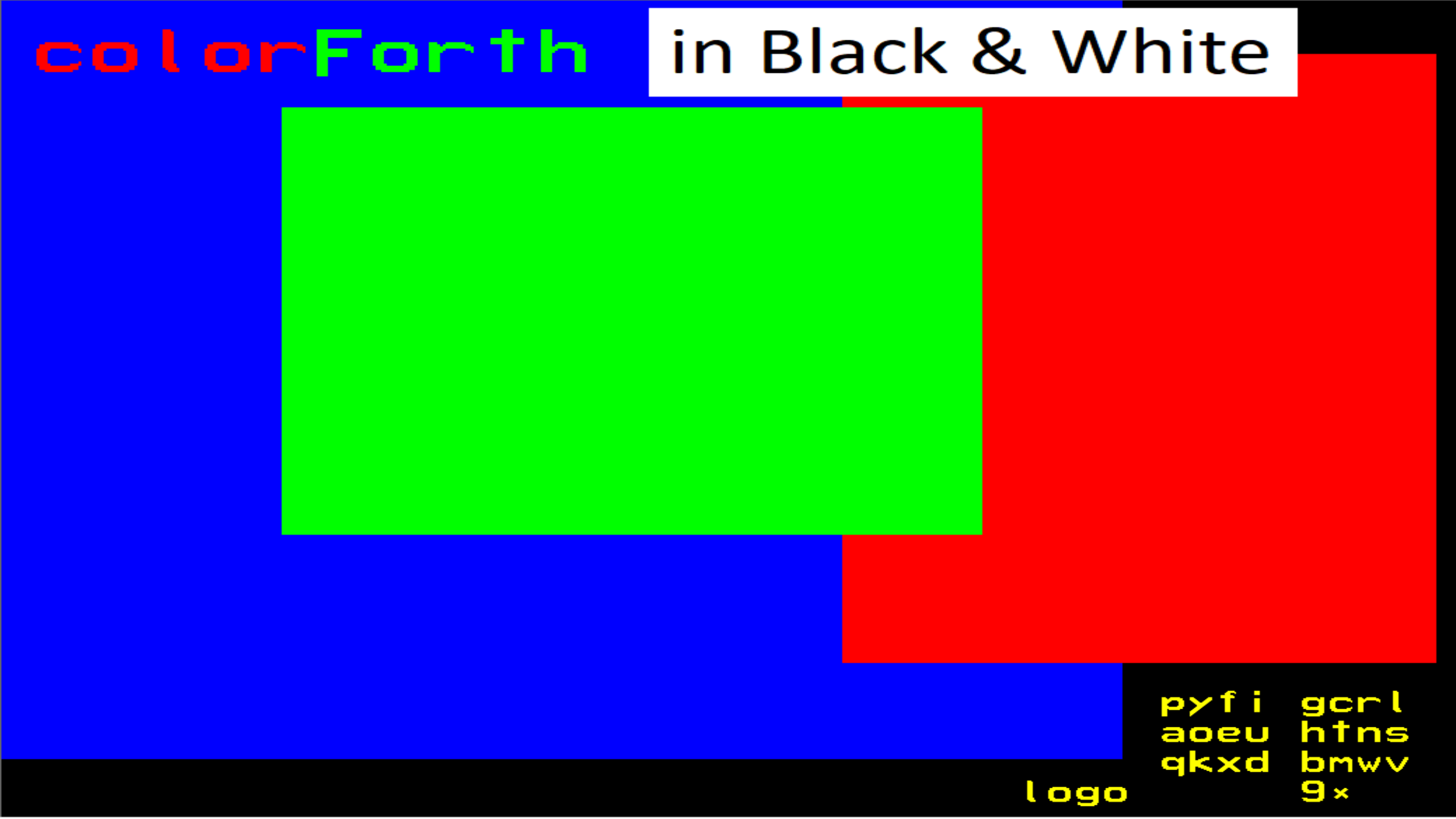

colorForth
in Black & White
pyfi gcrl
aoeu htns
qkxd bmwv
logo
9×

Shortening the Conceptual Gap

Shortening the Conceptual Gap

Edsger Dijkstra in his 1968 paper Go To Statement Considered Harmful states that:

“we should […] do our utmost to shorten the conceptual gap between the static program and the dynamic process […]”

Shortening the Conceptual Gap

Edsger Dijkstra in his 1968 paper Go To Statement Considered Harmful states that:

“we should […] do our utmost to shorten the conceptual gap between the static program and the dynamic process […]”

… which I interpret as “shorten the conceptual gap between source text and program execution”.

