

## **Composició de textos mitjançant Lout**

*Arnau Prat Gasull*

lout-users

Agost de 2016

### **Sinopsi**

En aquesta introducció a l'eina d'edició de textos Lout s'hi exposa el tractament adequat per obtenir un document extens, comprès d'un gran nombre de seccions i fitxers. Es tracta d'una referència simplificada per usuaris de Lout.

# Composició de textos mitjançant Lout

Arnau Prat Gasull

lout-users

Agost de 2016

## Taula de continguts

1. Recursos i informació .. .. .	2
1.1. Per què utilitzar Lout? .. .. .	2
1.2. Programari .. .. .	3
2. Codificació del document .. .. .	4
3. L'estructura de Lout—objectes i símbols .. .. .	4
4. Declaració del tipus de document .. .. .	4
5. Estructura d'un document llarg .. .. .	5
6. Seccions i subseccions .. .. .	6
7. Paràgrafs .. .. .	7
8. Referències creuades i enllaços .. .. .	9
9. Tipus i estils de tipus de lletra .. .. .	10
10. Matemàtiques i equacions .. .. .	10
10.1. Matrius .. .. .	11
10.2. Símbols matemàtics personalitzats .. .. .	11
11. Gràfics i figures .. .. .	12
12. Taules .. .. .	14
13. Compilació del document .. .. .	17
13.1. Conversió d'un fitxer PostScript a PDF .. .. .	18
14. Característiques avançades .. .. .	18
14.1. Definicions i símbols personalitzats .. .. .	18
14.2. Canvi del tipus de lletra .. .. .	19
<b>Referències .. .. .</b>	<b>21</b>

## 1. Recursos i informació

A [2], trobareu citat el recurs al qual ha de recórrer un usuari de Lout quan li apareix un problema. La quantitat d'informació exposada i la qualitat de la redacció converteix el document citat en la millor guia de l'eina per l'usuari. Per a usuaris més avançats, el mateix autor del programa ha publicat [1], que permet conèixer símbols que són presents en nivells més baixos.

Aquesta guia exposa els diferents passos que ha de seguir un usuari des de la creació d'un fitxer en blanc fins a obtenir el document preparat per publicar. Un usuari novell pot tractar la guia com un tutorial o introducció a l'eina Lout.

### 1.1. Per què utilitzar Lout?

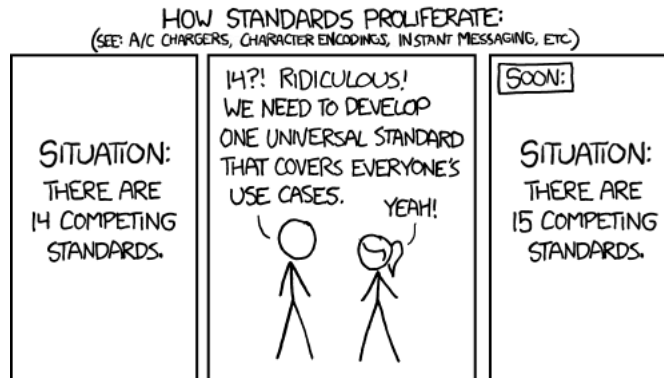
Lout, a diferència d'altres opcions per crear documents i informes, té les següents característiques:

- Compta amb documents de suport admirables.
- El compilador és prou explícit amb els errors.
- El sistema Lout ocupa una fracció del que ocupa  $\text{\LaTeX}$ .
- El sistema crea documents de qualitat.
- El llenguatge és consistent i coherent per crear textos, gràfics, matemàtiques i taules.
- S'ofereixen un gran nombre de plantilles per finalitats molt diverses. Plantilles per crear articles, presentacions i informes ja venen per defecte.
- El sistema permet automatització i és prou bo per documents de gran complexitat.
- Es garanteix la obtenció d'un fitxer PostScript o PDF d'un document creat mitjançant Lout d'aquí molts anys, mentre el programa Lout encara sigui compilable.
- Si no es produeixen gaires errors, utilitzar Lout és més productiu que utilitzar  $\text{\LaTeX}$ .

Tanmateix, té els inconvenients següents:

- Tots els documents han d'estar codificats amb ISO-LATIN-1 i no amb UTF.
- El sistema pot tenir problemes amb algunes llengües.
- Pocs usuaris utilitzen el sistema.
- Poca informació i recursos d'ajuda a la xarxa disponibles.
- Crear documents PDF de manera nativa ofereix pitjors resultats que crear-los a partir del fitxer PostScript obtingut prèviament amb el compilador. El document PostScript que es pot obtenir nativament és de bona qualitat, tanmateix.
- L'eina no presentarà actualitzacions per part de l'autor original a favor del nou projecte Non-pareil.

Per a molts usuaris els inconvenients guanyen als avantatges de manera que aquesta guia no té sentit. Si el lector creu que és interessant provar d'escriure un document amb aquest sistema peculiar, s'ha de dir que conèixer el paradigma i les maneres de fer és gratificant. Al final, el lector acaba amb la sensació que s'haurien de combinar els avantatges que ofereixen les diferents eines com L<sup>A</sup>T<sub>E</sub>X i Lout per desenvolupar un sistema de creació de fitxers sublim i de referència.



**Figura 1.** Per què s'hauria de crear un nou sistema de creació de documents?

L'autor va començar a utilitzar Lout en reflexionar com seria una alternativa vàlida i a l'altura de L<sup>A</sup>T<sub>E</sub>X, i pot proporcionar una llista de sistemes alternatius a T<sub>E</sub>X si el lector ho demana. S'ha de dir que Lout és la segona o tercera opció més consolidada, doncs no n'hi ha gaires de bones alternatives.

## 1.2. Programari

Lout pot ser usat tant en Ubuntu (i altres distribucions Linux) com en Windows. La descàrrega de Lout per a Windows es pot fer a través de Sourceforge<sup>1</sup>. També podeu compilar Lout perquè funcioni en aquesta plataforma, però això requereix més esforç, de la mateixa manera que si ho feu per Ubuntu. La manera més fàcil per obtenir Lout en aquesta última és fent-ho amb el gestor de paquets o `sudo apt install lout`, però la versió proporcionada pels repositoris oficials del sistema podria no correspondre's amb la última versió estable del programari.

