

Typeset Code Listings and Emulate Console Screenshots with L^AT_EX Beautifully

<https://github.com/xziyue/latex-beautiful-listings-screenshot>

Ziyue “Alan” Xiang

May 10, 2020

Contents

1	Quick Start Guide	1
2	Typeset Source Code Listings	1
3	Typeset Generic Verbatims	2
4	Typeset Console Screenshots	3
4.1	Unicode Support	5
5	Add Captions	5

1 Quick Start Guide

1. Download `customlisting.sty` and place it in your project folder.
2. Load the package with `\usepackage{customlisting}`.

This package provides the following environments:

- `tcbconsole`, `tcbconsole*`
- `tcbcode`, `tcbcode*`
- `tcbverbatim`, `tcbverbatim*`

This package also provides the following commands:

- `tcbinputcode`, `tcbinputcode*`
- `tcbinputverbatim`, `tcbinputverbatim*`

The starred environments/commands offer *unbreakable* listing boxes; while normal ones are *breakable*.

2 Typeset Source Code Listings

- Typeset source code inside \TeX files

```
1 \begin{tcbcode}{cpp}
2 #include <iostream>
3 using namespace std;
4
5 int main(){
6     cout<<"Hello World\n";
7     return 0;
8 }
9 \end{tcbcode}
```

Code

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout<<"Hello World\n";
6     return 0;
7 }
```

- Typeset source code from external source files

```
1 \tcbinputcode*{cpp}{example.cpp}
```

```
Code
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      cout<<"Hello World\n";
6      return 0;
7  }
```

- Inline source code

```
1 \cinline|printf("%s", "some text");|
2 \pyinline|map(lambda x:x, [1, 2])|
3 \rawinline|raw value|
```

```
printf("%s", "some text"); map(lambda x:x, [1, 2]) raw value
```

- Declare inline macros for other languages

```
1 \newmintinline[rubyinline]{ruby}{frame=none, fontsize=\fontsize{10}{10}}
2 \rubyinline|puts 'Hello, world!'|
```

```
puts 'Hello, world!'
```

3 Typeset Generic Verbatims

- Typeset generic verbatims inside T_EX files

