

# Generare codice $\text{\LaTeX}$ per via programmatica

Massimiliano Dominici

Gruppo utilizzatori Italiani di  $\text{\TeX}$

10 febbraio 2016

# Premessa

Questa presentazione è basata sull'intervento tenuto da Roberto Giacomelli e Gianluca Pignalberi al GuIT Meeting 2015, di cui segue l'impostazione generale.

Le slides dell'intervento originale sono disponibili qui:

<http://www.guitex.org/home/images/meeting2015/slides/gm2015-giacomelli-pignalberi.pdf>

# Motivazioni

Alcune tipologie di documento si prestano particolarmente bene ad essere costruite in maniera automatica.

Questo è particolarmente vero in quei casi in cui il documento si presenta come un **modulo** che deve essere riempito a partire da un insieme di **dati** i quali possono a loro volta richiedere un'elaborazione preliminare.

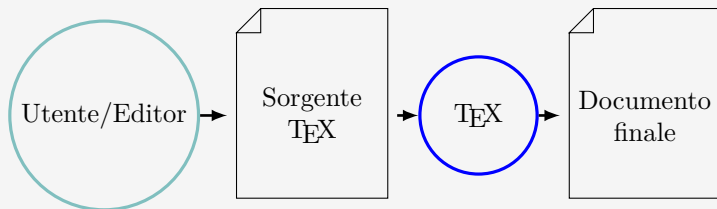
## Alcuni esempi

- il report complesso con dati residenti in un database;
- il mail-merge;
- la produzione di fatture a partire da un template;
- griglie di attività;
- computi metrici.

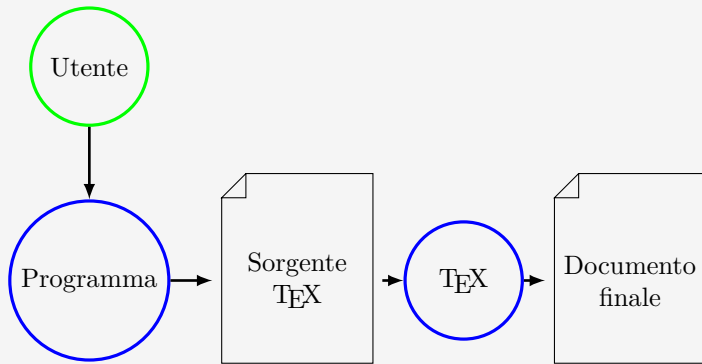
## $\TeX$ come possibile soluzione

In quanto file di testo, un documento  $\TeX$  si presta particolarmente bene ad essere generato automaticamente da un altro programma.  
Di conseguenza  $\TeX$  può essere usato in tutti casi elencati in precedenza.

# Normale flusso di lavoro con $\text{\TeX}$



# Flusso di lavoro con $\text{\TeX}$ in un processo automatizzato



# Vantaggi e svantaggi di $\text{\LaTeX}$

## Vantaggi:

- ottima qualità tipografica;
- modularità;
- un markup non particolarmente invasivo.

## Svantaggi:

- scarsamente adatto come linguaggio di programmazione e per l'elaborazione dei dati.



# Separazione

**Struttura:** template (Jinja2, lustache, Twig, ecc.)

**Presentazione:** classe o stile  $\LaTeX$

**Logica:** linguaggio di programmazione a uso generico  
(Lua, Python, ecc.)

**Dati:** formato di immagazzinamento di dati (database,  
XML, YAML, JSON, CSV, ecc.)

# Un esempio pratico

Produrre il PDF di una fattura con uno script Python, a partire da dati immagazzinati in un file YAML.

**Struttura:** Jinja2

**Presentazione:**  $\text{\LaTeX}$

**Logica:** Python

**Dati:** YAML

# Dati

```

---
number: 34843
date: '2001-01-03'
bill_to: &id001
  given: 'Chris'
  family: Dumars
  address:
    lines: |
      458 Walkman Dr.
      Suite 292
    city: 'Royal Oak'
    state: MI
    postal: 48046
ship_to: *id001
product:
  -
    sku: BL394D
    quantity: 4
    description: Basketball
    price: '450.00'
  -
    sku: BL4438
    quantity: 1
    description: 'Super Hoop'
    price: '2392.00'
tax: '251.42'
total: '4443.52'
comments: 'Late afternoon is best. Backup contact is Nancy Billsmer @ 338-4338'
...