Shortening the Conceptual Gap

Edsger Dijkstra in his 1968 paper Go To Statement Considered Harmful states that:

“we should […] do our utmost to shorten the conceptual gap between the static program and the dynamic process […]”

… which I interpret as “shorten the conceptual gap between source text and program execution”.

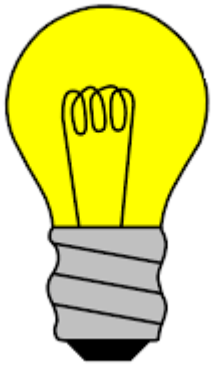
That is, make it as easy as possible for someone reading the source to create a conceptual model of what the program will do when it runs.

When Chuck Moore created colorForth one of his intentions was to use colour to replace punctuation:

Like this :

```
Editor Display cblind 0                                          228
cb cblind @ 0 + drop ; state 16)state× 16
yellow $ffff00 color ;
+txt white $6d emit space ;
-txt white $6e emit space ;
+imm yellow $58 emit space ;
-imm yellow $59 emit space ;
+mvar yellow $9 emit $11 emit $5 emit $1 emit space
;
```

While the use of colour to replace punctuation is an interesting idea…

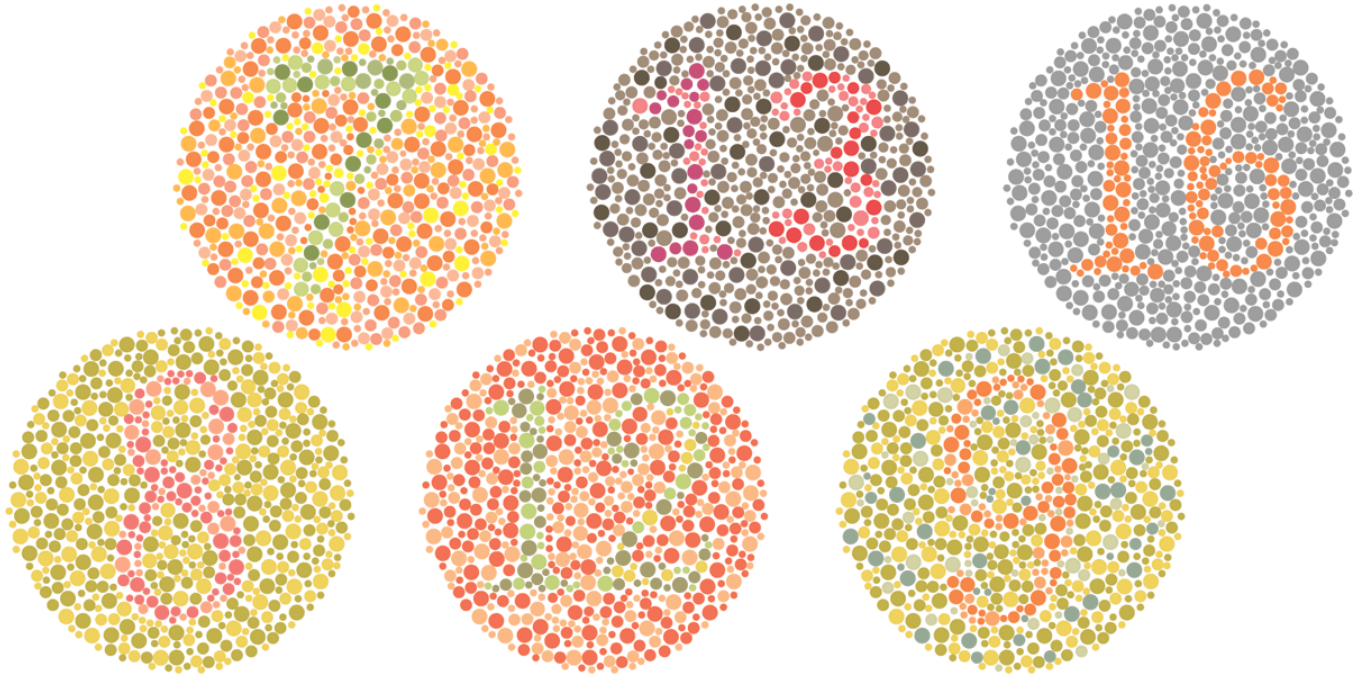


While the use of colour to replace punctuation is an interesting idea…

…it ultimately fails as a general-purpose programming language because a surprisingly high percentage of people are colour-blind.

…it ultimately fails as a general-purpose programming language because a surprisingly high percentage of people are colour-blind.

…it ultimately fails as a general-purpose programming language because a surprisingly high percentage of people are colour-blind.



…it ultimately fails as a general-purpose programming language because a surprisingly high percentage of people are colour-blind.

So why the interest in colorForth?

While the name “colorForth”, the coloured representation colorForth and the colourful appearance of the display all emphasise colour (spelled “color” in the USA), in fact the fundamental principles in colorForth go way beyond colour.

While the name “colorForth”, the coloured representation colorForth and the colourful appearance of the display all emphasise colour (spelled “color” in the USA), in fact the fundamental principles in colorForth go way beyond colour.

Colour in this context is just one way of conveying *meta-information* about a computer program.

While the name “colorForth”, the coloured representation colorForth and the colourful appearance of the display all emphasise colour (spelled “color” in the USA), in fact the fundamental principles in colorForth go way beyond colour.

