

# Rosewell BASIC

**BASIC**  
**Reborn!**

The manual that  
helps you like  
it's 1982

Python Syntax  
for BASIC

macOS

Windows

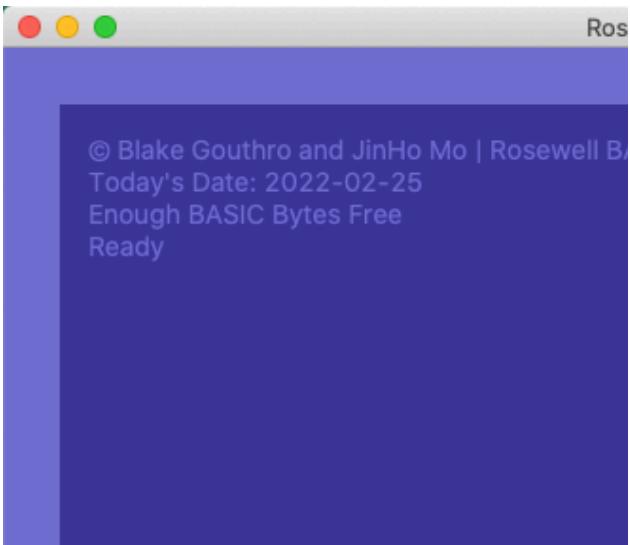
Linux

v2.1

# Table of Contents

Ready help -----Help Commands----- GoTo (num) - runs and repeats code (num) amou Help - list all the features of BASIC Save - Allows you to save BASIC programs to a f Run - allows you to run BASIC programs quit - Quits Rosewell BASIC List - list the PROGRAM saved in memory Load - loads an external PROGRAM from the San Load 8 - Loads a program from an External Disk Clear - Allows you to clear your PROGRAM code Dir - Lists the PROGRAM Directory ATSCII - Prints the Built-in ATSCII characters for all chr - Prints all of the characters at once Repeat Chr - can print All Chr PROGRAM if given	<ul style="list-style-type: none"><li>- Introduction Page. 1</li><li>- Programming Page. 2</li><li>- Programming Pt2 Page. 3</li><li>- Common Errors Page. 4</li><li>- GoTo - Feature Page. 4</li><li>- Help - Feature Page. 4</li><li>- Save - Feature Page. 5</li><li>- Run - Feature Page. 5</li><li>- Quit - Feature Page. 5</li><li>- List - Feature Page. 5</li><li>- Load - Feature Page. 6</li><li>- Load,8 Feature Page. 6</li></ul> <ul style="list-style-type: none"><li>- Clear - Feature Page. 6</li><li>- Dir - Feature Page. 7</li><li>- ATSCII - Program Page. 7</li><li>- All Chr -Program Page. 7</li><li>- Repeat Chr-Program Pg. 8</li><li>- Poke - Feature Page. 8</li><li>- Peek - Feature Page. 8</li><li>- Music - Program Page. 9</li><li>- Notes - Feature Page. 9</li><li>- Time/Date -Feature Pg.10</li><li>- Restart -Feature Page.10</li><li>- Secrets -Feature Page.10</li><li>- Text to Speech Page.11</li></ul> <ul style="list-style-type: none"><li>- Color Backgrounds Pg.11 and Pg.12</li><li>- —Notes— Page.12</li><li>- Special Thanks Page.12</li></ul>
---	--

# Introduction



So, you've Downloaded the Rosewell BASIC Folder that contains all of the Right Files for Rosewell BASIC, but the Problem is, you Don't Know what BASIC is and don't know how to use Rosewell BASIC. You've come to the right place!

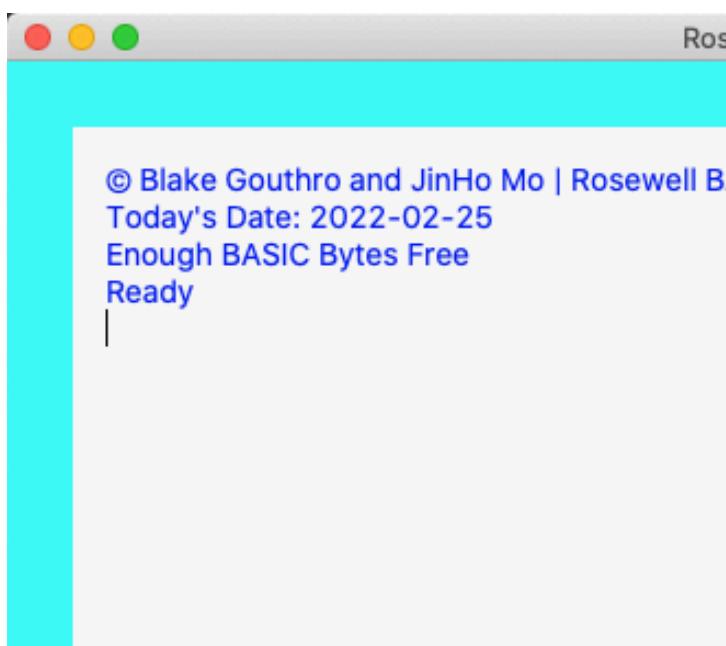
In this Manual you will learn how to do Basic things in BASIC and how to program a little bit. My name is CONRAD, I am an Artificial Intelligence and I am here to help You! Let's get Started.

