1 Basics

[1.1]

```
\ExplSyntaxOn  % <------
Some code.
Some code.
Some code.
\ExplSyntaxOff  % <------
```

[1.2]

```
Some code.
\group _begin: % <------
Some code.
\group _end: % <------
Some code.
```

[1.3] – Grouping

```
\ExplSyntaxOn
Text ~ text ~
\group_begin:
\color{red}
\large
text ~ text ~
\group_end:
text ~ text.
\ExplSyntaxOff
```

Text text text text text text.

[1.4] – Defining a function

```
\ExplSyntaxOn
\cs_new:Npn \ic_funca:n #1 { {\color{blue} \bfseries #1 } }

Text ~ \ic_funca:n { ~ This ~ is ~ blue ~ bold. } ~ Text.
\ExplSyntaxOff
```

Text **This is blue bold.** Text.

[1.5] – Defining a user command

```
\ExplSyntaxOn
\NewDocumentCommand { \mycmdb } { m } {
\cs_new:Npn \ic_funcb:n ##1 { {\color{blue} \bfseries ##1 } }
 \ic_funcb:n { #1 }
 }
\ExplSyntaxOff
Text \mycmdb{This is blue bold.} Text.
```

Text **This is blue bold.** Text.

[1.6] – Using a token list

```
\ExplSyntaxOn
\tl_new:N \l_myicc_tl
\NewDocumentCommand { \mycmdc } { m } {
\tl_set:Nn \l_myicc_tl { #1 }
\cs_new:Npn \ic_funcc:N ##1 { {\color{blue} \bfseries ##1 } }
 \ic_funcc:N \l_myicc_tl
 }
\ExplSyntaxOff
Text \mycmdc{This is blue bold.} Text.
```

Text **This is blue bold.** Text.

[1.7] – Appending to a token list

```
\ExplSyntaxOn
\tl_new:N \l_myicd_tl
\NewDocumentCommand { \mycmdd } { m } {
\tl_set:Nn \l_myicd_tl { #1 }
\tl_put_right:Nn \l_myicd_tl { $\leftarrow$ ! ! }
\tl_put_left:Nn \l_myicd_tl { ! ! $\rightarrow$ }
\cs_new:Npn \ic_funcd:N ##1 { {\color{blue} \bfseries ##1 } }
 \ic_funcd:N \l_myicd_tl
 }
\ExplSyntaxOff
Text \mycmdd{This is blue bold.} Text.
```

Text **!!$\rightarrow$This is blue bold.$\leftarrow$!!** Text.

[1.8] – Modifying a token list (\tl_replace)

```
\ExplSyntaxOn
\tl_new:N \l_myice_tl
\NewDocumentCommand { \mycmde } { m } {
\tl_set:Nn \l_myice_tl { #1 }
\tl_replace_all:Nnn \l_myice_tl { blue } { not ~ red }
\tl_replace_all:Nnn \l_myice_tl { bold } { italic }
\cs_new:Npn \ic_funce:N ##1 { {\color{blue} \bfseries ##1 } }
 \ic_funce:N \l_myice_tl
 }
\ExplSyntaxOff
Text \mycmde{This is blue bold.} Text.
```

Text **This is not red italic.** Text.

[1.9] – Modifying a token list (\regex_replace)

```
\ExplSyntaxOn
\tl_new:N \l_myicez_tl
\NewDocumentCommand { \mycmdez } { m } {
\tl_set:Nn \l_myicez_tl { #1 }
\regex_replace_all:nnN
{ (\c{color} \cB\{ [^\cE\}]* \cE\}) }
{ not \  }
\l_myicez_tl
\regex_replace_all:nnN
{ \c{textbf}(.+)(bold) }
{ is\ \c{textsc}\1 small\ caps }
\l_myicez_tl
\tl_use:N \l_myicez_tl
 }
\ExplSyntaxOff
Text \mycmdez{This is {\color{blue}blue} and \textbf{bold}.} Text.
```

Text This is not blue and is SMALL CAPS. Text.