Per visualitzar els fitxers PostScript, es requereix un intèrpret d'aquest llenguatge, com per exemple Ghostscript. Mentre que al sistema operatiu Ubuntu podria ser que ja estigués instal·lat, a Windows caldrà que es realitzi l'instal·lació manualment. Podeu utilitzar els paràmetres predeterminats per fer la instal·lació. A la secció *Compilació del document* hi trobareu més informació relacionada.

<sup>1</sup> A l'enllaç <https://sourceforge.net/projects/lout/files/lout-win32/> hi trobareu Lout compilat per a Microsoft Windows. La carpeta `lout` s'ha de posar al disc `C:\` de manera que `lout.exe` és a `C:\lout\bin`.

## 2. Codificació del document

Lout no treballa amb codificació Unicode (UTF), fet que pot causar problemes inicialment amb alguns caràcters, com per exemple els accentuats, els que presenten dièresis, la ç i la l·l presents en el català. Per utilitzar l'eina en aquesta llengua, cal guardar el fitxer amb la codificació<sup>1</sup> ISO-8859-1<sup>2</sup>. Altres codificacions queden fora de l'àmbit d'aquesta guia, però també poden ser utilitzades amb Lout<sup>3</sup>.

## 3. L'estructura de Lout—objectes i símbols

Lout treballa essencialment amb objectes, que poden ser caràcters, paraules, paràgrafs, pàgines... Aquests objectes poden ser modificats per símbols, una de les múltiples funcions que poden tenir. Els símbols s'obtenen mitjançant definicions (vegeu *Definicions i símbols personalitzats*), de manera que un usuari mitjanament avançat pot crear-ne de nous. Els símbols acostumen a presentar la forma `@HMirror MIRALL`, que produeix `ЛИАЯИМ`, o bé

`30d @Rotate { Hola, món },` que produeix el resultat *Hola, món*. Noteu que els símbols s'escriuen amb `@`—tot i que n'hi poden haver sense—i poden prendre una o dos variables a la dreta i/o a l'esquerra. Alguns símbols, com `@Section` presenten variables opcionals. A la secció especialitzada hi trobareu més informació.

## 4. Declaració del tipus de document

A l'hora de crear un text mitjançant Lout cal definir quin tipus de document es redactarà. Entre els tipus predefinits per l'eina, hi ha les següents classes:

- Document ordinari o **doc**
- Informe o **report**
- Llibre o **book**
- Transparències o **slides**

Pel fet que aquesta guia està orientada a la redacció de documents amb una gran quantitat d'informació, s'utilitzarà el tipus de document **report** per descriure les característiques d'un fitxer processat per l'eina.

Per utilitzar una classe predeterminada s'ha d'emprar `@SysInclude { classe },` on **classe** és un fitxer que conté les definicions per al tipus de document i es troba a `/usr/share/lout/include` o `/home/usuari/lout.lib/include`<sup>4</sup>. La declaració del tipus de

<sup>1</sup>L'opció de codificació acostuma a trobar-se a la finestra que apareix quan s'anomena i desa el fitxer des de l'editor.

<sup>2</sup>ISO-LATIN-1 o *Western*, alternativament.

<sup>3</sup>Hi ha caràcters que no estan definits a la norma ISO-8859-1, com €, que s'han de definir mitjançant `@Euro`.

<sup>4</sup>El nom dels directoris pot canviar en funció de la instal·lació de Lout en el sistema. En el primer cas, s'ha utilitzat **sudo apt install lout** amb Ubuntu 16.04 LTS (Xenial Xerus), mentre que la segona ubicació és conseqüència de la compilació del codi font de l'eina i utilitzant la majoria de paràmetres per defecte.

document implicarà una estructura del document font particular. L'estructura del fitxer font per un informe ha de ser la que hom pot trobar a continuació, basada en els exemples de [2].

```
@SysInclude { report }
@Report
  @Title { ... }
  @Author { ... }
  @Institution { ... }
  @Abstract { ... }
//
@Section
  @Title { ... }
  @Begin
  ...
  @End
@Section
```

Si es vol definir un nou tipus de document **novaclasse**, en comptes d'utilitzar **@SysInclude { report }** cal utilitzar **@Include { novaclasse }**. El símbol **@SysInclude { ... }** fa la cerca del fitxer especificat entre claudàtors a la carpeta **/usr/share/lout/include**, mentre que **@Include { ... }** fa la cerca al directori de treball. També es pot especificar la ubicació del fitxer, utilitzant **@Include { "carpeta/..." }** (pareu atenció a les cometes).

## 5. Estructura d'un document llarg

No és recomanable escriure un document de dimensions prou considerables en un mateix document. És convenient tenir un document que actuï com a base o marc on s'hi referencien els fitxers que contenen el text corresponent a un capítol o una secció. Això facilita enormement la redacció del document ja que els continguts estan repartits en temàtica i el programa d'edició només ha de carregar la fracció desitjada del document, i no pas la totalitat. En aquest document una part dels fitxers s'han referenciat de la manera que es mostra:

```
@Report
...
//
@Include { "conti/infor" }
@Include { "conti/codif" }
@Include { "conti/simbo" }
@Include { "conti/docti" }
@Include { "conti/llarg" }
...
```

Noteu que els fitxers **infor**, **codif**, **simbo**, **docti** i **llarg** es troben en una subcarpeta anomenada **conti**, de continguts. La carpeta es troba en el mateix directori que el fitxer base que inclou el codi presentat anteriorment. En els fitxers esmentats hi ha el text d'aquest document: a **simbo** s'expliquen els símbols de Lout, a **codif** es parla de la codificació dels documents, mentre que **llarg** conté aquest mateix text que esteu llegint.

Des del punt de vista del compilador utilitzar el símbol **@Include** no representa cap problema en absolut. De fet, tot i utilitzar el sistema de fitxers que s'ha explicat en el paràgraf anterior el compilador més veu la totalitat del document. És a dir, en compilar el fitxer base, l'eina ja inclou el contingut dels fitxers referenciats al marc.

Això vol dir que no hi ha problema en cridar etiquetes d'elements d'altres fitxers—sempre que s'hagin declarat abans de la crida—doncs el compilador veu el conjunt del text.

