BASIX: curso básico de LaTeX

Actividad formativa FDI-UCM

Oficina de Software Libre y Tecnologías Abiertas

OFICINA DE **SOFTWARE LIBRE**
VICERRECTORADO DE TECNOLOGÍAS DE LA INFORMACIÓN
UNIVERSIDAD **COMPLUTENSE** MADRID

Tema 8
-Código listing-

Conceptos que se aprenderán

En este tema de introducción se aprenderán los siguientes conceptos:

- El entorno verbatim.
- Usando listings para resaltar el código.
- Importando el código.
- Dando estilo al código.
- Listado y nombre del código.
- Poner palabras clave.

Índice

1. El entorno verbatim

El entorno que por defecto utiliza LaTeX para mostrar el código es el verbatim. Genera una salida de código monoespaciada. Para comenzar este entorno comenzaremos con el comando `\begin{verbatim}` y terminaremos con el comando `\end{verbatim}`, y entre medias de ambos colocaremos nuestra linea de código. Para tenerlo más claro vamos a poner un ejemplo:

```
Este texto mantiene los comandos \textbf{variados}
y se ignoran \LaTeX{}.
```

Listing 1: Código verbatim

```
\begin{verbatim}
Este texto mantiene los comandos \textbf{variados}
y se ignoran \LaTeX{}.
\end{verbatim}
```

Además, si queremos colocar una línea de código únicamente o destacar cualquier comando, utilizaremos el comando `\verb` y entre barras como esta `|` colocaremos nuestro comando. Para tenerlo más claro vamos a poner un ejemplo:
Este comando `D: Trabajo\Carpeta` indica la ubicación.

Listing 2: Código verbatim texto

```
Este comando \verb|D: Trabajo\Carpeta| indica la ubicación.
```

2. Usando listings para resaltar el código

Otro entorno que tenemos para presentar código es el lstings, es muy importante definir en el preámbulo el paquete listings y una vez hecho, se comenzará con el comando `\begin{lstlisting}[lenguaje, número de línea]`, entre corchetes colocaremos el lenguaje que estamos utilizando, y si nos dividen el código indicaremos la línea, y terminaremos con el comando `\end{lstlisting}`. Para tenerlo más claro vamos a poner un ejemplo:

```
\begin{frame}{Cryptoparty}
    \begin{alertblock}{Evento Internacional}
    Organizado por la UCM en España, este 6 de Abril en el Círculo
        de Bellas artes. \url{https://cryptoparty.ucm.es/}
    \end{alertblock}
    \begin{exampleblock}{ Qu é es?}
    Las CryptoParties son un evento gratuito y abierto para todo
       el mundo, especialmente para aquellos sin conocimientos
       previos que no hayan asistido previamente.
    \end{exampleblock}
    \begin{center}
     \includegraphics[width=0.2\textwidth]{Figures/Cryptoparty
        _2019.jpg}
```

```
    \end{center}
\end{frame}
```

Entre los lenguajes que podemos utilizar se encuentran los siguientes:

- C++.
- Cobol.
- Gnupot.
- HTML.
- Octave.
- Pascal.
- Python.
- Scilab.
- VHDL.
- XML.

3. Importando el código

Además de poder pegar nuestro código en el listing directamente, también podremos importar nuestro código con el comando `\lstinputlisting[lenguaje]{Programa}`. Para verlo más claro vamos a poner un ejemplo:

```
X=[365.015,404.64,407.55,435.91,546.01]; %valores de lamba
   experimental
Y=[365.015, 404.656, 407.783, 435.833]; %valores de lamba te rica
N=19; %n mero de puntos del ajuste


%Calculo de la penmdiente y la ordenada en el origen con su error
xm=mean(X);ym=mean(Y);
sumxx=sum((X-xm).^2);sumxy=sum((X-xm).*(Y-ym));sumyy=sum((Y-ym)
   .^2);
a=sumxy/sumxx
b=ym-a*xm
d=Y-a*X-b;
```

Listing 3: Código de importar el código

```
\lstinputlisting[language=Octave, firstline=2, lastline=12]{
   regresionTFG.m}
```

4. Dando estilo al código

Este entorno se puede modificar en el preámbulo los siguientes parámetros:

- **backgroundcolor**: Indica el color de fondo. Necesita el paquete color o xcolor.
- **commenstyle**: Estilo de los comentarios en el lenguaje.
- **basicstyle**: Fuente, tamaño de la letra en el código.
- **keywordstyle**: Estilo de las palabras clave.
- **numberstyle**: Estilo de las numeraciones.
- **numbersep**: Distancia entre los números del código.
- **stringstyle**: Estilo de las cadenas en el lenguaje.
- **showspaces**(true/false): Enfatiza en los espacios de las cadenas.
- **showstringspaces**(true/false): Enfatiza en los espacios en las cadenas.
- **showtabs**(true/false): Enfatiza en las tabulaciones en el código.
- **numbers**(left/right/none): Posición de los números.
- **prebreak**: Indica una marca al terminar una linea.
- **captionpos** (t/b): Posición del caption.
- **frame** (none/leftline/topline/bottomline/lines/single/shadowbox): Muesta el marco fuera del código.
- **breakwhitespace**: Muestra los espacios que ocurren cuando hay espacios en blanco.
- **breaklines**: Saltos de línea automáticos.
- **keepspaces**: Mantiene los espacios en el código, es útil para la indentación.
- **tabsize**: Tamaño de tabla por defecto.
- **escapeinside**: Especifica algunos caracteres en el código.
- **rulecolor**: Especifica el color en el marco de la caja.