[1.10] – Copying a control sequence

```
\ExplSyntaxOn
\cs_new:Npn \ic_funcf:N #1 { {\color{blue} \bfseries #1 } }
\cs_new_eq:NN \mybbcmd \ic_funcf:N
\ExplSyntaxOff
Text \mybbcmd{This is blue bold.} Text.
```

Text **This is blue bold.** Text.

[1.11] – Looping

```
1 \ExplSyntaxOn
2 \cs_set:Npn \ic_funcg:n #1 { \symbol{#1} }
3 \cs_set:Nn \ic_funcgb:  {
4 \int_step_function:nnnN { 97 } { 1 } { 122 } \ic_funcg:n
5 \
6 }
7 \cs_new_eq:NN \myloopcmd \ic_funcgb:
8 \ExplSyntaxOff
9 Text \myloopcmd Text.
```

Text abcdefghijklmnopqrstuvwxyz Text.

Commentary:
(1) Open the expl3 environment.
(2) Define a 1-parameter function, g, that will print a glyph, given the glyph's slot number, in the current font using the symbol command.
(3) Define a no-parameter function, gb, that will
(4) step through values 97 to 122 (inclusive) and pass each value to the g function.
(5) Add a space (replacing the one gobbled after the command in the user code).
(6) –
(7) Create a user-command, \myloopcmd, that will call the gb function.
(8) Close the expl3 environment.
(9) Use the user-command

Result: The letters a..z are printed, followed by a space.

[1.12] – Mapping function

```
\ExplSyntaxOn
\cs_set:Npn \ic_funch:n #1 { \fbox{#1}. }
\tl_new:N \l_myich_tl
\NewDocumentCommand { \mycmdh } { m } {
\tl_set:Nn \l_myich_tl { #1 }
\tl_map_function:NN \l_myich_tl \ic_funch:n
\par tl ~ = ~ >> \tl_use:N \l_myich_tl <<
}
\ExplSyntaxOff
Text \mycmdh{abc{de}fgh} Text.
```

Text $\boxed{\text{a}}$.$\boxed{\text{b}}$.$\boxed{\text{c}}$.$\boxed{\text{de}}$.$\boxed{\text{f}}$.$\boxed{\text{g}}$.$\boxed{\text{h}}$.
tl = »abcdefgh« Text.

[1.13] – Mapping inline function

```
\ExplSyntaxOn
\cs_set:Npn \ic_funci:n #1 { \fbox{#1}. }
\tl_new:N \l_myici_tl
\NewDocumentCommand { \mycmdi } { m } {
\tl_set:Nn \l_myici_tl { #1 }
\tl_map_inline:Nn \l_myici_tl { \ic_funci:n {##1} }
\par tl ~ = ~ >> \tl_use:N \l_myici_tl <<
}
\ExplSyntaxOff
Text \mycmdi{abc{de}fgh} Text.
```

Text $\boxed{\text{a}}$.$\boxed{\text{b}}$.$\boxed{\text{c}}$.$\boxed{\text{de}}$.$\boxed{\text{f}}$.$\boxed{\text{g}}$.$\boxed{\text{h}}$.
tl = »abcdefgh« Text.

[1.14] – Contents of a token list (1)

```
\ExplSyntaxOn
\tl_new:N \l_myicj_tl
\NewDocumentCommand { \mycmdj } { m } {
\tl_set:Nn \l_myicj_tl { #1 }
\par (a) ~ \tl_count:N \l_myicj_tl  \ token ~ groups.
\par (b) ~ \tl_count_tokens:n { \l_myicj_tl } ~ token
    \int_compare:nNnTF {\tl_count_tokens:n { \l_myicj_tl }} = { 1
    } { } { s }.
\par (c) ~ \exp_args:No \tl_count_tokens:n { \l_myicj_tl } ~
    tokens: ~
{ \color{blue} \tl_to_str:N \l_myicj_tl }.
}
\ExplSyntaxOff
\mycmdj{abc{de}fgh}
```

(a) 7 token groups.
(b) 1 token.
(c) 10 tokens: abc{de}fgh.