## 6. Seccions i subseccions

Quan s'escriu un informe mitjançant les definicions de **report**, l'estructura de les seccions no necessiten ser incloses entre els símbols **@BeginSections** i **@EndSections**. S'anomena símbol a tot aquell element que vingui descrit amb **@** immediatament seguit de text, com per exemple **@PP**, un símbol que permet definir un nou paràgraf.

```
...
@Report
...
//
@section
    @Title { Lorem ipsum }
@Begin
@PP
...
@end @Section
# Nova secció
@section
    @Title { Dolor sit amet }
@Begin
@PP
...
@end @Section
```

Per documents que no utilitzin **report**, cal utilitzar els símbols **@BeginSections** i **@EndSections** per obtenir seccions correctament, tal i com es mostra a continuació.

```
@BeginSections
    @Section
        @Title { Dolor sit amet }
    @Begin
    @PP
    ...
    @End @Section
    # Es poden afegir més seccions
@endSections
```

Per definir subseccions a l'informe o a qualsevol altre tipus de document, cal emprar la sintaxi

ordinària. Les subseccions hauran de definir-se entre `@BeginSubSections` i `@EndSubSections`, aquestes situades dins de la secció, tal i com es mostra. D'una manera similar, `@SubSubSection` s'ha de situar entre `@BeginSubSubSections` i `@EndSubSubSections`, que al seu torn es troba dins de la subsecció.

```
@Section
  @Title { Lorem ipsum }
@Begin
  @BeginSubSections
    @SubSection
      @Title { Dolor sit amet }
    @Begin
    @PP
    ...
  @End @SubSection
  # Es poden afegir més subseccions
@EndSubSections
@End @Section
# Es poden afegir més seccions
```

Tal i com succeeix amb les seccions, els apèndixs d'un informe no s'han d'incloure entre de `@BeginAppendices` i `@EndAppendices`, però sí que s'han de tenir en compte a l'hora de definir `@SubAppendix`. En qualsevol altre tipus de document, el símbol anomenat s'ha de tenir en compte.

## 7. Paràgrafs

El text dels escrits s'organitza en paràgrafs, que són les estructures on es desenvolupen cada una de les idees del document. Lout permet obtenir diferents estils de paràgraf amb certa facilitat.

A diferència de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  els paràgrafs no s'obtenen deixant una línia nova al fitxer font, sinó que s'ha d'escriure l'identificador del paràgraf en el punt on es vol començar el paràgraf. Per cada tipus de paràgraf existeix un identificador, l'efecte del qual es propagarà en tot el text present entre aquest i el pròxim identificador. Mai caldrà afegir l'identificador darrere d'estructures com llistes o figures, doncs s'entén que aquests formen part del text i ja produeixen els espais necessaris al fitxer de sortida.

En la majoria de casos, per començar un paràgraf només cal afegir `@PP Dolor sit amet, consectetur...` per obtenir un nou paràgraf. Supposeu un informe constituït per diferents seccions. L'estructura d'una de les seccions podria ésser la que el lector pot trobar a continuació. Noteu que els identificador són `@PP` i el seu àmbit d'efecte és obvi.

```
@Section
  @Title { Lorem ipsum }
@Begin
@PP
```



Dolor sit amet, consectetur adipiscing elit. Nulla ac venenatis orci. Pellentesque accumsan dolor nec diam cursus aliquam. Nunc.

@PP

Curabitur sit amet faucibus lorem. Duis ultrices mauris vel dui egestas, at tincidunt arcu egestas. Proin a lacus lacinia, sagittis.

@PP

Ut auctor a tellus in porta. Nullam imperdiet sollicitudin erat. Suspendisse lacinia at sem consectetur sagittis. Pellentesque fermentum ipsum nec.

@End @Section

Diferents identificadors, altrament anomenats símbols, permeten obtenir diferents tipus de paràgrafs. A sota hi trobareu un conjunt de símbols que us poden ésser útils.

- Per un paràgraf que produeixi el mateix efecte que @PP però que no tingui indentació cal utilitzar @LP.
- Per remarcar la importància d'un paràgraf, cal utilitzar @DP. Aquest símbol defineix un paràgraf amb més espai vertical que els altres tipus.

Per modificar la posició del text en casos puntuals podem utilitzar el conjunt de símbols @Display. Amb aquest símbol es poden obtenir resultats com el que hi ha a continuació:

Text a destacar.

Els símbols disponibles són @Display, @LeftDisplay, @IndentedDisplay, @QuotedDisplay, @CenteredDisplay o @RightDisplay. Tots aquests símbols menys l'últim es poden simplificar amb @D, @LD, @ID, @QD i @CD respectivament. Tingueu en compte que es poden definir nous símbols com a abreviacions en cas que fos necessari.

Per obtenir tot tipus de llista s'ha de repetir el patró següent:

```
@List
@ListItem { Cotxe }
@ListItem { Avió }
@EndList
```

Amb el codi anterior s'obté el resultat mostrat:

- 1 Cotxe
- 2 Avió

Com abans, també existeixen múltiples símbols amb les corresponents abreviacions per a les llistes. Entre les disponibles, destaquem @List o @L, @CenteredList o @CL i @BulletList o @BL. S'obtenen subllistes incloent el patró anterior en un @ListItem, afegint Raw o R entre @ i

el tipus de llista<sup>12</sup>, com es mostra a [2, pàgina 33]. En els casos que es vulgui enumerar diferents paràgrafs s'ha d'utilitzar **@ParagraphItem**.

Si s'utilitza **@TaggedList** o **@TL** ja no es pot utilitzar **@ListItem**, sinó que s'ha d'utilitzar **@TagItem { etiqueta } { descripció }**. Fent els canvis indicats es podran obtenir resultats com els mostrats:

Cotxe      Vehicle amb motor de quatre rodes.

Avió      Aeronau, més pesant que l'aire, capaç de volar per l'atmosfera.

En aquest cas hem utilitzat **@WideTaggedList** o **@WTL** en comptes de **@TaggedList** o **@TL** per obtenir més espai entre l'etiqueta i la descripció. També hi ha la opció **@VeryWideTaggedList** o **@VWTL** per aconseguir encara més espai.

Una nota al peu<sup>3</sup> s'obté amb **@FootNote { Com aquesta. }**. Hi ha múltiples opcions per configurar tant la posició com el símbol, però queden fora de l'abast de la guia. A la guia [2] s'exposen les configuracions disponibles pels símbols anteriors i apareixen nous símbols que poden ser d'especial interès.