Colour in this context is just one way of conveying *meta-information* about a computer program.

This meta-data can be used to control what the user sees in the editor, what the compiler compiles or what the interpreter does.

The colorForth colours and their meanings :

```
dd colour_orange  ; 0 extension token, remove space from previous word, do not change colour
dd colour_yellow  ; 1 yellow "immediate" word
dd colour_yellow  ; 2 yellow "immediate" 32 bit number in the following pre-parsed cell
dd colour_red     ; 3 red forth wordlist "colon" word
dd colour_green   ; 4 green compiled word
dd colour_green   ; 5 green compiled 32 bit number in the following pre-parsed cell
dd colour_green   ; 6 green compiled 27 bit number in the high bits of the token
dd colour_cyan    ; 7 cyan macro wordlist "colon" word
dd colour_yellow  ; 8 yellow "immediate" 27 bit number in the high bits of the token
dd colour_white   ; 9 white lower-case comment
dd colour_white   ; A first letter capital comment
dd colour_white   ; B white upper-case comment
dd colour_magenta ; C magenta variable
dd colour_silver  ; D
dd colour_blue    ; E editor formatting commands
dd colour_black   ; F
```

I am looking forward to discovering new ways of *simplifying* the total colorForth system by

I am looking forward to discovering new ways of *simplifying* the total colorForth system by

adding carefully controlled complexity into certain key areas :

I am looking forward to discovering new ways of *simplifying* the total colorForth system by

adding carefully controlled complexity into certain key areas :

- Version control

I am looking forward to discovering new ways of *simplifying* the total colorForth system by

adding carefully controlled complexity into certain key areas :

- Version control
- Multi-language

I am looking forward to discovering new ways of *simplifying* the total colorForth system by

adding carefully controlled complexity into certain key areas :

- Version control
- Multi-language
- Multi-user

I am looking forward to discovering new ways of *simplifying* the total colorForth system by

adding carefully controlled complexity into certain key areas :

- Version control
- Multi-language
- Multi-user
- Test framework

Editor

```
1
2 : squared ( n -- n )    dup * ;
3
```

File

: squared (n -- n) dup * ;

OS

include

Compiler

Output

Traditional Text Editor Forth

Editor

```
1
2 : squared ( n -- n )    dup * ;
3
```

File

: squared (n -- n) dup * ;