BASIC Stands for Beginner's All-Purpose Symbolic Instruction Code. In other words, BASIC is a High-level Programming language used to control a computer from the 70's to 80's era. Some notable computers that use BASIC are; The Commodore 64, The Commodore PET, The Atari 8-bit computers(They had a cartridge with basic on it then eventually they had it built in.), The ZX-Spectrum, The Apple II, The Apple IIC and even the Apple I even know you had to load Applesoft BASIC from Cassette on the Apple I.

Rosewell BASIC is a Recreation of the BASIC Operating System for modern Computers. It is meant to be an Updated version of BASIC built on top of the PYTHON Programming Language. Rosewell BASIC uses PYTHON Syntax when you program on it but to save the program to system memory, you put a number infront of the line your coding. 'Ex. "10 print("Hello World")" output -> Hello World'. The line Numbers are from the older versions of the BASIC operating systems.

In Rosewell BASIC, you don't have to type big long lines of code to use any of the features, all you need is to type 'help' and a list of things you can type print on the screen. NOTE; The list of things you can type is ONE Variation of the way you can

# Programming in Rosewell BASIC



You Might be wondering how we can list the PROGRAM by typing 'list' in the Image next to this Paragraph. Well the Original BASIC Ones were in an era where they were the only OS in the computer, BASIC was inside of the ROM chip(Read Only Memory.), and BASIC could save programs to RAM(Random Access Memory) or Floppy Disk(Old CD or USB like storage.) Well as Rosewell BASIC is designed to run inside of any Python Compiler(the app that

Makes your code into a workable program.), we don't have access to saving to RAM chips from this OS because Rosewell BASIC is not on ROM chips, it is a PROGRAM, we have to find another way to save our PROGRAM. Fortunately for us PYTHON provides us with a solution for this, Variables. In BASIC I used a Variable called 'save'. Variables is pretty much PYTHON's Version of Built-in RAM. This variable is essentially my built in 'System memory'. When we type 'list', we read from the SYSTEM MEMORY and print whatever is saved there. When we type run, we run whatever program is saved in SYSTEM MEMORY. In the above picture you can see how I also use the line numbers to save lines to the SYSTEM MEMORY. Now that we understand how to save PROGRAMS, we can program something.

Try typing '10' first(BASIC Line numbers always goes up in 10's) and don't press ENTER. That number means we want to save to system memory and whatever we type next, we save to SYSTEM MEMORY. Then type 'print("Put whatever you want to program in here, don't forget the quotes!")' And press enter. You will see we've moved down to the next line and the 'print' statement we typed above will be saved in SYSTEM MEMORY. Type 'list' next. You will see that the OS has read the SYSTEM MEMORY and printed it on the next

# Programming in Rosewell BASIC

## Continued

Lines. Then you will want to run your BASIC Program. Once you're at the Ready Prompt, you can type 'run', and BASIC will run whatever's in SYSTEM MEMORY.

Example 2.

### **Internal Code:**

```
10 games = input("Do you like Video Games?: ")  
20 if games == 'yes':  
30   print("Awesome")  
40 elif games == 'no':  
50   print("Ok Then")  
List
```

### **What BASIC saves:**

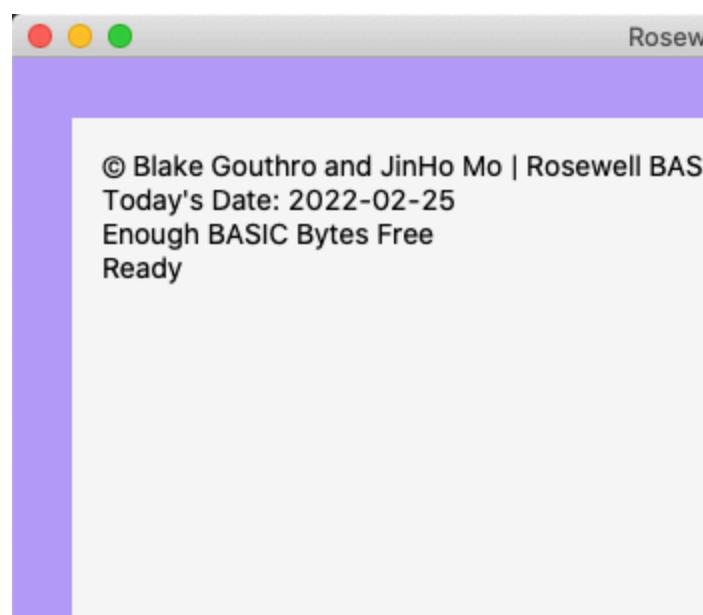
```
games = input("Do you like Video Games?: ")  
if games == 'yes':  
  print("Awesome")  
elif games == 'no':  
  print("Ok Then")
```