## 8. Referències creuades i enllaços

Per fer referència a alguna part del text, com per exemple a la secció *Referències creuades i enllaços* de la pàgina 8, podem emprar els símbols **@TitleOf { etiqueta }** o **@NumberOf { etiqueta }**, on l'etiqueta és característica de la part a referenciar.

Les etiquetes poden definir-se dins de qualsevol text amb **@PageMark referencia.a**. Aquest símbol permet referenciar la pàgina sense haver d'escriure el número directament. El número es pot recuperar amb **@PageOf referencia.a**, que produeix 9.

Ara bé, si en comptes de pàgines volem fer referència a capítols, seccions, peus de pàgina, elements de llista, figures o taules, podem utilitzar la opció **@Tag { etiqueta }** que proporciona. Prenent com a exemple una secció<sup>4</sup>, tenim el següent:

```
@Section
  @Title { Títol de la secció }
  @Tag { referencia.b }
@Begin
@PP
...
```

Podem recuperar el número de la secció amb **@NumberOf { referencia.b }**. En el cas dels símbols que presenten l'opció **@Title { ... }**, com capítols i seccions, també podem recuperar el títol amb **@TitleOf { referencia.b }**.

---

<sup>1</sup>Així, obtnidrem **@RawList**.

<sup>2</sup>Si hem usat abreviacions utilitzarem **R** menys per a **@L**.

<sup>3</sup>Com aquesta

<sup>4</sup>El patró que es mostra també funciona per tots els símbols que fan referència als elements exposats.


Les referències creuades requereixen tractar el fitxer múltiples vegades, doncs amb una execució de `lout` no es resolten les etiquetes. L'autor utilitza `lout -r2 -o 0.ps 0` per obtenir una versió PostScript del document. A vegades, amb dues execucions (donades per `-r2`) no és suficient. Llavors cal canviar el valor exposat.

Per utilitzar adreces URL en un document, el més recomanable és utilitzar `http://savannah.nongnu.org/projects/lout @ExternalLink { Lout - Summary }`, que produeix Lout - Summary. Lout també ofereix enllaços interns mitjançant l'ús d'etiquetes, que podeu consultar a la guia de l'autor de Lout.

## 9. Tipus i estils de tipus de lletra

En molts casos l'usuari voldrà poder disposar de lletres en **negreta**, *itàliques* o una **combinació dels dos estils**. En aquests casos s'han demprat els símbols `@B negreta`, `@I itàliques` o `@BI { combinació dels dos estils }`, segons es desitgi.

També hi ha estils de `VERSALETES`, que s'obtenen amb `@S versaletes` o la variant romana `@R romana`, que no té sentit si la font base del document és una lletra amb serifes. És útil per tornar a la font base si el text és en itàliques.. Amb `@I @S { versaletes i itàliques }` s'obtenen lletres en *VERSALETES I ITÀLIQUES*

El tipus de lletra es pot canviar fàcilment, però les opcions són més limitades que les que ofereixen altres programaris. Durant l'última compilació del document, les opcions disponibles eren AvantGarde, **Bookman**, Chancery, Courier, Helvetica, Schoolbook, Palatino, Σμβολ (Symbol), Times i  (Dingbats). Totes aquestes fonts compten amb un nombre més o menys important d'estils entre els que seleccionar, com **Base**, **Bold** o **Slope**. A [2, pàgina 14] s'exposen tots els estils, mentre que a la subsecció 14.2 s'explica com utilitzar tipus de lletra propis.

## 10. Matemàtiques i equacions

La representació de símbols matemàtics es pot aconseguir afegint `@SysInclude { eq }` al principi del document. Certs caràcters matemàtics i expressions es podrien obtenir sense l'addició d'aquesta línia, però expressions matemàtiques més complexes la requereixen.

Les expressions matemàtiques en línia com  $2 + 2 = 5$  s'obtenen amb `@E { 2+2=5 }`. `@E { ... }` és una variant de `@Eq { ... }`, el símbol per defecte per expressions matemàtiques, que evita que una expressió matemàtica pugui separar-se en dos línies. Per altra banda, les expressions a línia completa (*display equations*, en anglès) com

$$2 + 2 = 5$$

o

$$2 + 2 = 5$$

s'obtenen amb `@CD @Eq { 2+2=5 }` o `@ID @Eq { 2+2=5 }` respectivament.

Per enumerar les equacions podem utilitzar `@NumberedDisplay @Eq { 2+2=5 }` o l'expressió abreujada corresponent, `@ND @Eq { 2+2=5 }` per obtenir l'equació desitjada.

$$2 + 2 = 5 \quad (10.1)$$

Les equacions es poden referenciar amb el símbol `@Tag { dosdoscinc }` i es pot obtenir el número de l'equació mitjançant `@NumberOf { dosdoscinc }`, com es pot apreciar en el cas de l'equació 10.1.

```
@ND
  @Tag { dosdoscinc }
  @Eq { 2+2=5 }
```

Per conèixer els símbols per crear les expressions consulteu [2, pàgina 156].

### 10.1. Matrius

La sintaxi per crear matrius a Lout podria semblar poc intuïtiva. A continuació es mostra el codi font i la matriu resultant.

```
@Eq{ pmatrix { row col a col b col c
               row col d col e col f
               row col g col h col i } }
```

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

Múltiples matrius adjuntes poden fer evidents diferències entre elles, com es mostra. Noteu que l'altura de les matrius és diferent.

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix}$$

Afegint `strut { yes }` entre `pmatrix` i el primer claudàtor de cada una de les matrius, s'obté un resultat millor.

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} j & k & l \\ m & n & o \\ p & q & r \end{pmatrix}$$

### 10.2. Símbols matemàtics personalitzats

Quan hi ha una expressió que apareix en multitud d'ocasions resulta convenient disposar d'un símbol que la representi cada vegada que s'invoqui. Seguint les explicacions de la secció *Definicions i símbols personalitzats*, podem definir aquest tipus de símbols. A [2, pàgina 155] s'hi exposa un exemple similar al que trobareu a continuació.

```
import @Eq # Importació dels símbols de @Eq per un ús dins de la funció
def baselog left x right y # Nom de la funció i els paràmetres que pren
{ p sub y log sub x p sub y } # Descripció de la funció, entre claudàtors
```

Aquesta expressió permetrà escriure  $p_i \log_2 p_i + p_j \log_3 p_j + p_k \log_4 p_k$  fàcilment mitjançant `@E { 2 baselog i + 3 baselog j + 4 baselog k }`. Cal dir que **baselog** només es renderitzarà correctament dins de l'entorn `@Eq { ... }`. Si emprem el símbol **baselog** fora d'aquest entorn<sup>1</sup>, com `2 baselog i` directament, es renderitzarà com  $p_i \log_2 p_i$ , que no és el resultat que busquem.