[1.15] – Contents of a token list (2)

```
\ExplSyntaxOn
\tl_new:N \l_myick_tl
\NewDocumentCommand { \mycmdk } { m } {
\tl_set:Nn \l_myick_tl { #1 }
\par head: ~ \tl_head:N \l_myick_tl
\par tail: ~ \tl_tail:N \l_myick_tl
\par reverse: ~ \tl_reverse:N \l_myick_tl   \tl_use:N \l_myick_tl
    ~<~ \tl_to_str:N \l_myick_tl
\par 5th ~ item : ~ \tl_item:Nn \l_myick_tl { 5 }
\par reverse: ~ \tl_reverse:N \l_myick_tl   \tl_use:N \l_myick_tl
    ~<~ \tl_to_str:N \l_myick_tl
\par reverse ~ items: ~ \exp_args:No \tl_reverse_items:n {
    \l_myick_tl } ~<~ \tl_to_str:N \l_myick_tl
\par 5th ~ item : ~ \tl_item:Nn \l_myick_tl { 5 }
}
\ExplSyntaxOff
\mycmdk{abc{de}fgh}
```

```
head: a
tail: bcdefgh
reverse: hgfdecba< hgf{de}cba
5th item: c
reverse: abcdefgh< abc{de}fgh
reverse items: hgfdecba < abc{de}fgh
5th item: f
```

[1.16] – Tokens

```
\ExplSyntaxOn
\tl_new:N \l_myiclz_tl
\cs_set:Npn \ic_funcl:n #1 {
\tl_set:Nn \l_myiclz_tl { #1 }
\fbox{ \strut \tl_to_str:N \l_myiclz_tl } $^ \exp_args:No
    \tl_count_tokens:n { \l_myiclz_tl }$
\int_compare:nNnT
{ \exp_args:No \tl_count_tokens:n { \l_myiclz_tl } }
>
{ 1 }
 { >> \mycmdl{#1} << }
}
\tl_new:N \l_myicl_tl
\NewDocumentCommand { \mycmdl } { m } {
\tl_set:Nn \l_myicl_tl { #1 }
\tl_map_function:NN \l_myicl_tl \ic_funcl:n
}
\ExplSyntaxOff
 \mycmdl{abc{d\textit{e}}fgh}
```

$\boxed{\text{a}}^{1}\boxed{\text{b}}^{1}\boxed{\text{c}}^{1}\boxed{\text{d\textbackslash textit \{e\}}}^{5}$ » $\boxed{\text{d}}^{1}\boxed{\text{\textbackslash textit }}^{1}\boxed{\text{e}}^{1}$ « $\boxed{\text{f}}^{1}\boxed{\text{g}}^{1}\boxed{\text{h}}^{1}$

*=====

2 regexpatch

[2.1] – testca

```
\newcommand{\testca}{\textit{label}}
Before: \testca
\par \regexpatchcmd{\testca}{\c{textit}}{\c{textbf}}{S}{F}
\par \xpatchcmd{\testca}{label}{babble}{S}{F}
\par After: \testca
```

Before: *label*
S
S
After: **babble**

[2.2] – ph: Too many brace levels

```
\newcommand{\ph}[1]{
\textbf{\textsc{{\color{blue}#1}}}\ \ }
Before: {\testfont\ph{Snail in the Bottle}}
\regexpatchcmd{\ph}{\c{color}\cB\{blue\cE\}}{red}{}{F}
\par After: {\testfont\ph{Snail in the Bottle}}
```

Before: **SNAIL IN THE BOTTLE** F
After: **SNAIL IN THE BOTTLE**

[2.3] – ph2: Two levels of braces

```
\newcommand{\phb}[1]{\textsc{{\color{blue}#1}}\ \ }
Before: {\testfont\phb{Snail in the Bottle}}
\regexpatchcmd{\phb}{\c{color}\cB\{blue\cE\}}{red}{}{F}
\par After: {\testfont\phb{Snail in the Bottle}}
```