### **Output:**

```
Run  
Do you like video Games?:  
yes  
Awesome!
```

This program uses an Input and If/Elif/Else statements. An input will ask the user to type something. If/Elif/Else statements can structure your Responses to the input. If statements will always be checked. If the If statement is already went through, it will Skip the Elif and Else statements.

That was how to build very Basic Programs in PYTHON using Rosewell BASIC.



# Common Error Codes

```
Save - Allows you to save BASIC  
Run - allows you to run BASIC  
quit - Quits Rosewell BASIC  
List - list the PROGRAM saved  
Load - loads an external PROGRAM  
Load 8 - Loads a program from  
Clear - Allows you to clear your screen  
Dir - Lists the PROGRAM Directories  
ATASCII - Prints the Built-in ATASCII  
all chr - Prints all of the characters  
Repeat Chr - can print All Characters  
Poke - Print an ATASCII Character  
Peek - Return the value of an Address  
Music - Opens the Music Menu  
Note - Opens the Note Built in  
Time/Date/Calendar/Clock - Tel  
Restart - Restarts Rosewell BASIC  
?Syntax Error? - means something is wrong with your code  
?Dir Error? - The PROGRAM Either  
in the same Folder.  
  
Ready
```

## Common Error Codes;

- ?Syntax Error? This means that you would have typed some code or words incorrectly and is asking you to fix it before proceeding
- ?Dir Error? This is another Name error type error. Like syntax error, this would be found if you typed a Directory name in wrong or some sort of name.

## Goto - Feature

Goto, is a feature where you'll need to type 'Goto' along with a number. BASIC will then repeat running whatever is in SYSTEM MEMORY for the number that you typed in. If you typed 'Goto 10', it will then run whatever is in SYSTEM MEMORY 10 times.

## Help - Feature

Help, is a great feature if you want to use Rosewell BASIC to full Advantage. Typing 'Help' will print a list of commands you can type to do various things around BASIC.

# Save - Feature

The screenshot shows the Rosewell BASIC application window. At the top, there are three colored circles (red, yellow, green) and the text "Rosewell BASIC". Below that, it says "© Blake Gouthro and JinHo Mo | Rosewell BASIC | V2.1 | Ro". The main area displays the following code:

```
Enough BASIC Bytes Free
Ready
10 print("Rosewell BASIC")
list
print("Rosewell BASIC")

Ready
save
Enter new PROGRAM Name to Save [Type 'Quit' or 'Exit' to
Don't Forget to type '.py' to save as a python file. If your sav
d in the app ready to go!
test.py
Saving .....
Your PROGRAM Has Been Saved!
Ready
```

The application has a light orange background and a white text area.

Save, is a great feature for saving you PROGRAMS to an External PYTHON File. All you have to do is type 'Save', then enter the name of the PYTHON document, then it saves in the same folder that Rosewell BASIC is in. In the App Version, your programs are saved to the internal Resources Folder in the App.

# Run - Feature

Run is the Feature that starts the program that is save in SYSTEM MEMORY. Type 'Run' to use.