## 11. Gràfics i figures

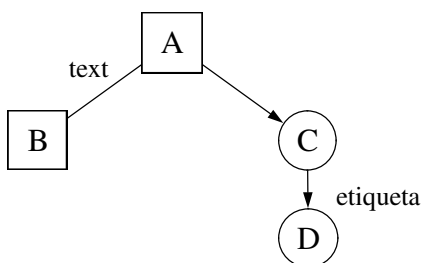
Lout proporciona eines per aconseguir un control més o menys complex sobre l'estètica dels elements que constitueixen els documents. Amb `{ rgb 0 0 1.0 } @Colour { text en blau }` obtenim, òbviament, **text en blau**. A [2, pàgina 162] s'especifiquen els colors predefinits que el símbol `@Colour` entén i que es poden substituir per el conjunt `{ rgb 0 0 1.0 }`. També es pot utilitzar el model de color CMYK amb **cmk** en comptes de **rgb**, afegint un altre argument per descriure correctament el color a utilitzar.

També s'ofereix la possibilitat d'utilitzar textures, amb resultats com **aquest**. La utilització de textures amb `@Texture`, com en el cas de `@Colour`, no es limita només a text i es pot utilitzar en diferents elements. El lector interessat pot consultar-ne l'ús i les diferents opcions a [2, pàgines 163 i 164].

Es pot encaixar el text amb **una caixa** mitjançant `@Box { una caixa }`. De fet, qualsevol objecte que segueixi `@Box` apareixerà encaixat. Aquest símbol disposa de múltiples opcions d'especial interès, que es poden consultar a [2, pàgina 167]. Utilitzant `@CurveBox arrodonits` i `@ShadowBox ombra` s'obté una caixa amb els vèrtexs **arrodonits** o amb una **ombra** de la caixa, respectivament.

Hi ha un munt de símbols, com `@Rotate` o `@HMirror`<sup>2</sup>, que permeten satisfer les necessitats de l'usuari. Aquesta guia no és una referència per un usuari quotidià, sinó una introducció per a un usuari inexpert que necessiti escriure un document per primera vegada. Per resoldre qualsevol problema relacionat amb l'ús de Lout el més assenyat és dirigir-se a [2] i no a aquesta guia.

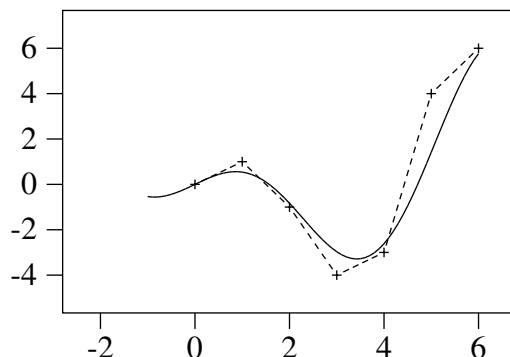
Lout proporciona símbols per crear organigrames, gràfics matemàtics de funcions i dades i diagrames de barres o circulars amb certa facilitat amb les llibreries **diag**, **graph** i **pie**, respectivament. A continuació es mostren alguns exemples de l'autor basats en [2] per tal que el lector copsi els resultats que es poden obtenir amb relativament poc esforç.



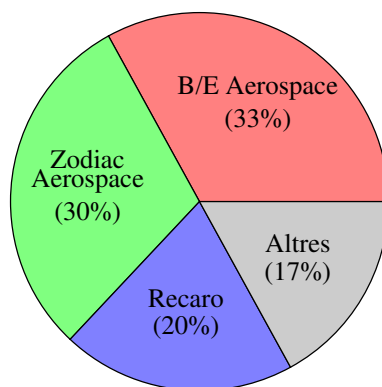
<sup>1</sup>Com que **baselog** és un símbol a tot el document, no es pot restringir el seu ús només a l'àmbit que li correspondria, de manera que fora de l'àmbit pel qual ha estat pensada es pot utilitzar però s'obindrà un resultat diferent.

<sup>2</sup>Aquests símbols s'han presentat a *L'estructura de Lout—objectes i símbols*.

Gràfic amb  $x \cdot \cos\left(\frac{180x}{\pi}\right)$



Quota de mercat de la competència



Evidentment és recomanable utilitzar i tractar les dades amb les eines que proporciona Lout per obtenir un document consistent estèticament. Tanmateix, és evident que la majoria de lectors preferirien o necessitaran utilitzar les seves eines per aconseguir el resultat que es busca. Lout permet incloure il·lustracions produïdes amb eines externes. Aquestes han de ser forçosament fitxers EPS (Encapsulated PostScript), fet que es pot presentar com un inconvenient<sup>1</sup>. A continuació s'explica com incloure un fitxer EPS al document.