```

# Frammento di template

```

\NumFattura{%{\num_fattura%}}
\Data{%{\data%}}

\intestazione

\Cliente{
  %{\cliente%}
}

\begin{center}
  \setlength\heavyrulewidth{0.1em}
  \begin{tabular}{llrr}
    \toprule
    \bfseries Id & \bfseries Descrizione & \bfseries Quantità & \bfseries Importo (€) \\
    \otoprule
    {%- for prodotto in prodotti %}
    %{\prodotto['sku']}% & %{\prodotto['description']}% & & \\
    %{\prodotto['quantity']}% & %{\prodotto['price']}% & & \\
    {%- endfor %}
    \midrule
    \bfseries Imponibile & & \bfseries {480,00} & \\
    Tasse & %{\tasse%} & & \\
    \bfseries Totale compensi & & \bfseries %{\totale%} & \\
    \bottomrule
  \end{tabular}
\end{center}

```

# Lo script: lettura dei dati e assegnazione delle variabili

```
import yaml, os

with open('fatture.yaml') as f:
    dati = yaml.load(f)

address = '\\\\'.join(['%s' % (v) for k,v in dati['bill_to']['address'].iteritems()])
name = ' '.join([dati['bill_to']['given'],dati['bill_to']['family']])
bill_to = '\\\\'.join([name, address])
prices = [product['price'] for product in dati['product']]
sub_total = sum(float(i) for i in prices)

params = {'num_fattura': dati['number'],'cliente': bill_to,'data': dati['date'],
          'prodotti': dati['product'],'tasse': dati['tax'], 'totale': dati['total']}
```

# Lo script: compilazione del template

```
from jinja2 import Environment, FileSystemLoader

latex_renderer = Environment(
    block_start_string = '%{',
    block_end_string = '%}',
    variable_start_string = '%{{',
    variable_end_string = '%}}',
    loader = FileSystemLoader(os.path.abspath('./'))
)

template = latex_renderer.get_template('fattura.tpl')
out = file('fattura.tex', 'w')
out.write(template.render(params).encode('utf-8'))
```

# Il risultato

```

\NumFattura{34843}
\Data{2001-01-03}

\intestazione

\Cliente{
  Chris Dumars\\Royal Oak\\MI\\48046\\458 Walkman Dr.
  Suite 292
}

\begin{center}
  \setlength\heavyrulewidth{0.1em}
  \begin{tabular}{llrr}
    \toprule
    \bfseries Id & \bfseries Descrizione & \bfseries Quantità & \bfseries Importo (€) \\
    \otoprule
    BL394D & Basketball & 4 & 450.00 \\
    BL4438 & Super Hoop & 1 & 2392.00 \\
    \midrule
    \bfseries Imponibile & & \bfseries {480,00} & \\
    Tasse & & 251.42 & \\
    \midrule
    \bfseries Totale compensi & & \bfseries 4443.52 & \\
    \bottomrule
  \end{tabular}
\end{center}

```

# Un report finanziario

È possibile produrre preventivi finanziari personalizzati, raccogliendo i dati dell'utente in un database tramite una web form, e popolando con essi un template che genera il file  $\TeX$  intermedio, prima della compilazione finale.

**Struttura:** Twig

**Presentazione:**  $\LaTeX$

**Logica:** PHP

**Dati:** Database SQL

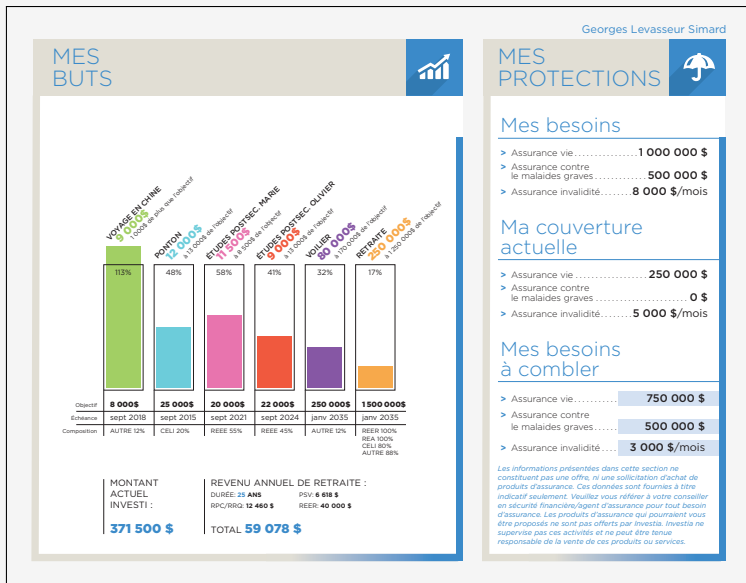


# Una porzione del template

```

\IRContainer{<%= recipient %>}{%
% ----- Parte sinistra
\IRBox{MES\ BUTS}{arrow}{472pt}{%
% ----- Primo elemento
\IRChart{%
  <% for but in buts %>
  \ChartBar{<%= but.target %>}{<%= but.amount %>}{<%= but.composition %>}{<%= but.formatted_date %>}
    {<%= but.formatted_amount %>}{<%= but.label %>}{<%= but.formatted_target %>}
    {<%= but.need %>}{<%= but.color %>}
  <% endfor %>
}{%
% ----- Secondo elemento
  MONTANT\par
  ACTUEL\par
  INVESTI:\par\vspace{16pt}
  \IRLegendTotal{<%= but.tot_inv %>}}%
}{%
% ----- Terzo elemento
  REVENU ANNUEL DE RETRAITE:\par
  \IRLegendItem{DURÉE}{<%= but.years %>}
  \IRLegendItem{PSV}{\textcolor{IRblue}{<% but.formatted_psv %> ANS}}\par
  \IRLegendItem{RPC/RRQ}{<% but.formatted_rpc_rrq %>}
  \IRLegendItem{REER}{<% but.formatted_reer %>}\par\vspace{16pt}
  TOTAL \IRLegendTotal{<%= but.tot_rev %>}}%
}%
}%
% ----- Parte Destra
[...]
```