```

1 \begin{tcbverbatim}
2
3 \begin{tikzpicture}
4 \draw (0,0) -- (1,0) -- (1,1) -- (0,1) -- (0,0);
5 \draw (1,0) -- (2,0) -- (2,1) -- (1,1) -- (1,0);
6 \draw (2,0) -- (3,0) -- (3,1) -- (2,1) -- (2,0);
7 \draw (3,0) -- (4,0) -- (4,1) -- (3,1) -- (3,0);
8 \draw (4,0) -- (5,0) -- (5,1) -- (4,1) -- (4,0);
9 \draw (5,0) -- (6,0) -- (6,1) -- (5,1) -- (5,0);
10 \draw (6,0) -- (7,0) -- (7,1) -- (6,1) -- (6,0);
11 \draw (7,0) -- (8,0) -- (8,1) -- (7,1) -- (7,0);
12 \draw (8,0) -- (9,0) -- (9,1) -- (8,1) -- (8,0);
13 \draw (9,0) -- (10,0) -- (10,1) -- (9,1) -- (9,0);
14 \draw (10,0) -- (11,0) -- (11,1) -- (10,1) -- (10,0);
15 \draw (11,0) -- (12,0) -- (12,1) -- (11,1) -- (11,0);
16 \draw (12,0) -- (13,0) -- (13,1) -- (12,1) -- (12,0);
17 \draw (13,0) -- (14,0) -- (14,1) -- (13,1) -- (13,0);
18 \draw (14,0) -- (15,0) -- (15,1) -- (14,1) -- (14,0);
19 \draw (15,0) -- (16,0) -- (16,1) -- (15,1) -- (15,0);
20 \draw (16,0) -- (17,0) -- (17,1) -- (16,1) -- (16,0);
21 \draw (17,0) -- (18,0) -- (18,1) -- (17,1) -- (17,0);
22 \draw (18,0) -- (19,0) -- (19,1) -- (18,1) -- (18,0);
23 \draw (19,0) -- (20,0) -- (20,1) -- (19,1) -- (19,0);
24 \draw (20,0) -- (21,0) -- (21,1) -- (20,1) -- (20,0);
25 \draw (21,0) -- (22,0) -- (22,1) -- (21,1) -- (21,0);
26 \draw (22,0) -- (23,0) -- (23,1) -- (22,1) -- (22,0);
27 \draw (23,0) -- (24,0) -- (24,1) -- (23,1) -- (23,0);
28 \draw (24,0) -- (25,0) -- (25,1) -- (24,1) -- (24,0);
29 \draw (25,0) -- (26,0) -- (26,1) -- (25,1) -- (25,0);
30 \draw (26,0) -- (27,0) -- (27,1) -- (26,1) -- (26,0);
31 \draw (27,0) -- (28,0) -- (28,1) -- (27,1) -- (27,0);
32 \draw (28,0) -- (29,0) -- (29,1) -- (28,1) -- (28,0);
33 \draw (29,0) -- (30,0) -- (30,1) -- (29,1) -- (29,0);
34 \draw (30,0) -- (31,0) -- (31,1) -- (30,1) -- (30,0);
35 \draw (31,0) -- (32,0) -- (32,1) -- (31,1) -- (31,0);
36 \draw (32,0) -- (33,0) -- (33,1) -- (32,1) -- (32,0);
37 \draw (33,0) -- (34,0) -- (34,1) -- (33,1) -- (33,0);
38 \draw (34,0) -- (35,0) -- (35,1) -- (34,1) -- (34,0);
39 \draw (35,0) -- (36,0) -- (36,1) -- (35,1) -- (35,0);
40 \draw (36,0) -- (37,0) -- (37,1) -- (36,1) -- (36,0);
41 \draw (37,0) -- (38,0) -- (38,1) -- (37,1) -- (37,0);
42 \draw (38,0) -- (39,0) -- (39,1) -- (38,1) -- (38,0);
43 \draw (39,0) -- (40,0) -- (40,1) -- (39,1) -- (39,0);
44 \draw (40,0) -- (41,0) -- (41,1) -- (40,1) -- (40,0);
45 \draw (41,0) -- (42,0) -- (42,1) -- (41,1) -- (41,0);
46 \draw (42,0) -- (43,0) -- (43,1) -- (42,1) -- (42,0);
47 \draw (43,0) -- (44,0) -- (44,1) -- (43,1) -- (43,0);
48 \draw (44,0) -- (45,0) -- (45,1) -- (44,1) -- (44,0);
49 \draw (45,0) -- (46,0) -- (46,1) -- (45,1) -- (45,0);
50 \draw (46,0) -- (47,0) -- (47,1) -- (46,1) -- (46,0);
51 \draw (47,0) -- (48,0) -- (48,1) -- (47,1) -- (47,0);
52 \draw (48,0) -- (49,0) -- (49,1) -- (48,1) -- (48,0);
53 \draw (49,0) -- (50,0) -- (50,1) -- (49,1) -- (49,0);
54 \draw (50,0) -- (51,0) -- (51,1) -- (50,1) -- (50,0);
55 \draw (51,0) -- (52,0) -- (52,1) -- (51,1) -- (51,0);
56 \draw (52,0) -- (53,0) -- (53,1) -- (52,1) -- (52,0);
