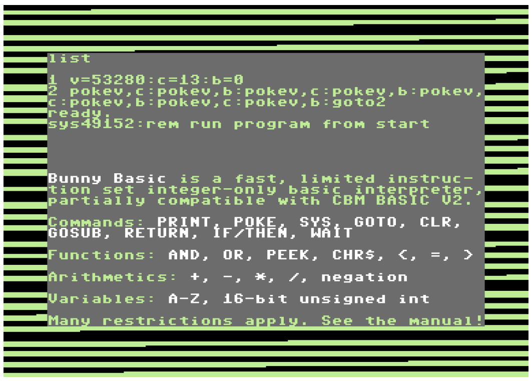
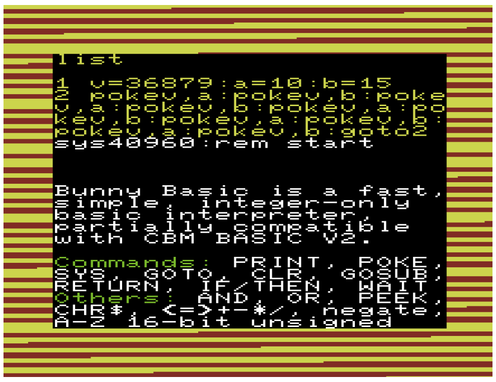
Bunny Basic

Bunny Basic is a relatively fast, limited instruction set, integer-only basic interpreter for Commodore 64 and Commodore VIC 20. It's partially compatible with CBM BASIC V2, but there's quite a few restrictions. However, you can run most Bunny Basic code in CBM BASIC V2 interpreter, it is just a dozen times slower.



```
list

1 v=53280:c=13:b=0
2 pokev,c:pokev,b:pokev,c:pokev,b:pokev,
c:pokev,b:pokev,c:pokev,b:goto2
ready.
sys49152:rem run program from start



Bunny Basic is a fast, limited instruc-
tion set integer-only basic interpreter,
partially compatible with CBM BASIC V2.

Commands: PRINT, POKE, SYS, GOTO, CLR,
GOSUB, RETURN, IF/THEN, WAIT

Functions: AND, OR, PEEK, CHR$, <, =, >

Arithmetics: +, -, *, /, negation

Variables: A-Z, 16-bit unsigned int

Many restrictions apply. See the manual!
```



```
list

1 v=36879:a=10:b=15
2 pokev,a:pokev,b:poke
v,a:pokev,b:pokev,a:po
kev,b:pokev,a:pokev,b:
pokev,a:pokev,b:goto2
sys40960:rem start



Bunny Basic is a fast,
simple, integer-only
basic interpreter,
partially compatible
with CBM BASIC V2.

Commands: PRINT, POKE,
SYS, GOTO, CLR, GOSUB,
RETURN, IF/THEN, WAIT
Others: AND, OR, PEEK,
CHR$, <=>+-*/, negate,
A-Z 16-bit unsigned
```

Commands: PRINT, POKE, SYS, GOTO, CLR, GOSUB, RETURN, IF/THEN, WAIT, END

Functions: AND, OR, PEEK, CHR$, <, =, >, <>, <=, >=

Arithmetics: 16-bit unsigned integer addition, subtraction, multiplication, division, negation

Variables: 26 one-letter variables A-Z, 16-bit unsigned integer

Biggest differences from CBM BASIC V2

Bunny Basic has no support for string variables or tables. Access memory directly with POKE and PEEK instructions instead to create larger data structures.

Only one arithmetic operation can be done at a single time. Arithmetics can be done only as separate instructions and not as regular expressions in command parameters.

PRINT command can display a fixed text or contents of a variable, but not both. Split the output to multiple PRINT's.

Whitespaces cannot be used. You will get a ?BUNNY ERROR READY.

IF instruction doesn't have an implied GOTO, so you must always type THENGOTO10.

GOTO and GOSUB can use a variable as destination. There's no FOR/NEXT instructions, use counter variables and IF/THEN/GOTO instead.

Running Bunny Basic programs

Use your standard CBM BASIC screen editor to write (and maybe even test) your programs. To run the program using Bunny Basic interpreter, call up the bunny with some SYS magic:

Commodore 64

```
SYS49152         (Run from start)
SYS49152,10      (Run starting from line 10)
```

Commodore VIC 20 (32K expansion, RAM at block 5)

```
SYS40960         (Run from start)
SYS40960,10      (Run starting from line 10)
```

You can stop the program with RUN/STOP key. Bunny Basic programs do not initialize variables at startup. Variable values can be random at start and they persist from previous program execution. Use the CLR command to zero all variables A-Z at start of program if desired.

When Bunny Basic reaches an END instruction, execution returns to CBM BASIC. You can mix CBM BASIC and Bunny Basic code in a single program, start with a RUN:

```
10 SYS49152,20:PRINT"BUNNY DONE!":END     (Executes Bunny Basic code from 20)
20 POKE198,0:WAIT198,1:POKE198,0:END      (Returns to CBM BASIC after END)
```

PRINT

Print a fixed string of text (or any usual PETSCII control codes, graphic symbols, etc.), the value of a variable, or a single CHR$ by character code. PRINT uses KERNAL CHROUT routine, so it's not really much faster than CBM BASIC.