Before: SNAIL IN THE BOTTLE
After: REDSNAIL IN THE BOTTLE

[2.4] – ph3: Entire \color command replaced

```
\newcommand{\phc}[1]{{
\bfseries\scshape\color{blue}#1\ \ }}
Before: {\testfont\phc{Snail in the Bottle}}
\regexpatchcmd{\phc}{\c{color}\cB\{blue\cE\}}
    {\c{color}\cB\{red\cE\}}{}{F}
\par After: {\testfont\phc{Snail in the Bottle}}
```

Before: **SNAIL IN THE BOTTLE**
After: **SNAIL IN THE BOTTLE**

[2.5] – ph4: Text replaced: ‘blue’ > ‘red’

```
\newcommand{\phd}[1]{{
\bfseries\scshape\color{blue}#1\ \ }}
Before: {\testfont\phd{Snail in the Bottle}}
\xpatchcmd{\phd}{blue}{red}{}{F}
\par After: {\testfont\phd{Snail in the Bottle}}
```

Before: **SNAIL IN THE BOTTLE**
After: **SNAIL IN THE BOTTLE**

[2.6] – ph5: Text (‘blue’) replaced by a macro (‘\mycolour’)

```
\newcommand{\mycolour}{green}
\newcommand{\phe}[1]{{
\bfseries\scshape\color{blue}#1\ \ }}
Before: {\testfont\phe{Snail in the Bottle}}
\regexpatchcmd{\phe}{blue}{\c{mycolour}}{}{F}
\par After: {\testfont\phe{Snail in the Bottle}}
```

Before: **SNAIL IN THE BOTTLE**
After: **SNAIL IN THE BOTTLE**

[2.7] – ph6: Multi-level grouping without braces[a]

[a]“patchable” = it can be reconstructed from its decomposition under the current category code egime. – Manual, §7.1 (2018/05/02)

```
\newcommand{\mycolour}{brown}
\newcommand{\phf}[1]{\begingroup
\bfseries\begingroup\scshape\begingroup\color{blue}#1\endgroup\
    smallcaps\endgroup \ bold\endgroup\ normal \ \ }
Before: {\testfont\phf{Snail in the Bottle}}
\regexpatchcmd{\phf}{blue}{\c{mycolour}}{}{F}
\par After: {\testfont\phf{Snail in the Bottle}}
```

Before: **SNAIL IN THE BOTTLE SMALLCAPS bold** normal
After: **SNAIL IN THE BOTTLE SMALLCAPS bold** normal

\dca{x} : nostar; noopt=-NoValue-; MArg=$\boxed{\text{x}}$.
\dca*{y} : star; noopt=-NoValue-; MArg=$\boxed{\text{y}}$.
\dca[abc]{z} : nostar; OArg=abc; MArg=$\boxed{\text{z}}$.
\dca[xyz]{zz} : nostar; OArg=xyz; MArg=$\boxed{\text{zz}}$.

\dca*[xyzz]{zzz} : star; OArg=xyzz; MArg=$\boxed{\textbf{zzz}}$.

[2.8] – ph4a: Text replaced: ‘blue’ > ‘red’

```
\newcommand{\phda}[2]{{
\bfseries\scshape\color{blue}#1\normalcolor\ in the
    \color{blue}#2}}
Before: {\testfont\phda{Snail}{Bottle}}
\xpatchcmd{\phda}{blue}{red}{}{F}
\par After: {\testfont\phda{Snail}{Bottle}}
```

Before: **SNAIL IN THE BOTTLE**
After: **SNAIL IN THE BOTTLE**

[2.9] – ph4b: Text replaced: all ‘blue’ > ‘red’

```
\newcommand{\phdb}[2]{{
\bfseries\scshape\color{blue}#1\normalcolor\ in the
    \color{blue}#2}}
Before: {\testfont\phdb{Snail}{Bottle}}
\xpatchcmd*{\phdb}{blue}{red}{}{F}
\par After: {\testfont\phdb{Snail}{Bottle}}
```

Before: **SNAIL IN THE BOTTLE**
After: **SNAIL IN THE BOTTLE**