57 \draw (53,0) -- (54,0) -- (54,1) -- (53,1) -- (53,0);
58 \draw (54,0) -- (55,0) -- (55,1) -- (54,1) -- (54,0);
59 \draw (55,0) -- (56,0) -- (56,1) -- (55,1) -- (55,0);
60 \draw (56,0) -- (57,0) -- (57,1) -- (56,1) -- (56,0);
61 \draw (57,0) -- (58,0) -- (58,1) -- (57,1) -- (57,0);
62 \draw (58,0) -- (59,0) -- (59,1) -- (58,1) -- (58,0);
63 \draw (59,0) -- (60,0) -- (60,1) -- (59,1) -- (59,0);
64 \draw (60,0) -- (61,0) -- (61,1) -- (60,1) -- (60,0);
65 \draw (61,0) -- (62,0) -- (62,1) -- (61,1) -- (61,0);
66 \draw (62,0) -- (63,0) -- (63,1) -- (62,1) -- (62,0);
67 \draw (63,0) -- (64,0) -- (64,1) -- (63,1) -- (63,0);
68 \draw (64,0) -- (65,0) -- (65,1) -- (64,1) -- (64,0);
69 \draw (65,0) -- (66,0) -- (66,1) -- (65,1) -- (65,0);
70 \draw (66,0) -- (67,0) -- (67,1) -- (66,1) -- (66,0);
71 \draw (67,0) -- (68,0) -- (68,1) -- (67,1) -- (67,0);
72 \draw (68,0) -- (69,0) -- (69,1) -- (68,1) -- (68,0);
73 \draw (69,0) -- (70,0) -- (70,1) -- (69,1) -- (69,0);
74 \draw (70,0) -- (71,0) -- (71,1) -- (70,1) -- (70,0);
75 \draw (71,0) -- (72,0) -- (72,1) -- (71,1) -- (71,0);
76 \draw (72,0) -- (73,0) -- (73,1) -- (72,1) -- (72,0);
77 \draw (73,0) -- (74,0) -- (74,1) -- (73,1) -- (73,0);
78 \draw (74,0) -- (75,0) -- (75,1) -- (74,1) -- (74,0);
79 \draw (75,0) -- (76,0) -- (76,1) -- (75,1) -- (75,0);
80 \draw (76,0) -- (77,0) -- (77,1) -- (76,1) -- (76,0);
81 \draw (77,0) -- (78,0) -- (78,1) -- (77,1) -- (77,0);
82 \draw (78,0) -- (79,0) -- (79,1) -- (78,1) -- (78,0);
83 \draw (79,0) -- (80,0) -- (80,1) -- (79,1) -- (79,0);
84 \draw (80,0) -- (81,0) -- (81,1) -- (80,1) -- (80,0);
85 \draw (81,0) -- (82,0) -- (82,1) -- (81,1) -- (81,0);
86 \draw (82,0) -- (83,0) -- (83,1) -- (82,1) -- (82,0);
87 \draw (83,0) -- (84,0) -- (84,1) -- (83,1) -- (83,0);
88 \draw (84,0) -- (85,0) -- (85,1) -- (84,1) -- (84,0);
89 \draw (85,0) -- (86,0) -- (86,1) -- (85,1) -- (85,0);
90 \draw (86,0) -- (87,0) -- (87,1) -- (86,1) -- (86,0);
91 \draw (87,0) -- (88,0) -- (88,1) -- (87,1) -- (87,0);
92 \draw (88,0) -- (89,0) -- (89,1) -- (88,1) -- (88,0);
93 \draw (89,0) -- (90,0) -- (90,1) -- (89,1) -- (89,0);
94 \draw (90,0) -- (91,0) -- (91,1) -- (90,1) -- (90,0);
95 \draw (91,0) -- (92,0) -- (92,1) -- (91,1) -- (91,0);
96 \draw (92,0) -- (93,0) -- (93,1) -- (92,1) -- (92,0);
97 \draw (93,0) -- (94,0) -- (94,1) -- (93,1) -- (93,0);
98 \draw (94,0) -- (95,0) -- (95,1) -- (94,1) -- (94,0);
99 \draw (95,0) -- (96,0) -- (96,1) -- (95,1) -- (95,0);
100 \draw (96,0) -- (97,0) -- (97,1) -- (96,1) -- (96,0);
101 \draw (97,0) -- (98,0) -- (98,1) -- (97,1) -- (97,0);
102 \draw (98,0) -- (99,0) -- (99,1) -- (98,1) -- (98,0);
103 \draw (99,0) -- (100,0) -- (100,1) -- (99,1) -- (99,0);
104 \draw (100,0) -- (101,0) -- (101,1) -- (100,1) -- (100,0);
105 \draw (101,0) -- (102,0) -- (102,1) -- (101,1) -- (101,0);
106 \draw (102,0) -- (103,0) -- (103,1) -- (102,1) -- (102,0);
107 \draw (103,0) -- (104,0) -- (104,1) -- (103,1) -- (103,0);
108 \draw (104,0) -- (105,0) -- (105,1) -- (104,1) -- (104
```

Verbatim