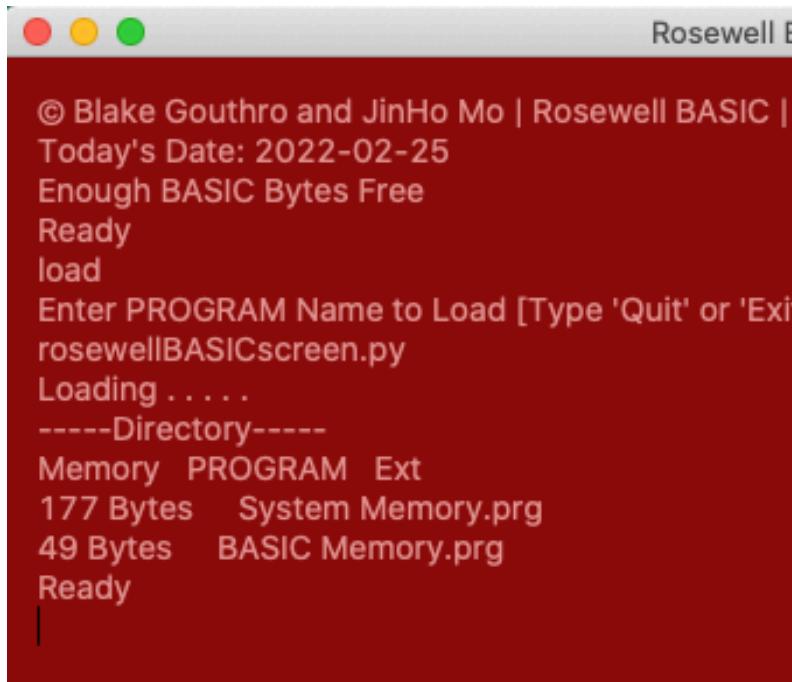
# Quit - Feature

The Quit feature should be pretty Obvious. Typing 'Quit' or 'Exit' will quit the OS and ALL program code in SYSTEM MEMORY WILL BE DELETED unless you Save to an external File.

# List - Feature

The List Feature will Access SYSTEM MEMORY and read whatever is in there. Then it will print the contents to the screen. NOTE; This will not RUN the Program stored in SYSTEM MEMORY

# Load - Feature



The screenshot shows a terminal window titled "Rosewell BASIC". The text output is as follows:

```
© Blake Gouthro and JinHo Mo | Rosewell BASIC |  
Today's Date: 2022-02-25  
Enough BASIC Bytes Free  
Ready  
load  
Enter PROGRAM Name to Load [Type 'Quit' or 'Exit'  
rosewellBASICscreen.py  
Loading .....  
----Directory----  
Memory PROGRAM Ext  
177 Bytes System Memory.prg  
49 Bytes BASIC Memory.prg  
Ready
```

The Load feature is a pretty cool Feature too. Typing 'Load' will then bring you to an input. Then you type the path to the file you want to load into BASIC. (NOTE; If the file is in the same folder, there is no need to type the path, just the file name. NOTE2; loading the File in, will write the file in the SYSTEM MEMORY so you can start the PROGRAM by typing 'Run'.) Loading the file in will also display a Directory with how much storage the PROGRAM will take up.

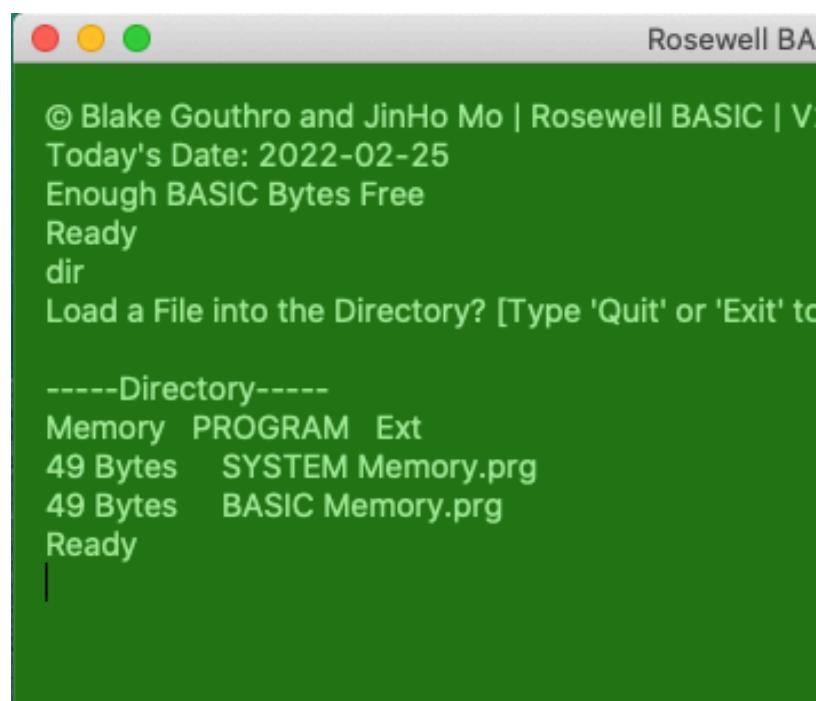
## Load, 8 - Feature

Load, 8 is a similar Feature to the Load Feature. Instead of loading in local files., you can plug in a USB Drive or SD Card or some sort of storage and load a file through there. There will be 2 inputs, 1 will ask you for the name of the Storage Device, and 2 will ask you for the program name in the usb.(NOTE: If the file is in a folder or 2 in the storage device, you will need to type the name and path after the name to the file not including the file name.(That's what input 2 is for.))

