime. Puteam folosi o lățime fixă. Cu toate acestea, utilizând o lungime flexibilă, cum este `\fill`, coloanele sunt spațiate automat în mod egal.

Pachetul `tabularx` - extinderea coloanelor

Acest pachet oferă un mediu tabelar numit `tabularx`, similar cu `tabular*`, cu excepția faptului că are un nou specificator de coloană, `x`. Coloana(e) specificat(e) cu acest specificator vor fi extinse pentru a aduce tabelul la lățimea specificată, simplificând în mare măsură crearea tabelelor.

```
\usepackage{tabularx}
```

...

```
\begin{tabularx}{\textwidth}{|X|X|X|X|}  
  \hline  
  eticheta 1 & eticheta 2 & eticheta 3 & eticheta 4 \\ \hline  
  articolul 1 & articolul 2 & articolul 3 & articolul 4 \\ \hline  
\end{tabularx}
```

eticheta 1	eticheta 2	eticheta 3	eticheta 4
articolul 1	articolul 2	articolul 3	articolul 4

Conținutul căsuțelor este tratat la fel ca pentru o coloană tip paragraf `p`, cu excepția faptului că lățimea se calculează automat. Dacă folosiți pachetul `array`, puteți aplica orice comandă `>\cmd` sau `<\cmd` pentru a realiza comportamentul dorit (precum `\centering` sau `\raggedright\arraybackslash`), după cum a fost descris mai devreme.

O altă opțiune este folosirea `\newcolumntype` pentru a formata coloanele într-un mod specific. Aceasta definește un nou specificator pentru coloane, de exemplu, `R`. În exemplul următor, coloanele a doua și a patra sunt ajustate într-un mod diferit (`\raggedleft`):

```
\usepackage{tabularx}
```

...

```
\newcolumntype{R}{>\raggedleft\arraybackslash}X  
\begin{tabularx}{\textwidth}{|l|R|l|R|}  
  \hline  
  eticheta 1 & eticheta 2 & eticheta 3 & eticheta 4 \\ \hline
```

```

\hline
articolul 1 & articolul 2 & articolul 3 & articolul 4 \\
\hline
\end{tabularx}

```

eticheta 1	eticheta 2	eticheta 3	eticheta 4
articolul 1	articolul 2	articolul 3	articolul 4

`Tabularx` poate fi folosit cu linii care se întind pe mai multe coloane, cu ajutorul comenzii `\multicolumn`. Cele două coloane centrale sunt tratate ca una singură, folosind opțiunea `X@{}`. Notați că lățimea liniei formate de comanda `\multicolumn` (care în exemplul nostru este 2) ar trebui să fie egală cu lățimea coloanelor unite (în exemplul de față 1+1):

```

\usepackage{tabularx}
...
\begin{tabularx}{1\textwidth}{>{\setlength\hspace{1\hspace}\centering}X|>{\setlength\hspace{1\hspace}\raggedleft}X@{ }>{\setlength\hspace{1\hspace}\raggedright}X|>{\setlength\hspace{1\hspace}\centering}X|}
\hline
Eticheta 1 &
\multicolumn{2}{>{\centering\setlength\hspace{2\hspace}}X|}{Eticheta 2} &
Eticheta 3 \tabularnewline
\hline
123 & 123 & 456 & 123 \tabularnewline
\hline
123 & 123 & 456 & 123 \tabularnewline
\hline
\end{tabularx}

```

Eticheta 1	Eticheta 2	Eticheta 3
123	123456	123
123	123456	123

Imagini centrate vertical

Inserarea de imagini pe o linie a unui tabel le va alinia în partea superioară a celulei. Utilizând pachetul `array`, această problemă poate fi rezolvată. Definierea unui nou tip de coloană (*newcolumnntype*) va menține imaginea centrată vertical.

```
\newcolumnntype{V}{>{\centering\arraybackslash} m{.4\linewidth} }
```

Sau folosiți o căsuță în modul paragraf (*parbox*) pentru a centra imaginea.

```
\parbox[c]{1em}{\includegraphics{image.png}}
```

Tabele profesionale

Multe tabele din cărțile și revistele profesionale prezintă tabele simple, care au un spațiu corespunzător înainte și după linii, și care nu folosesc aproape *niciodată* linii verticale. Multe exemple de tabele LaTeX (inclusiv cele din acest wikimanual) prezintă utilizarea liniilor simple (orizontale: "`\hline`" și verticale: "`|`") și duble ("`\hline\hline`", respectiv "`||`"), considerate nenecesare și care distrag atenția în cazul unui format profesional pentru publicații. Pachetul `booktabs` poate oferi cu ușurință acest profesionalism în tabelele LaTeX, iar [documentația](#) aferentă vă pune la dispoziție un ghid pentru a înțelege ce înseamnă un tabel "reusit".

Pe scurt, pachetul folosește `\toprule` pentru linia ce marchează începutul tabelului, `\midrule` pentru liniile ce apar în interiorul tabelului și `\bottomrule` pentru linia ce marchează sfârșitul tabelului. Asta asigură faptul că lățimea liniei și spațierea sunt acceptabile. În plus, puteți folosi `\cmidrule` în locul `\cline` pentru linii care marchează un număr specificat de coloane. Exemplul următor compară implementările uzuale din LaTeX (cu și fără pachetul `array`) cu `booktabs`:

LaTeX uzual	Cu pachetul <code>array</code>	Cu <code>booktabs</code>
<code>\begin{tabular}{llr}</code>	<code>\usepackage{array}</code>	<code>\usepackage{booktabs}</code>
<code>\hline</code>	<code>}</code>	
<code>\multicolumn{2}{c}{Artico</code>	...	
<code>l} \\\</code>	...	
<code>\cline{1-2}</code>		<code>\begin{tabular}{llr}</code>
Animal & Descriere & Preț	<code>\begin{tabular}{llr}</code>	<code>\toprule</code>
(\ \$) \\\	<code>}</code>	<code>\multicolumn{2}{c}{Artico</code>
<code>\hline</code>	<code>\firsthline</code>	<code>l} \\\</code>

```

Gnat & per gram & 13.65 \multicolumn{2}{c} \cmidrule(r){1-2}
\\ \} \{Articol} \\ Animal & Descriere & Preț
& la bucată & 0.01 \\ \cline{1-2} (\$) \\
Gnu & la bax & 92.50 \\ Animal & \midrule
Emu & la bax & 33.33 \\ Descriere & Preț Gnat & per gram & 13.65
Armadillo & congelat & (\$) \\ \\
8.99 \\ \hline & la bucată & 0.01 \\
\hline Gnat & per gram Gnu & la bax & 92.50 \\
& 13.65 \\ Emu & la bax & 33.33 \\
& la bucată & Armadillo & congelat &
0.01 \\ 8.99 \\
Gnu & la bax & \bottomrule
92.50 \\ \end{tabular}
Emu & la bax
& 33.33 \\
Armadillo &
congelat & 8.99 \\
\lasthline
\end{tabular}

```

Articol	
Animal	Descriere
Gnat	per gram la bucată
Gnu	la bax
Emu	la bax
Armadillo	congelat

Articol	
Animal	Descriere
Gnat	per gram la bucată
Gnu	la bax
Emu	la bax
Armadillo	congelat

Poate veți avea nevoie de note de subsol sub un tabel (și nu la sfârșitul paginii) și de o legendă (sau titlu) așezate la o distanță corespunzătoare deasupra tabelului. Aceste lucruri se pot realiza folosind pachetul [ctable](#). Acesta oferă opțiunea inserării unui scurt titlu în lista de tabele, în locul titlului sau legendei efective (care pot fi foarte lungi sau nepotrivite pentru lista de tabele). Pachetul [ctable](#) utilizează pachetul [booktabs](#).

Adăugarea de spațiu deasupra și sub liniile create de comenzile `\hline` și `\cline`

Un mod alternativ de a ajusta spațierea liniilor este să adăugăm

`\noalign{\smallskip}` înainte și/sau după comenzile `\hline` și `\cline{i-j}`:

LaTeX uzual

```
\begin{tabular}{llr}
\hline\noalign{\smallskip}
\multicolumn{2}{c}{Articol} \\
\cline{1-2}\noalign{\smallskip}
Animal & Descriere & Preț (\$) \\
\noalign{\smallskip}\hline\noalign{\smallskip}
Gnat & per gram & 13.65 \\
& la bucată & 0.01 \\
Gnu & la bax & 92.50 \\
Emu & la bax & 33.33 \\
Armadillo & congelat & 8.99 \\
\hline
\noalign{\smallskip}\hline
\end{tabular}
```

Articol		
Animal	Descriere	Preț
Gnat	per gram	13
	la bucată	0
Gnu	la bax	92
Emu	la bax	33
Armadillo	congelat	8

Puteți de asemenea să specificați explicit un spațiu după terminatorul de linie:

```
\begin{tabular}{|l|l|}
\hline
Mineral & Culoare \\
\hline
Rubin & roșu \\
Safir & albastru \\
\hline
\end{tabular}
```

Mineral	Culoare
Rubin	roșu
Safir	albastru

Tabele cu mărimi diferite de fonturi

Puteți modifica mărimea implicită a fontului pentru un tabel folosiți pur și simplu o comandă care setează mărimea fontului (aici: `\footnotesize`) după declarația `\begin{table}`...

```

\begin{table}[h]\footnotesize
\caption{Rezultate la temperatura maximă}
\begin{tabular}{|r|r||c|c|c|}
...
\end{tabular}
\end{table}

```

Mărimea fontului pentru titlul sau legenda tabelului nu este afectată.

Pentru a controla mărimea fontului pentru legendă, vezi [Stiluri de titluri](#).

Vreți ceva mai complicat?

Uitați-vă peste următoarele pachete:

- [hhline](#): controlați liniile orizontale
- [array](#): definiți mai ușor coloanele
- [colortbl](#): faceți tabele mai colorate
- [supertabular](#): pentru tabele care se întind pe mai multe pagini
- [longtable](#): similar cu `supertabular`
 - Notă: notele de subsol nu merg cum trebuie într-un mediu tabelar obișnuit, însă situația se schimbă într-un mediu *longtable*
- [xtab](#): încă un pachet pentru tabele care trebuie să se întindă pe mai multe pagini
- [tabulary](#): `tabular*` modificat, permițând setarea lățimii coloanelor pentru înălțimi egale
- [arydshln](#): creează linii orizontale și verticale punctate
- [ctable](#): permite afișarea notelor de subsol sub tabel și a titlului sau legendei spațiate corespunzător deasupra tabelului (încorporează pachetul `booktabs`)
- [slashbox](#): creează tabele 2D, prima celulă conținând o descriere pentru ambele axe

Rezumat

Aici se încheie discuția noastră despre elementele de bază privind tabelele. Experimentarea conduce repede la stăpânirea cunoștințelor. Sintaxa tabelelor

din LaTeX poate arăta destul de neîngrijit, iar exemplele noi pot crea confuzie. Însă din fericire, am acoperit destul de multe aspecte aici, suficient pentru a crea orice tabele doriți. Deloc surprinzător, LaTeX e mult mai complex de atât, așa dar așteptați-vă la un tutorial următor cu trăsături mai avansate în viitorul apropiat.