Listing 4: Código ejemplo personalizado

```
\documentclass{article}
\usepackage[utf8]{inputenc}

\usepackage{listings}
\usepackage{color}
```

```
%Colores definidos
\definecolor{codegreen}{rgb}{0,0.6,0}
\definecolor{codegray}{rgb}{0.5,0.5,0.5}
\definecolor{codepurple}{rgb}{0.58,0,0.82}
\definecolor{backcolour}{rgb}{0.95,0.95,0.92}

%Dandole estilo al codigo
\lstdefinestyle{mystyle}{
  backgroundcolor=\color{backcolour},   commentstyle=\color{
     codegreen},
  keywordstyle=\color{magenta},
  numberstyle=\tiny\color{codegray},
  stringstyle=\color{codepurple},
  basicstyle=\footnotesize,
  breakatwhitespace=false,
  breaklines=true,
  captionpos=b,
  keepspaces=true,
  numbers=left,
  numbersep=5pt,
  showspaces=false,
  showstringspaces=false,
  showtabs=false,
  tabsize=2
}
%Set de mi c digo
\lstset{style=mystyle}
```

Además de lo explicado anteriormente, podemos darle estilo con los siguientes comandos:

- `\lstdefinestyle{mystyle}{}`: Definimos el código llamandolo mystyle y le damos las características que queramos.
- `\lstset{stylemystyle`: Estilo ya definido.

5. Listado y nombre del código

Como en las imágenes y en las tablas, podremos darle nombre y numerar nuestro código con el comando caption al lado del lenguaje. Para tenerlo más claro vamos a poner un ejemplo:

Listing 5: Ejemplo con nombre

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
```

```
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```

Listing 6: Ejemplo con nombre código

```
\begin{lstlisting}[language=Python, caption=Ejemplo con nombre]
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
```

```
VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```

Y si queremos que nos aparezca en una lista utilizaremos el comando `\lstlistoflistings`.

6. Poner palabras clave

Para poner palabras clave utilizaremos el comando `\lstdefinelanguage{lenguaje}{Características}`. Para tenerlo más claro vamos a poner un código de ejemplo:

Listing 7: Código palabras clave

```
\lstset{language=LaTeX,
    keywordstyle=\color{rojo},
    texcsstyle=*\color{myblue},
    basicstyle=\textbf\normalfont\ttfamily,
    commentstyle=\color{comments}\ttfamily,
    stringstyle=\rmfamily,
    numbers=left,
    numberstyle=\scriptsize,
    stepnumber=1,
    numbersep=8pt,
    captionpos=top,
    showstringspaces=false,
    breaklines=true,
    frameround=ftff,
    morekeywords={RequirePackage,ProvidesPackage,NeedsTeXFormat},
    backgroundcolor=\color{background},
    literate=
            *{\{}{{\textcolor{myblue}{\{}}}{1}
            {\}}{{\textcolor{myblue}{\}}}}{1}
            {\\}{{\textcolor{myblue}{\textbackslash}}}{1}
            {\$}{{\textcolor{rojo}{\$}}}{1}
            {\&}{{\textcolor{rojo}{\&}}}{1}
            {\documentclass}{{\textcolor{rojo}{\textbackslash
                \documentclass}}}{12}
            {\%\\\documentclass}{{\textcolor{rojo}{\textbackslash
                \documentclass}}}{12}
```

```
{\%\\usepackage}{{\textcolor{rojo}{\textbackslash usepackage}}}{9}
            {[}{{\textcolor{myblue}{[}}}{1}
            {]}{{\textcolor{myblue}{]}}}{1}
            {á}{{\'a}}1 {é}{{\'e}}1 {í}{{\'i}}1 {ó}{{\'o}}1 {ú}{{\
               'u}}1 {ñ}{{\~n}}1 {Á}{{\'A}}1 {É}{{\'E}}1 {Í}{{\'I}
               }1 {Ó}{{\'O}}1 {Ú}{{\'U}}1,
    frame=single,
    frameround={t}{t}{t}{t},
    framexleftmargin=6mm,
    numbers=left,
    numberstyle=\tiny\color{halfgray},
}
```

Tema 8: Código listing.
Marzo 2019
Ult. actualización 22 de marzo de 2019

LaTeX lic.LPPL & powered by OTEA – CC-ZERO