## Clear - Feature

The Clear Feature, is a smaller feature with 1 input. The clear feature, clears whatever is saved to SYSTEM MEMORY. Typing 'Clear' will bring you to an Are you sure input. Tying 'yes' will clear the SYSTEM MEMORY, And tying 'no' will not clear the SYSTEM MEMORY.

## Directory - Feature



The Directory Feature, will print a Directory of the SYSTEM MEMORY Contents and a few other Variables used for saving a couple of things. Type 'Directory' or 'Dir' to use.

## ATSCII -

ATSCII is a built in Character Set, named after the famous PETSCII Characters from the Commodore line of computers and

Unicode Character Sets. It contains;

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890`@&%'()\*+,.-./:;>=?\[{}]^\_`|~!ç£¤

## All Characters - Program

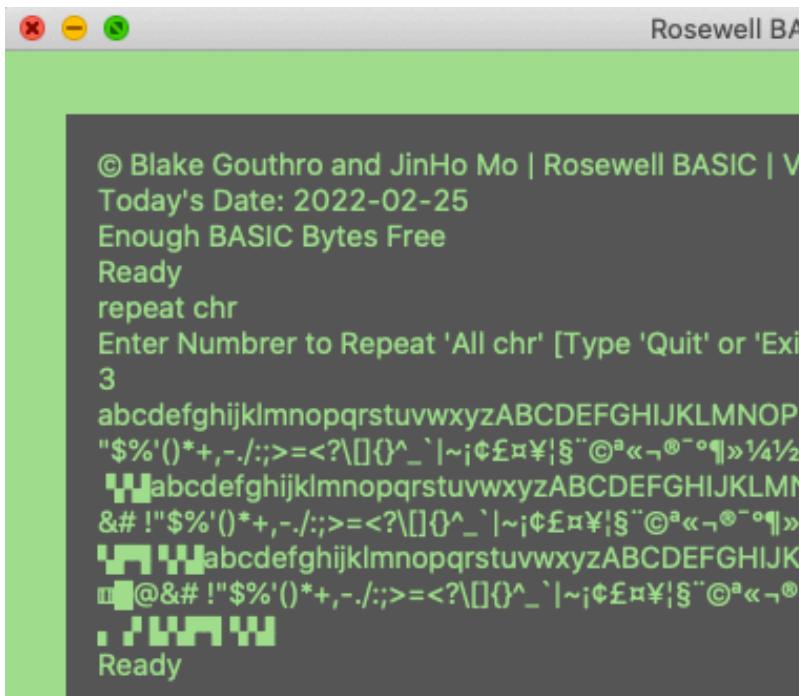
All Chr, is a PROGRAM similar to the ATSCII Program. Where instead of printing the ATSCII Characters line by line it prints them like;

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890□□□□□  
█ @!#\$%^&\*+,./:;>=<?\[{}^`|~¡¢£¤  
¥|§„©«¬®¬°¶»½¾¿×÷█ █ █ █ █ █ █ █ █ This PROGRAM  
was inspired by the first PROGRAM users would type into their  
Apple I computers on the WOZ MON Operating system. The PROGRAM  
would print all built in Characters. Type 'All Chr' to use.

© Rosewell Software 2021 - 2022 ©

Fun Fact: The WOZ MON Operating System is only 256 Kilo Bytes Pg. 7 in Size! We have MB and GB sized Operating Systems now.

# Repeat Chr - Program



The screenshot shows a window titled "Rosewell BA". Inside, there's a status bar at the top with icons for close, minimize, and maximize. The main area contains the following text:

```
© Blake Gouthro and JinHo Mo | Rosewell BASIC | V
Today's Date: 2022-02-25
Enough BASIC Bytes Free
Ready
repeat chr
Enter Number to Repeat 'All chr' [Type 'Quit' or 'Exi
3
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
$%"()*+,-./;:>=<?\[]{}^_`|~¡¢£¤¥¦§¨©¤«¬¤¬¤»½
WabcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
&#!"$%"()*+,-./;:>=<?\[]{}^_`|~¡¢£¤¥¦§¨©¤«¬¤¬¤»
VabcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
@#!"$%"()*+,-./;:>=<?\[]{}^_`|~¡¢£¤¥¦§¨©¤«¬¤¬¤»
, /W
Ready
```

The Repeat Chr PROGRAM is a similar PROGRAM to the All Chr PROGRAM. Typing 'Repeat Chr' will bring you to a number input. The only difference between the two PROGRAMS is the Repeat Chr PROGRAM can print the ATSCII Character set Multiple times by inputting any number you want.

