

Code Golf Workshop in C

@ Labortage 2022 - Samstag (12.11.) 14:30

[madex](#) - Martin Boekhoff

Code Golf is a game designed to let you show off your code-fu by solving problems in the least number of characters.

Was ist Code Golf

- Ein (geheimer) Wettbewerb wo:
- das funktionierende Programm
- mit den wenigsten Zeichen (oder Bytes) gewinnt

Die Idee stammt aus der Programmiersprache *Perl* und wurde vorher auch oft *Script Golf* genannt

Wie bin ich zum Code golfen gekommen?

- [Obfuscated C Code Contest](#)
- demoscene, interesse an kleinen Programmen
- Warum ist der Overhead so groß. tcc
- [Advend of Code](#)
- [Programmiersprachen lernen mit exercism](#)
- was machen die besten bei AoC sonst noch?
- Schnell programmieren? [CodinGame](#)

VORSICHT

Codegolf ist zum Spaß an der Herausforderung. Auf der Arbeit nicht so programmieren

- write only Code, nicht wartbar
- man kann sich einen schlechten Codingstyle angewöhnen
- manche Tricks sind nicht portierbar!

Aber man lernt seine Programmiersprache besser kennen, wie sie wirklich funktioniert, und nicht nur wie man damit am sichersten programmiert.

<https://code.golf>

Bitte mit GitHub Account anmelden. Falls ihr noch keinen habt, bei GitHub anmelden, es lohnt sich.

Lasst uns FizzBuzz in C golfen

Print the numbers from 1 to 100 inclusive, each on their own line.

If, however, the number is a multiple of **three** then print `Fizz` instead, and if the number is a multiple of **five** then print `Buzz`.

If multiple conditions hold true then all replacements should be printed, for example 15 should print `FizzBuzz`.

straight forward aproach (162)

```
#include <stdio.h>
int main(){
for(int i=1;i<101;i++)
if(i%15<1)puts("FizzBuzz");
else if(i%3<1)puts("Fizz");
else if(i%5<1)puts("Buzz");
else printf("%d\n",i);
}
```

`<1` ist kürzer als `==0`

`puts()` ist 2 Byte kürzer und hängt `'\n'` an

remove #include, new lines, if, else, int

C (127)

```
main(i){for(;i<101;i++)if(i%15<1)puts("FizzBuzz");else if(i%3<1)puts("Fizz");  
else if(i%5<1)puts("Buzz");else printf("%d\n",i);}
```

- Der Linker linkt automatisch `printf` und `puts` auch ohne Prototyp.
- Funktionen haben als default `int` als Rückgabewert.
- Parameter haben als default `int` als Typ. Der erste Parameter ist `argc` und hier `1`. Keine Übergabeparameter.
- `n;` vor der main ist wie `int n = 0;` das geht auch mit arrays.

C improved (103)

```
main(i){for(;i<101;i++)i%15<1?puts("FizzBuzz"):i%3<1?puts("Fizz"):i%5<1?puts("Buzz"):printf("%d\n",i);}
```

Was wäre wenn wir die `<1` weg bekommen?
Bedingung negieren, da `!=0` ist gleich `true` => neu sortieren

C improved 2 (97/72)

```
main(i){for(;i<101;i++)i%15?i%3?i%5?printf("%d\n",i):puts("Buzz"):puts("Fizz"):puts("FizzBuzz");}
```

Was wäre wenn wir die mehrfachen Aufrufe von `puts()` wegbekommen?

Kann man das auf einen `puts` (für newline) und ein `printf` reduzieren?

Bitte versucht es selbst: *15 min*

Weitere Tricks

```
if(i)puts("hallo"); (19) =>
```

```
i?puts("hallo"):0; (18) =>
```

```
i&&puts("hallo"); (17)
```

```
if(!i)puts("hallo"); (20) =>
```

```
i||puts("hallo"); (17)
```

Python (72/58)

```
for x in range(1,101):print(("Fizz"if x%3<1else"")+("Buzz"if x%5<1else"")or x)
```

Divisors

A number is a divisor of another number if it can divide into it with no remainder.

Print the positive divisors of each number from 1 to 100 inclusive, on their own line, with each divisor separated by a space.

C optimized (71/60)

```
j;main(i){for(;i<101;i++,puts(" "))for(j=0;j++<i;)i%j||printf("%d ",j);} 
```

Versucht mal ob ihr eine for Schleife weg bekommt.

C optimized (65/60)

```
j;main(i){for(;i<101;)printf(j++<i?i%j?" ":"%d ":"  
"+(j-=++i),j);}
```

- Die Weise wie code.golf C Code speichert und compiliert ermöglicht das ersetzen von `\n` durch ein richtiges new line.
- es wird mir `"\n"+(j-=++i)` ein Newline-String definiert und parallel `j=0` und `i++`. +0 auf den `const char*` macht nichts.

JavaScript (59/49)

```
for(j=0,i=1;i<101;)write(j++<i?i%j?" ":j+" ":(j-=++i)||"\n")
```

Fibonacci

The Fibonacci numbers are a numerical sequence in which each number is the sum of the two preceding numbers:

0, 1, 1, 2, 3, 5, 8, 13 ...

Print the first **31** Fibonacci numbers from $F_0 = 0$ to $F_{30} = 832040$ (inclusive), each on a separate line.

C (57/47)

```
l, f; main(h) { for( ; f < 1000000; l = f, f += h, printf( "%d\n", h = l ) ); }
```

Versuche die lange Konstante kürzer zu bekommen oder durch die Anzahl zu ersetzen.

Versuche eine Variable loszuwerden.

JavaScript (36/33)

```
for(o=0,h=1;o<1e6;)print(h=(o+=h)-h)
```

Python (38/36)

```
a,b=0,1  
while a<1e6:print(a);a,b=b,b+a
```