```
-----  
 \_   ^_   ^_   ^_   V /  
 |   |   |   |   |   |  
 |___|   |___|   |___|  
          /_____\ /_____\ /_____  
            \      /      /      \  
             V      V      V      \  
              \     /     /     /
```

- Typeset generic verbatims from external files

```
1 \tcbininputverbatim*{wireshark.txt}
```

No.	Time	Source	Destination	Protocol	Length	Info
118	10.159070602	2604:6000:1419:404a::6	2607:f800:4009:805::2003	TCP	86	4236>> 443 [ACK] Seq=936 ack=52751 Win=2933 Len=0 TSval=2838670553 TSrc=302725484
119	0.17775109	192.168.0.1	192.168.0.100	DNS	140	Standard query response 0x8290 A flags:attnic.com CHAME getaticsdns1.l.google.com A 172.17.1.35 OPT
120	0.18038905	192.168.0.1	192.168.0.100	DNS	152	Standard query response 0x0ec8 AAAA flags:attnic.com CHAME getaticsdns1.l.google.com AAAA 2607:fb0
0000:802::2003 OPT						
121	0.174444739	2604:6000:1419:404a::6	2607:f800:4009:802::2003	TCP	94	3696>> 443 [STW] Seq=0 Win=64800 Len=0 MSS=1440 SCAP_PERN=1 TSval=401672760 TSrc=0 WS=128
122	0.180381333	2607:f800:4009:805::2003	2604:6000:1419:404a::6	TCP	86	4K5> 42188 [ACK] Seq=1883 Ack=531 Win=6806 Len=0 TSval=319629076 TSrc=2386670536
123	1.85264251	2607:f800:4009:806::2003	2604:6000:1419:404a::6	TCP	86	4K5> 51584 [ACK] Seq=1 Ack=591 Win=6816 Len=0 TSval=1161394810 TSrc=356743249
124	0.174559861	2607:f800:4009:806::2003	2604:6000:1419:404a::6	TLSv1.3	298	Server Hello, Change Cipher Spec, Application Data
125	0.187406401	2604:6000:1419:404a::6	2607:f800:4009:806::2003	TCP	86	63564> 443 [ACK] Seq=591 Ack=213 Win=64768 Len=0 TSval=356743285 TSrc=1161394811
126	0.187881868	2604:6000:1419:404a::6	2607:f800:4009:806::2003	TLSv1.3	150	Change Cipher Spec, Application Data
127	0.18975438	2604:6000:1419:404a::6	2607:f800:4009:806::2003	TLSv1.3	172	Application Data
128	0.19007758	2604:6000:1419:404a::6	2607:f800:4009:806::2003	TLSv1.3	323	Application Data
129	0.189267503	2604:6000:1419:404a::6	2607:f800:4009:806::2003	TLSv1.3	648	Application Data
130	0.19477468	2607:f800:4009:805::2004	2604:6000:1419:404a::6	TCP	86	4K5> 42136 [ACK] Seq=52751 Ack=936 Win=609 Len=0 TSval=302725520 TSrc=2838670548
131	0.21161087	2607:f800:4009:802::2003	2604:6000:1419:404a::6	TCP	94	4K5> 39596 [STW, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1360 SCAP_PERN=1 TSval=3015134314 TSrc=0
WS=256						
132	0.211641489	2604:6000:1419:404a::6	2607:f800:4009:802::2003	TCP	86	3696>> 443 [ACK] Seq=1 Win=64896 Len=0 TSval=4016727640 TSrc=3015134314

Typesetting console screenshots is a bit trickier. By far, it can be done most conveniently on Ubuntu 18.04+. The key is to convert ANSI color codes used by the console into HTML. As it is shown in Figure 1, on Ubuntu 18.04+, this can be done simply by selecting the desired region, right click and select “Copy as HTML”. On other platforms, this should be also doable by dumping the terminal output to a file and using a conversion tool such as `ansi2html`.

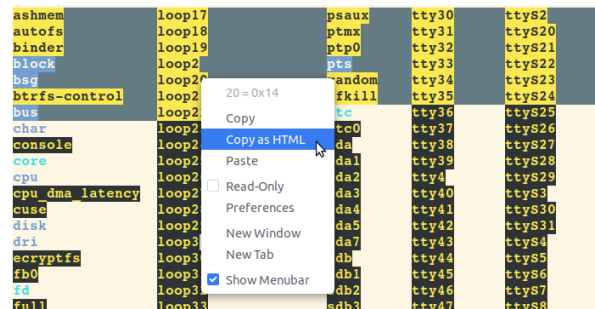


Figure 1: Converting terminal output to HTML on Ubuntu 18.04+.

Generally speaking, one needs to fulfill the following requirements:

1. Have a way of converting terminal output to HTML.
2. Be able to run the `html2latex` L^AT_EX Python script. Currently, the script is dependent on wxPython, TexSoup and PyLaTeX. Notice that this software is very primitive and does not support many HTML features.

To typeset this screenshot in L^AT_EX, one needs to run `html2latex` and paste the HTML in the upper text box. By pressing the “Convert” button, the corresponding L^AT_EX code will appear in the lower text box, as it is shown in Figure 2. The result is shown as below.