## Poke - Feature

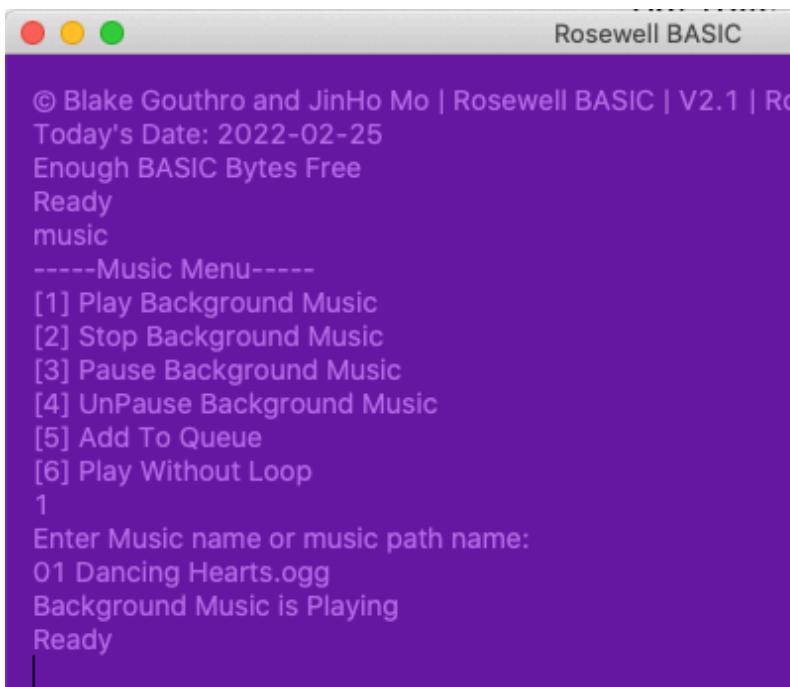
The Poke Feature is a small one. You must type 2 words for it to work. First Type 'Poke' then

Then Type the name of the ATSCII Character you want to 'Poke' to the screen.(This feature will print a Character on the Screen.)

## Peek - Feature

The Peek Feature will ask you for either Ord or Chr. First Type 'Peek' followed by a letter or a number. If you type a letter, then select ORD on the next input. If you type a number, then select either ORD or CHR on the next input, ORD will return the Ascii value of a Letter or Number, and CHR will return the Character of a number.

# Background Music - Program



The screenshot shows the Rosewell BASIC application window. At the top, it displays "Rosewell BASIC". Below that, there's a status bar with copyright information: "© Blake Gouthro and JinHo Mo | Rosewell BASIC | V2.1 | Ro". It also shows the date: "Today's Date: 2022-02-25". The main area of the window contains the following text:

```
Enough BASIC Bytes Free
Ready
music
-----Music Menu-----
[1] Play Background Music
[2] Stop Background Music
[3] Pause Background Music
[4] UnPause Background Music
[5] Add To Queue
[6] Play Without Loop
1
Enter Music name or music path name:
01 Dancing Hearts.ogg
Background Music is Playing
Ready
```

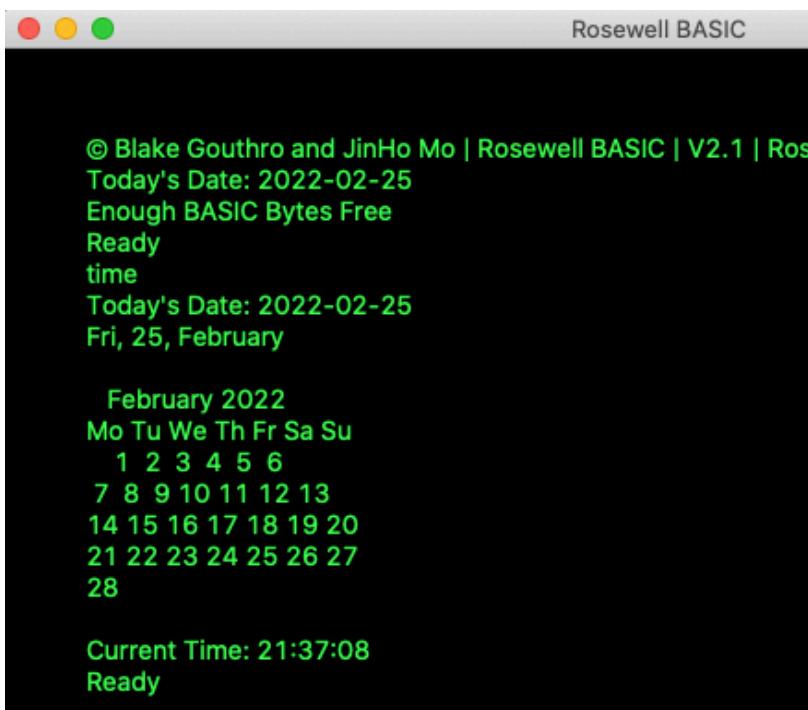
The Music PROGRAM is a Background Music PROGRAM that requires the use of the PYGAME PYTHON Module. This version of the PROGRAM, unlike the Rosewell OS version of the PROGRAM, is more limited to 2 Options. If you select Play Background Music, it will ask you for the path or file name of the music.(File name if the music is in the same folder as Rosewell BASIC. NOTE; The music PROGRAM only accepts WAV or OGG music files.) Then as it is