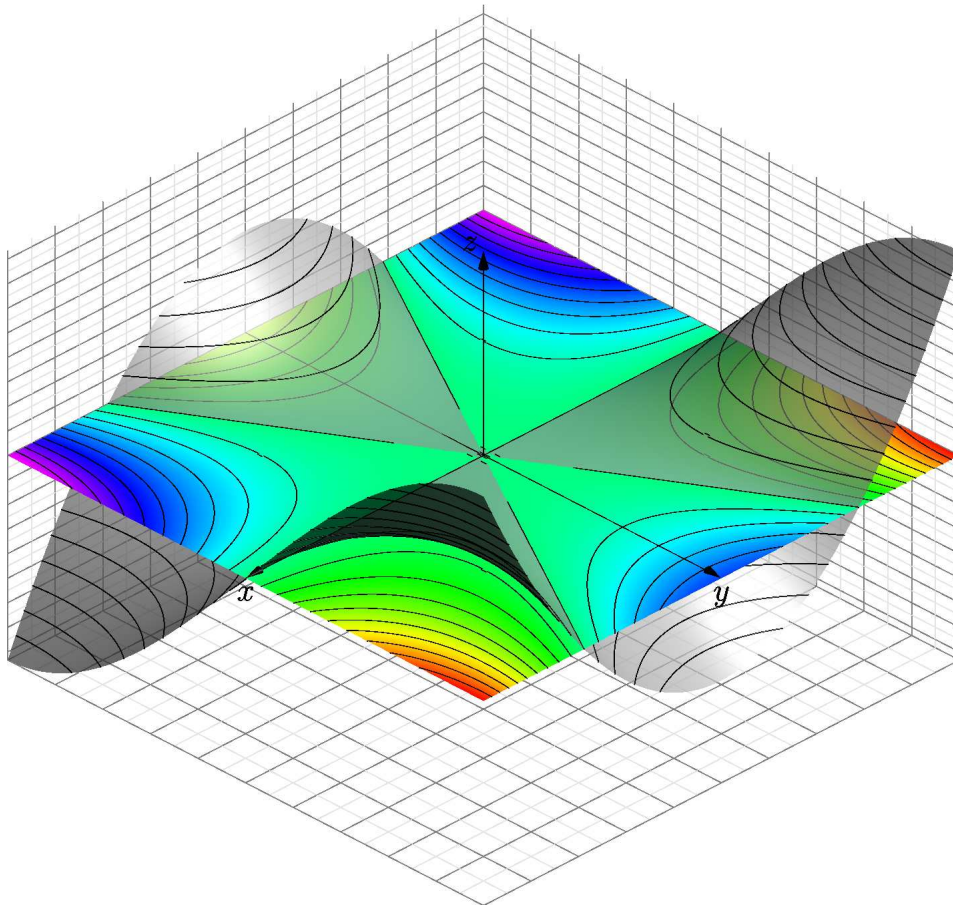
Amb `@IncludeGraphic plot.eps`, el gràfic `plot.eps` apareixerà en el punt on s'hagi invocat el símbol. En casos on una imatge s'hagi de repetir múltiples vegades i aquesta tingui un pes excessiu, cal afegir `@IncludeGraphicRepeated { plot.eps }` a la zona d'inclúderes, sota la definició de la classe del document. Per invocar la imatge repetida, utilitzant el símbol `@IncludeGraphic plot.eps` és suficient.

Les figures s'obtenen amb el símbol `@Figure`, que presenta les mateixes característiques que `@Table`. Aquests símbols provoquen que el que el segueix apareixi on s'hagi configurat. Per al fragment següent s'obté la figura 2.

---

<sup>1</sup>Si l'eina utilitzada per produir l'il·lustració no exporta a EPS, caldrà utilitzar algun conversor o encapsulador per tal que Lout pugui utilitzar fitxers amb altres extensions.

```
@Figure
  @Caption { Presentació de plot.eps }
  @Tag { plot }
  @Location { Display }
  @IncludeGraphic plot.eps
```



**Figura 2.** Presentació de `plot.eps`

## 12. Taules

Les definicions per les taules es troben en un fitxer anomenat `tbl`, de manera que per utilitzar-les hem d'afegir al principi del document la línia `@SysInclude { tbl }`. Les taules donades per aquestes definicions poden tenir un màxim de 26 columnes, tal i com es veurà en els exemples.

Per crear una taula correctament, l'estructura és la que hi ha a continuació.

```
@Table
  @Caption { Una taula }
  @Tbl
    aformat { @Cell @B A | @Cell @B B | @Cell @B C }
    bformat { @Cell A | @Cell B | @Cell C }
```

```

cformat { @Cell @I A | @Cell @I B | @Cell @B C }
rule { yes }
{
@Rowa
  A { Cel·la a A }
  B { Cel·la a B }
  C { Cel·la a C }
@Rowb
  A { Cel·la b A }
  B { Cel·la b B }
  C { Cel·la b C }
@Rowb
  A { Cel·la b A }
  B { Cel·la b B }
  C { Cel·la b C }
@Rowc
  A { Cel·la c A }
  B { Cel·la c B }
  C { Cel·la c C }
}

```

<b>Cel·la a A</b>	<b>Cel·la a B</b>	<b>Cel·la a C</b>
Cel·la b A	Cel·la b B	Cel·la b C
Cel·la b A	Cel·la b B	Cel·la b C
<i>Cel·la c A</i>	<i>Cel·la c B</i>	<b>Cel·la c C</b>

**Taula 1.** Una taula

També podem definir els estils a cada fila en comptes de definir-los amb anterioritat. Acostuma a ser recomanable l'ús dels estils com en el cas anterior.

```

@Table
  @Caption { Una altra taula }
@Tbl
  rule { yes }
{
@Row
  format { @Cell @B A | @Cell @B B | @Cell @B C }
  A { Cel·la a A }
  B { Cel·la a B }
  C { Cel·la a C }
@Row
  format { @Cell A | @Cell B | @Cell C }
  A { Cel·la b A }
  B { Cel·la b B }
}

```



```

C { Cel·la b C }
@Row
format { @Cell A | @Cell B | @Cell C }
A { Cel·la b A }
B { Cel·la b B }
C { Cel·la b C }
@Row
format { @Cell @I A | @Cell @I B | @Cell @B C }
A { Cel·la c A }
B { Cel·la c B }
C { Cel·la c C }
}

```

Cel·la a A	Cel·la a B	Cel·la a C
Cel·la b A	Cel·la b B	Cel·la b C
Cel·la b A	Cel·la b B	Cel·la b C
<i>Cel·la c A</i>	<i>Cel·la c B</i>	<b>Cel·la c C</b>

**Taula 2.** Una altra taula

Òbviament també podem fer taules més complexes. Considereu el següent cas de cel·les que n'ocupen d'altres.

```

@Table
@Caption { Una taula més }
@CaptionPos { Above }
@Location { Display } # Vegeu uns paràgrafs més endavant
@Tbl
rule { yes }
{
@Row
format { @Cell @B A | @Cell @B B | @StartVSpan @Cell @B C }
A { Cel·la a A }
B { Cel·la a B }
C { Cel·la a C }
@Row
format { @StartHSpan @Cell A | @HSpan | @VSpan }
A { Cel·la b A }
@Row
format { @Cell A | @StartHVSpan @Cell B | @HSpan }
A { Cel·la b A }
B { Cel·la b B }
@Row
format { @Cell @I A | @VSpan | }

```

```
A { Cel·la c A }  
}
```

**Taula 3.** Una taula més

Cel·la a A	Cel·la a B	Cel·la a C
Cel·la b A		
Cel·la b A	Cel·la b B	
Cel·la c A		

Aquestes indicacions permeten solucionar una gran diversitat de demandes. Tanmateix, per documents més complexos on es requereixin taules que perdurin múltiples pàgines o que tinguin només algunes línies de vora pintades, l'usuari haurà de recórrer a la guia recomanada.