```
1 \input{console-dev.txt}
```

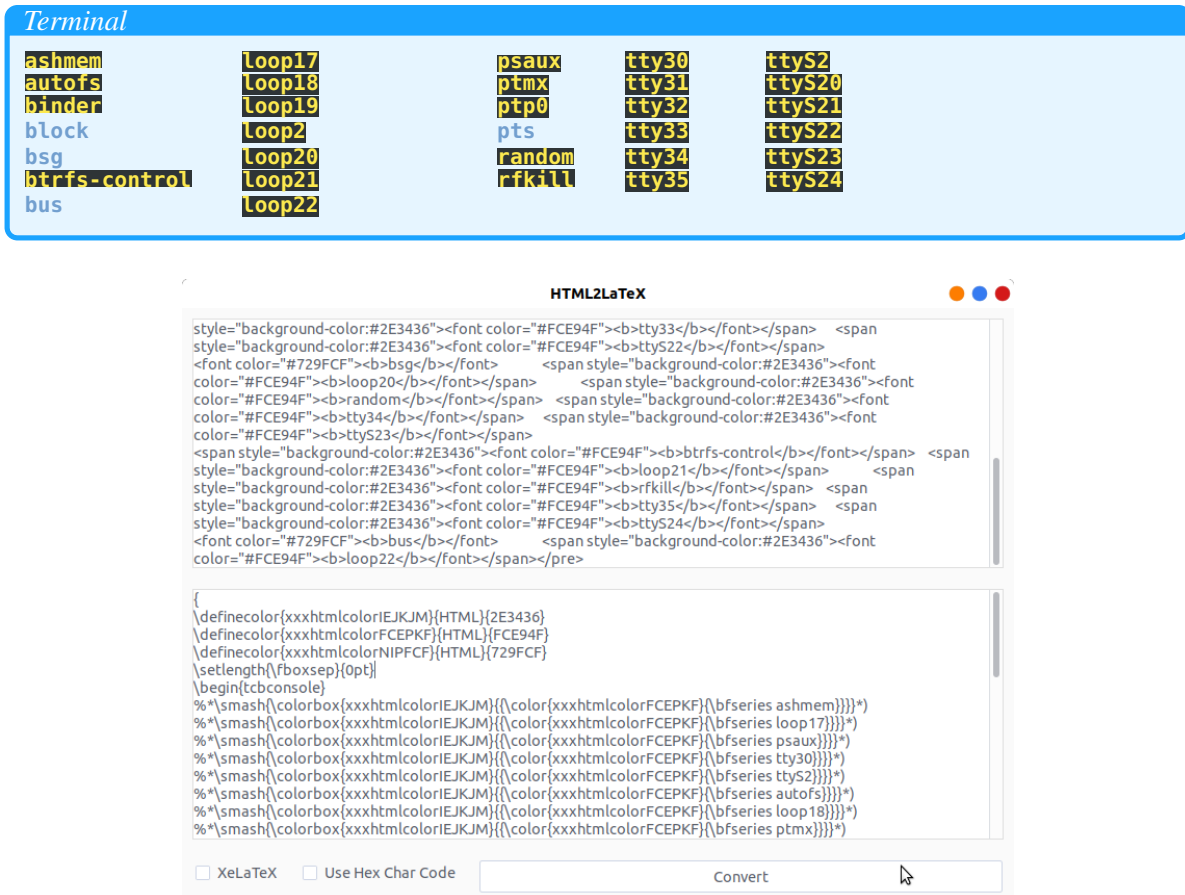
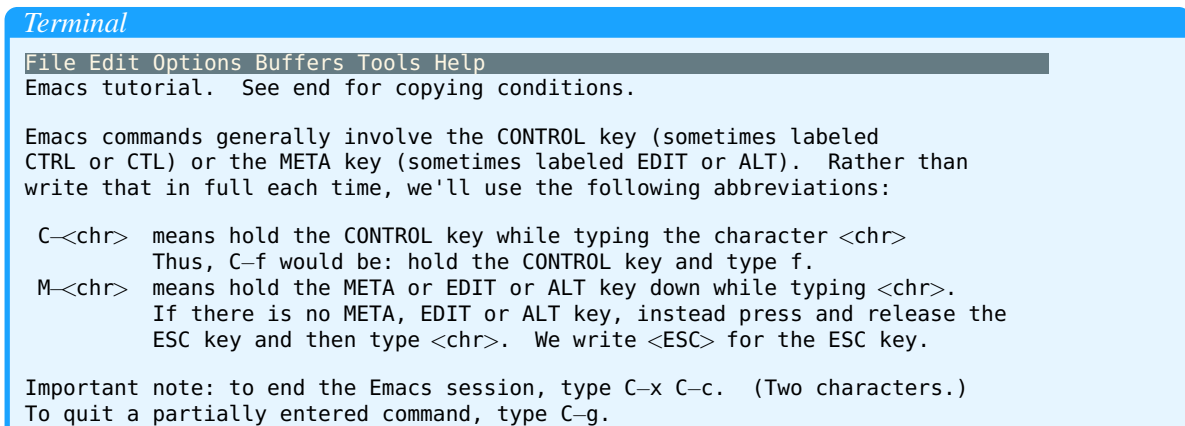


Figure 2: Using html2latex to convert HTML to \LaTeX .

Other classic command-line tools, such as emacs, are supported as well.