Background Music, it will play infinitely until you select option 2. Option 2, is Stop playing Background Music. As the name implies it stops playing the Playing Background Music. You can now Pause the Background music if it's not in looped mode, also you can add music to the queue to go through, and finally, play music without the loop. Type 'Music' to use.

## Notes - Feature

The Notes Feature, is quite simple. Option 1 will add a new note and delete the old note, Option 2 will allow you to Review the note. Option 3 will allow you to edit the note, and Option 4 will return you to the Ready prompt. Type 'Note' or 'Notes' to use.

## Time/Date - Feature

A screenshot of a computer screen displaying the Rosewell BASIC software interface. The window title is "Rosewell BASIC". The main area shows the following text:

© Blake Gouthro and JinHo Mo | Rosewell BASIC | V2.1 | Ros  
Today's Date: 2022-02-25  
Enough BASIC Bytes Free  
Ready  
time  
Today's Date: 2022-02-25  
Fri, 25, February  
  
February 2022  
Mo Tu We Th Fr Sa Su  
1 2 3 4 5 6  
7 8 9 10 11 12 13  
14 15 16 17 18 19 20  
21 22 23 24 25 26 27  
28  
  
Current Time: 21:37:08  
Ready

The Time/Date Feature is a small feature but a good one at that. Typing 'Time' or 'Date' will print the current Date, a calendar, and current Time before returning to the Ready Prompt.

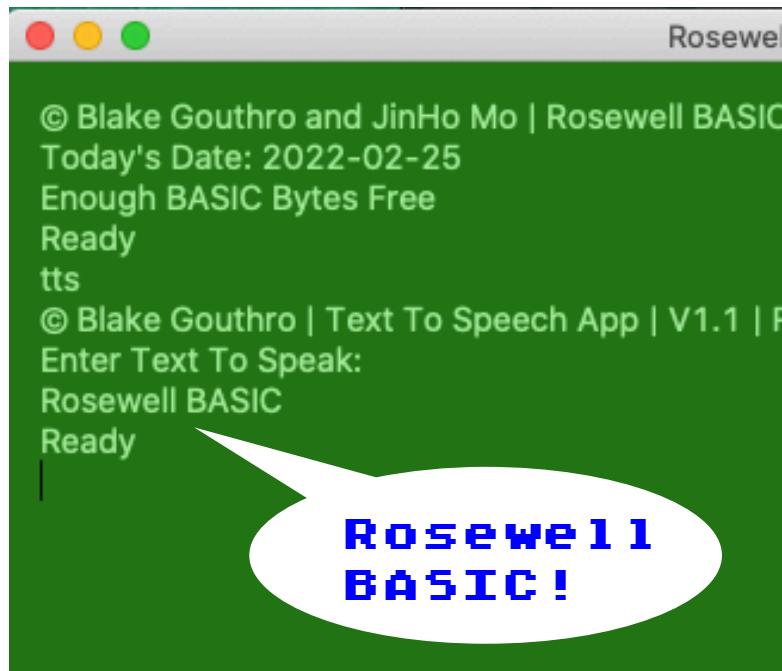
## Restart - Feature

And Last but not least, The Restart Feature. As the name implies, Typing 'Restart' will restart Rosewell BASIC BUT unlike Quitting, you Will save what is in SYSTEM MEMORY.

## Secrets

Like Rosewell OS, I can't tell you where the Secrets are in Rosewell BASIC but maybe Typing 'Secret' or 'Secrets' will get you started looking for them????