OS

include

Compiler

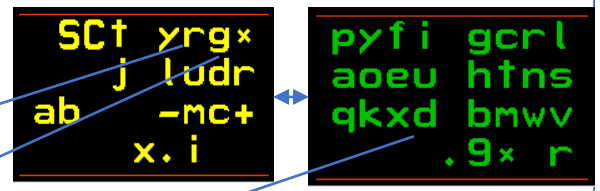
Parser

Output

Traditional Text Editor Forth

Editor F4

squared n-n dup * ;



Block

<r>squared <w>n-n <g>dup <g>* <g>;

Hardware

load

Compiler

Output

colorForth native mode

Editor F4

: squared (n – n) dup * ;

SCt yrg×
j ludr
ab -mc+
x. i

pyfi gcrl
aoeu htns
qkxd bmwv
.9× r

Block

<r>squared <w>n-n <g>dup <g>* <g>;

Hardware

load

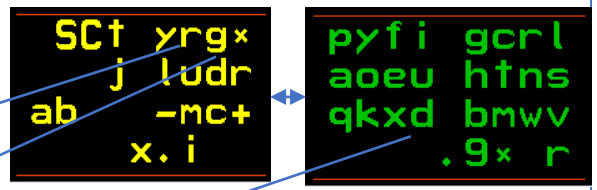
Compiler

Output

colorForth colour-blind mode

Editor F7

: zum-quadrat (n – n) dup * ;



Block

load

Compiler

<r>squared <w>n-n <g>dup <g>* <g>;

Hardware

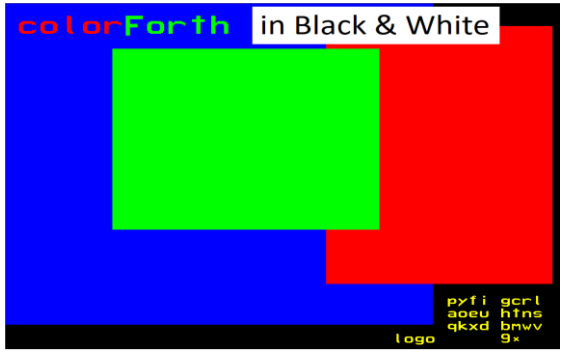
Output

colorForth Deutsch

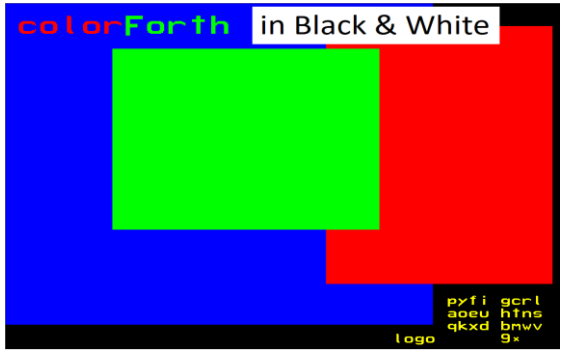
The possibilities are endless because

The possibilities are endless
because
„colorForth is infinitely powerful“

The possibilities are endless because „colorForth is infinitely powerful“

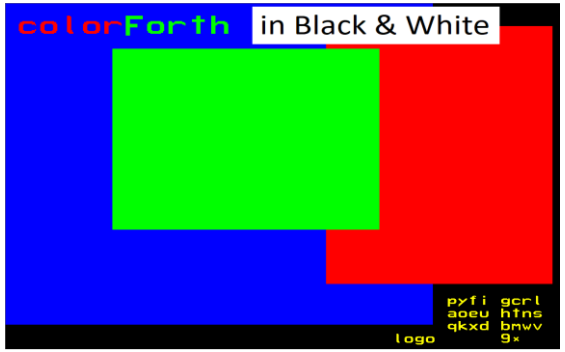


The possibilities are endless
because
„colorForth is infinitely powerful“



Questions?

The possibilities are endless
because
„colorForth is infinitely powerful“



www.inventio.co.uk/cf2019