```
1 \input{console-emacs.txt}
```



The characters ">>" at the left margin indicate directions for you to try using a command. For instance:

```
>> Now type C-v (View next screen) to move to the next screen.  
    (go ahead, do it by holding down the CONTROL key while typing v).  
    From now on, you should do this again whenever you finish  
    reading the screen.
```

```
-UU-:----F1  TUTORIAL  Top L1  (Fundamental) -----
```

4.1 Unicode Support

Very frequently, the terminal output contains Unicode characters. For \TeX distribution that supports Unicode input natively (e.g. \XeLaTeX , \LuaLaTeX), this should not be a problem. Just remember to tick the “XeLaTeX” check box in `html2latex`.

As for the most commonly used \pdfLaTeX , special treatment is needed. The solution is to use the `\unichar` command provided by loading `\usepackage[utf8x]{inputenc}`. Therefore, if you are using \pdfLaTeX and there is Unicode character inside the terminal output, you should do the following:

1. Make sure to include `\usepackage[utf8x]{inputenc}` in your preamble.
2. Put `customlisting-unicode.sty` into your project folder and load it with `\usepackage{customlisting-unicode}`.
3. In `html2latex`, make sure “XeLaTeX” is unchecked.

A \pdfLaTeX example is shown as below. However, keep in mind that this Unicode support is extremely limited: many characters are simply unavailable in \pdfLaTeX .

```
1 \input{console-unicode.txt}
```

Terminal

```
(base) user@machine:~/latex.typeset.listings/doc$ cat unicode-test.txt  
Basic Latin  
! " # $ % & ' ( ) * +  
Latin-1 Supplement  
  i ç £ ¤ ¥ ¦ § ¨ © º  
Latin Extended-A  
Ā ā Ă ă Ć ć Ĉ ĉ Ċ ċ
```

5 Add Captions