# Risultato



# Definizione degli elementi ( $\text{\LaTeX}$ )

```

\RequirePackage{tikz}
\usetikzlibrary{fadings,calc,positioning}

[...]

\newcommand\IRContainer[3]{% #1 = Recipient's name; #2 = Left box; #3 = Right box
  \begin{tikzpicture}[remember picture, overlay, inner sep = 18pt, outer sep = 0pt]
    \node[anchor = south east,inner sep=0pt] at ([xshift=-18pt,yshift=-22pt]current page.north east)
      {\color{IRblue}\fontseries{12}\fontseries{12}\fontseries{m}\selectfont #1};
    \node[anchor = south west] at (current page.south west) {#2};
    \node[anchor = south east] at (current page.south east) {#3};
  \end{tikzpicture}
}

\newcommand\IRChart[3]{%
  \tcblower
  \setcounter{IRBar}{0}
  \begin{tikzpicture}[y=14.5pt,x=55pt]
    \draw ([xshift=-30pt,yshift=\IRvshift]0,1) -- ([xshift=0pt,yshift=\IRvshift]6,1);
    \draw ([xshift=-30pt,yshift=\IRvshift]0,2) -- ([xshift=0pt,yshift=\IRvshift]6,2);
    \node[irheader style] at (0,0) {Composition};
    \node[irheader style] at (0,1) {Échéance};
    \node[irheader style] at (0,2) {Objectif};
    #1
    \draw (0,-50pt) -- (0,-110pt);
    \node[irtable style,text width=100pt,] at ([xshift=\IRleftoffset]0,-50pt) {#2};
    \draw (2,-50pt) -- (2,-110pt);
    \node[irtable style,text width=200pt,] at ([xshift=\IRleftoffset]2,-50pt) {#3};
  \end{tikzpicture}
  \vspace{0pt}
}

```

# Definizione degli elementi ( $\text{\LaTeX}$ )

```

\newcommand\ChartBar[9]{%
  \FPdiv\IR@perc{#2}{#1}
  \FPMul\IR@perc{\IR@perc}{100}
  \FPround\IR@perc{\IR@perc}{0}
  \FPMul\IR@cib@height{\IR@perc}{1.37}
  \ifnum\IR@perc>100
    \def\IRnode{x1\theIRBar}
  \else
    \def\IRnode{x2\theIRBar}
  \fi
  \node[irlabel style] at ([xshift=\IRleftoffset]\c@IRBar,0) {\fontsize{6.7}{8}\selectfont#3};
  \node[irlabel style] at ([xshift=\IRleftoffset]\c@IRBar,1) {\fontsize{8.3}{8.3}\selectfont#4};
  \node[irlabel style] at ([xshift=\IRleftoffset]\c@IRBar,2) {\fontseries{k}\fontsize{8.3}{8.3}\selectfont#5};
  \node[colored irbar style,fill=#9,text height=\IR@cib@height] (x1\theIRBar) at ([xshift=2.5pt,yshift=2.5pt]
    \c@IRBar,3.25) {};
  \node[irbar style] (x2\theIRBar) at (\c@IRBar,3.25) {\fontsize{8.3}{8.3}\selectfont\IR@perc \%};
  \node[irpin style] at ([yshift=10pt]\IRnode.north west) {\fontseries{mb}\fontsize{8.3}{8.3}\selectfont#6};
  \node[irpin style] at ([yshift=10pt,xshift=-10pt]\IRnode.north)
    {\fontseries{k}\fontsize{12}{12}\selectfont\color{#9}#7};
  \node[irpin style] at ([yshift=10pt,xshift=-12pt]\IRnode.north east) {\fontsize{6.7}{6.7}\selectfont#8};
  \draw (x2\theIRBar.south west) -- ++(-90:50pt);
  \stepcounter{IRBar}
}

```

# Progetti reali

Ecco alcuni progetti concreti illustrati sulle pagine di *ArsTeXnica*:

- Applicazione “Facile” del Comune di Napoli – lettere interne, Agostino De Marco e Paolo Eugenio Cresci (<http://www.guitex.org/home/images/ArsTeXnica/AT012/FAciLE.pdf>);
- Menù per ristorante – web app in PHP, Claudio Fiandrino (<http://www.guitex.org/home/images/ArsTeXnica/AT018/menutex.pdf>);
- Rapporti di prova – documenti di valore tecnico/legale, Renato Battistin (<http://www.guitex.org/home/images/ArsTeXnica/AT018/rb-rapportiprova.pdf>).