La posició predeterminada de les taules es pot configurar afegint la línia `@TableLocation { ... }` al fitxer d'estil del document en comptes d'escriure `@Location { ... }` a cada una de les taules. Les opcions més utilitzades són **Display** i **Page-Top**. També hi ha les opcions **FullPage**, **PageFoot**, **ColTop**, **ColFoot**, **ColEnd**, **TryAfterLine** o **Display**, entre d'altres. Les opcions que tenen en compte la columna són molt útils en documents amb més d'una columna de text o amb múltiples capítols. En aquest document s'ha utilitzat **Display**.

### 13. Compilació del document

Per obtenir un document preparat per imprimir cal executar `lout -o output.ps input` on `output.ps` és un fitxer PostScript i `input` és el fitxer font. Per visualitzar el fitxer Postscript és necessari el programari adequat, com per exemple Ghostscript<sup>1</sup>. Lout també pot crear fitxers PDF mitjançant `lout -z -o output.pdf input`. Es recomana crear un fitxer PostScript i convertir-lo a PDF, ja que durant la creació directa d'un fitxer PDF algunes característiques de Lout no són renderitzades correctament. Vegeu Conversió d'un fitxer PostScript a PDF per l'explicació de mètodes de conversió. Com diu a [2, pàgina 91], per generar un fitxer PostScript tampoc no hauriem d'utilitzar mai **-EPS**<sup>2</sup>.

Un document mitjanament complex requerirà més d'una execució de Lout per obtenir el fitxer desitjat. Aquestes múltiples execucions permeten solucionar les referències creuades<sup>3</sup>, índexs i bases de dades. En comptes d'executar l'eina múltiples vegades manualment, també podem executar `lout -r2 -o output.ps input`. `-r2` indica que es faran dues passes (de

<sup>1</sup>A Ubuntu Xenial Xerus ja hi és present, mentre que a Microsoft Windows s'ha d'instal·lar per compte propi. Una vegada instal·lat Ghostscript en aquest sistema operatiu, podeu instal·lar SumatraPDF, que facilita la manipulació del fitxer, com per exemple la conversió a PDF.

<sup>2</sup>Un fitxer EPS (*Encapsulated PostScript*) només està constituït per una pàgina, de manera que l'execució de `lout` amb **-EPS** produirà un fitxer amb només la primera pàgina, denotat al terminal com a **fatal error: truncating -EPS document at end of first page**.

<sup>3</sup>Apareixen com a **línia,columna: unresolved cross reference** al terminal quan no han sigut solucionades.

l'anglès, *r* de *run*) i `-r4`, que s'en faran quatre. Això és útil per documents complexos que requereixen múltiples passades.

En algunes ocasions, Lout no aconsegueix resoldre el fitxer font encara que hagim rectificat els possibles errors que hi puguin haver. En casos on la sintaxi és correcta però l'eina retorna error, un dels problemes més comuns és que els fitxers d'altres execucions encara es mantenen. Per tal que Lout torni a crear tots aquests fitxers haurem d'esborrar manualment aquells que acaben amb `.li`, `.lix` i altres que no ens resulten familiars. Una vegada esborrats, l'execució hauria de ser fluida i no retornar cap error.

### 13.1. Conversió d'un fitxer PostScript a PDF

Un mètode fàcil per convertir un fitxer PostScript a PDF és utilitzant l'eina `ps2pdf`, que funciona al terminal. Per obtenir el fitxer `.pdf` només cal que obriu el terminal i executeu `ps2pdf input.ps output.pdf`, on `input` i `output` són els noms dels fitxers. El fitxer `.pdf` apareixerà al mateix directori que el fitxer `.ps`.

Si utilitzeu el visualitzador SumatraPDF per veure els fitxers `.ps`<sup>1</sup> podeu utilitzar l'element del menú File → Save As... i escollir el tipus de fitxer en el menú d'opcions, entre les quals hi ha PDF.

També podeu aprofitar les capacitats del vostre lector PostScript per imprimir a un fitxer PDF directament.

## 14. Característiques avançades

### 14.1. Definicions i símbols personalitzats

Lout ofereix la possibilitat de definir nous símbols. Les definicions s'han de redactar en un fitxer situat al mateix directori que el document base anomenat `definicions`. Per tal que aquestes definicions dels símbols siguin accessibles a tot el document, s'ha d'incloure la línia `@Include { definicions }` a sobre de la declaració del tipus de document<sup>2</sup>.

Una vegada creat el fixer, podem definir símbols com els que es poden trobar a *Símbols matemàtics personalitzats* i altres. La definició d'un nou símbol segueix l'estructura que hi ha a sota (extret de *Símbols matemàtics personalitzats*).

```
import @Eq # Importació dels símbols de @Eq per un ús dins de la funció
def baselog left x right y # Nom de la funció i els paràmetres que pren
{ p sub y log sub x p sub y } # Descripció de la funció, entre claudàtors
```

Per crear un símbol que introdueix el nom complet de l'autor del document amb la invocació de `@arnau` s'han d'afegir les següents línies al fitxer de definicions.

```
def @arnau
{ Arnau Prat Gasull }
```

---

<sup>1</sup>Per fer-ho, heu d'instal·lar Ghostscript anteriorment.

<sup>2</sup>Si es crea un fitxer de definicions anomenat `mydefs` en comptes de `definicions` no cal fer explícita la inclusió, Lout ja la fa automàticament.

El resultat obtingut és Arnau Prat Gasull.

Per actuar sobre l'estètica s'han de fer accessibles les definicions bàsiques del Lout a la definició, com es mostra en el codi següent.

```
import @BasicSetup
def @arnauSC
{ @ShadowBox { Arnau @S { Prat Gasull } } }
```

S'obté Arnau PRAT GASULL.