Examples

```
PRINT"BUNNY BASIC RULES! ";
PRINT"THIS IS SHIT"
PRINTD;
PRINTCHR$(D);
PRINTZ
PRINTCHR$(A)
PRINTCHR$(147);
PRINT"VARIABLE Z IS ";:PRINTZ;
PRINT
```

Not Supported

```
PRINT"VARIABLE Z IS ";Z        (Can't combine text/variable, split to 2 PRINT's)
PRINT "TEXT"                   (Remove whitespace before " or get ?BUNNY  ERROR)
```

SYS

Call a machine code subroutine. To make things super easy, Bunny Basic copies the low 8 bits of variable A to accumulator before calling the machine code routine. Also, the low 8 bits of variables X and Y are copied to registers X and Y respectively.

When returning from SYS the contents of accumulator and registers X and Y are copied back to corresponding variables.

Examples

```
10 SYS32768
20 A=10:X=155:Y=128:SYS32000:PRINTA:PRINTX:PRINTY
```

Not Supported

```
10 SYS49152        (Restarting Bunny Basic from Bunny Basic makes a stack yoyo)
```

POKE

Set contents of a memory location or a chip register or whatever reachable with a POKE.

Examples

```
POKE53280,0
C=53280:POKEC,0
C=53280:A=11:POKEC,A
```

Not Supported

```
POKEC+1,0          (Sorry, no arithmetics here, too much for little Bunny!)
POKE 53280,0       (Whitespaces not allowed, it's a ?BUNNY  ERROR for you)
```

CLR

Clears all variables A-Z. Note that variables are not automatically reset when returning to Bunny Basic program from CBM BASIC.

Examples

```
10 CLR
20 A=10:PRINTA:CLR:PRINTA
```

GOSUB / RETURN

Call subroutine, return from subroutine. Bunny Basic supports 20 nested subroutine calls.

Examples

```
10 A=1:GOSUB100:A=2:GOSUB100:A=3:GOSUB100 (Call subroutine three times)
20 END
100 POKE53280,A:PRINTA:RETURN
```

Not Supported

```
POKEC+1,0         (Sorry, no arithmetics here, too much for little Bunny!)
POKE 53280,0      (Whitespaces not allowed, it's a ?BUNNY  ERROR for you)
```

Invisible "LET" / PEEK / Variables & Arithmetics

LET instruction is never typed in Bunny Basic. It's not actually even recognized. Every command starting with a variable name A-Z is interpreted as a LET instruction.

You can do only one calculation at once. Break your formula into smaller segments, like when writing machine code.

Examples

```
A=10         (Set value)
A=10+23      (Calculate sum of two numbers)
A=B+100      (Calculate sum of variable + number)
A=B+C        (Sum up two variables)
Z=C-B        (Subtract B from C, result in Z)

A=B*10       (16-bit unsigned multiplication)
A=B*C

C=C/10       (16-bit unsigned division)
A=C/B

A=-2         (Same as A=65534)
F=-F         (Negate value of F)

A=AAND511    (16-bit logical AND)
C=COR16      (16-bit logical OR)

A=PEEK(197) (Read contents of memory location 197 to A)
A=PEEK(X)    (Read contents of memory location X to A)
```

Not Supported

```
A=C+D+10     (Must break down into additions of two elements at a time)
LETA=10      (?BUNNY  ERROR for typing useless keyword)
A=PEEK(X+1) (Too much maths for little Bunny!)
```

GOTO

Jump to a line. Unlike CBM BASIC, you can also use a variable as target line in Bunny Basic.

Examples

```
10 PRINT"BUNNY POWER!";:GOTO10

5 A=10
10 PRINT"BUNNY SPECIAL!";:GOTOA
```

Not Supported

```
GOTOA+5           (Bunny says no)
```

IF/THEN

Conditionals. Also known as Artificial Intelligence. It's big business nowadays.

Examples

```
IFA=30000THENPRINT"YES!":POKE53280,0
IFA<100THENA=A+1
IFX>0THENX=X-1

IFC<>6THENGOSUB100:SYS64738
IFA<=10THENPRINT"A IS INDEED SMALLER OR EQUAL TO 10"
IFA>=11THENPRINT
```

Not Supported

```
IFA=10THEN30      (Little Bunny can't see invisible goto, write: THENGOTO30)
IFATHENGOTO10     (Bunny doesn't do logic very well)
IFPEEK(197)=3...  (Must read memory to a variable first, A=PEEK(197), etc.)
IFB><10THEN...    (Only <> is supported, Bunny is a little grumpy)
IFB=<10THEN...    (Only <= is supported, not =<)
IFB=>10THEN...    (Only >= is supported, not =>)
```

WAIT

Wait for contents of a memory location to change.

Examples

```
POKE198,0:WAIT198,1:POKE198,0 (Wait for a keypress)
WAIT653,2                     (Wait for Commodore key)
WAIT653,1                     (Wait for Shift key)
WAIT203,64,64                 (A fancy way to wait for a keypress)
WAITM,A,E                     (Memory location, AND-mask, optional EOR-mask)
```

Not Supported

```
WAITA+1,1                     (Little Bunny can't do calculations here)
```

END

End Bunny Basic program and return to CBM BASIC just after the SYS where you launched your Bunny code.

You can execute fragments of the same program listing using Bunny Basic interpreter and then return to CBM BASIC when you need a bigger arsenal of (slower) commands. See **Running Bunny Basic programs** on page 2.

Examples

```
530 END                     (End Bunny Basic program and return to CBM BASIC)

500 IFL=0THENPRINT"GOODBYE!":END          (Sweet!)
```

HAVE FUN!

Random future updates may or may not appear in Aleksi's Eight Big Shed:

http://bit.ly/eightbitshed
https://www.facebook.com/eightbitshed

Released 22 April 2019

Bunny Basic

Bunny Basic

Bunny Basic

Bunny Basic