Evidentment també es pot actuar sobre un objecte. Amb el codi que hi ha a continuació podem modificar la paraula o paraules que segueixen el símbol, com `@cref Paraula` amb resultat *Paraula* o `@cref { Text alterat }` amb resultat *Text alterat*.

```
import @BasicSetup
def @cref right x
{ @I x }
```

Noteu que quan el símbol el segueix un sol objecte els claudàtors són optatius.

També es podria haver modificat la paraula precedent: en comptes d'escriure `right x`, s'hauria d'haver escrit `left x`. Lout també contempla paràmetres optatius, però ja es situen fora de l'abast d'aquesta guia.

## 14.2. Canvi del tipus de lletra

En certes ocasions es requereix l'ús de múltiples tipus de lletra. Ja sigui per disseny propi o per causa major, els tipus distribuïts amb Lout podrien no ser suficients per l'usuari. Afegir un tipus de lletra a Lout no és complicat, encara que un usuari novell podria veure's desbordat. És recomanable conèixer el domini una mica abans de seguir.

El primer pas és crear la base de dades de tipus de lletra. En el directori de treball s'ha de crear un fitxer, també amb codificació ISO-8859-1 anomenat `myfontdefs.ld` (és important el `.ld` per indicar que és una `database`) amb el contingut necessari per definir la font, com es mostra a continuació.

```
{ @FontDef
  @Tag { Overpass-Base }
  @Family { Overpass }
  @Face { Base }
  @Name { Overpass-Reg }
  @Metrics { Overpass-Reg.afm }
  @Mapping { LtLatin1.LCM }
}
```

S'ha d'emplenar el fitxer `myfontdefs.ld` com a mínim per a les fonts **Base**, **Bold** i **Slope**, les fonts mínimes necessàries per a Lout.

Per tal que una font estigui ben referenciada, s'han d'omplir correctament els camps. A `@Tag { ... }` hi ha d'haver la família seguida pel pes, ambdòs units amb guionet. És a dir, ha de contenir el camp `@Family { ... }` i el camp `@Face { ... }` units amb guionet en aquest or-

dre. A `@Metrics { ... }` s'ha de referenciar el fitxer `.afm` que conté la descripció mètrica de la font. Vegeu *Obtenció dels fitxers de font .afm i .pfa* per saber com obtenir-lo. A `@Name { ... }` s'ha d'escriure el nom de la font. El nom de la font es pot trobar al principi del fitxer `.afm` i és fàcilment identificable, tal i com es veu en l'extracte del fitxer que hi ha a continuació.

```
FontName Overpass-Reg
FullName Overpass-Reg
FamilyName Overpass
```

Pel que fa a `@Mapping { ... }`, en la majoria de casos s'utilitzarà el contingut exposat anteriorment `LtLatin1.LCM`, ja que els caràcters emprats en l'escriptura de la llengua catalana són descrits en el fitxer indicat.

Lout utilitza el fitxer `.afm` per crear les caixes on s'inseriran els caràcters, sense conèixer en cap moment quina lletra hi anirà a cada una. Tanmateix, l'encarregat de renderitzar els caràcters és un altre programa, per exemple, GhostScript. Per tal que el programa sigui capaç de mostrar els caràcters al fitxer, s'han de posar a disposició del programa els fitxers necessaris (sinó s'ha fet per defecte). Per fer-ho, s'ha d'obrir el fitxer `/usr/share/ghostscript/9.18/Resource/Init/Fontmap.GS` i afegir al final

```
/Overpass-Reg      (/home/arnau/Dropbox/lout/ediciolout/estil/Overpass-Reg.pfa)    ;
```

o l'equivalent per la font desitjada. S'ha de seguir estrictament l'espaiat,

```
/FontName<tab><tab> (/camí/del/fitxer/FontName.pfa) <tab>;
```

on `<tab>` és una tabulació. El fitxer `.pfa` l'acostuma a proporcionar l'autor de la font. En els casos que no es disposi del fitxer, consulteu *Obtenció dels fitxers de font .afm i .pfa*.

L'addició de la font s'haurà realitzat correctament si en obrir un terminal i executar `gs pr-font.ps` i llavors `/Overpass-Reg DoFont` (a `GS`), apareix una pantalla amb la font renderitzada.

Per obtenir el fitxer final, s'ha d'executar `lout -F /directori/del/fitxer/afm -D /directori/del/fitxer/myfontdefs.ld -o output.ps input.A -F` (de font) s'ha de posar el directori del fitxer `.afm`, mentre que a `-D` (de *database*), el directori de `myfontdefs.ld`.

#### 14.2.1. Obtenció dels fitxers de font .afm i .pfa

En molts casos, el proveïdor de la font ja proporciona el fitxer `.afm`. En cas que ja ens sigui proporcionat, cal revisar que la codificació sigui ISO-8859-1. En els casos que només es disposa del fitxer `.ttf` (TrueType) o `.otf` (OpenType) es pot utilitzar el programa FontForge per obtenir el fitxer `.afm` i el `.pfa` (que més endavant el necessitarem). Els passos per aconseguir-ho són els que es troben a continuació.

1. Obriu el fitxer `.ttf` o `.otf` amb FontForge.
2. Cambieu la codificació amb 

Encoding
----------

 → 

Reencode
----------

 → 

ISO 8859-1 (Latin 1)
----------------------

.
3. Obriu el quadre 

Font Info
-----------

 des del menú, emprant 

Element
---------

 → 

Font Info
-----------

. En el

quadre de diàleg, aneu a **General** i en el camp **Em Size**, canvieu el valor donat per 1000.

4. Genereu els fitxers necessaris amb **File** → **Generate Fonts**. Cal seleccionar **PS Type 1 (Ascii)** (noteu que l'extensió és **.pfa**) i a **Options,** escolliu **Output AFM**.

## Referències

- [1] Jeffrey H. Kingston. *An Expert's Guide to the Lout Document Formatting System (Version 3)*. Basser Department of Computer Science, University of Sydney, 1995.
- [2] Jeffrey H. Kingston. *A User's Guide to the Lout Document Formatting System (Version 3)*. Basser Department of Computer Science, University of Sydney, 1995.