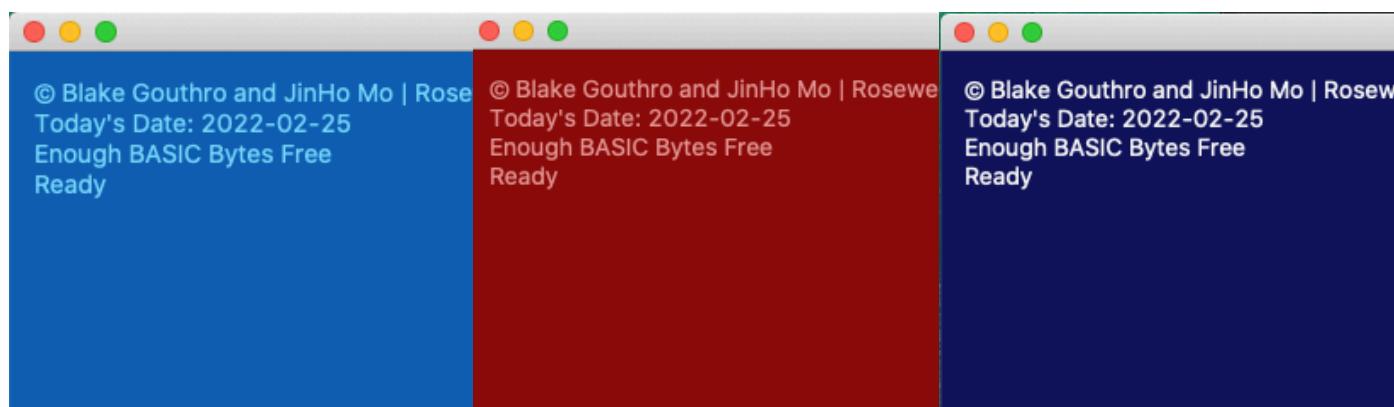
# Text To Speech - Feature



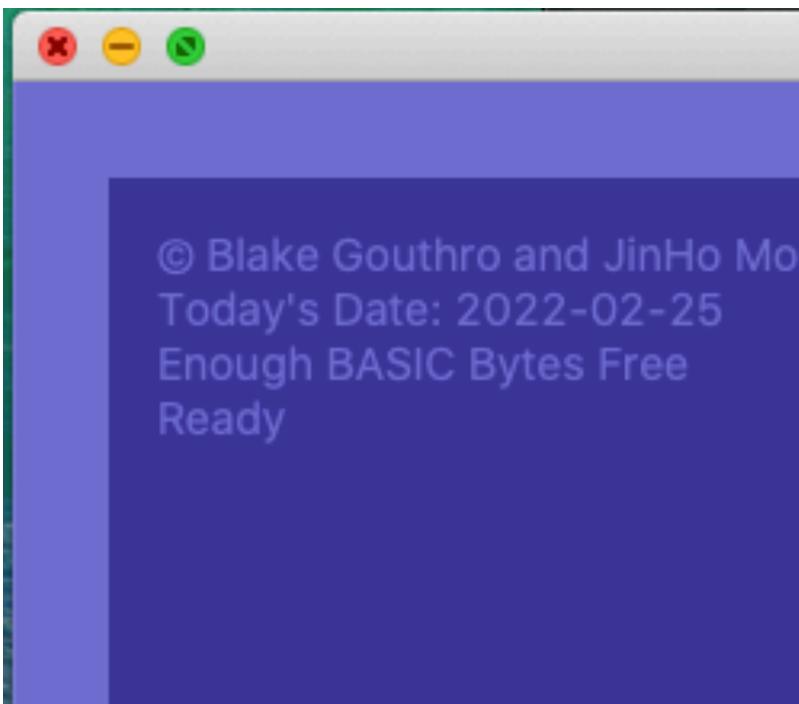
Starting with Version 2.0.3, a new feature called Text to Speech is included in the app. To use Text to speech, you type 'text to speech' or one of it's abbreviations 'tts' to enter the menu. Once you see "Enter Text to Speak:", you can type a 1 line sentence of any length and the app will say it out loud using the system's default voice then it will return to the ready prompt.

## Change Background Color - Feature

**Only on the APP Versions, not the Script Versions**



# Change Background Color Continued - Feature



## —Background Colours and Presets—

- Default or BASIC Blue
- Red or Dark Red
- Blue or Dark Blue
- Teal
- Orange
- Green
- Gray - Only Available in Gray plus white text. DOES NOT REQUIRE TYPING 'WHITE'
- Purple
- Yellow - 'white' will make the text black for this color
- Pink - 'white' will make the text black for this color
- White - no 'white' needed, uses Black text
- Black - no 'white' needed, uses White text
- Amber - No White Text Available
- Special Edition; Commodore 64
- Special Edition; Commodore Vic-20
- Special Edition; Commodore Plus-4
- Special Edition; Commodore 128
- Special Edition; Commodore PET
- Special Edition; Gameboy Peasoup
- Special Edition; Gameboy Light

Background Colours is a very personal choice when using Rosewell BASIC. Starting with Version 2.1, you can now change the Background Colours to 20 Different versions! That's a lot of colours. And it's not just colours. There are some presets that look like the nostalgic Commodore BASIC Screens. To change colours, you type 'color' followed by a colour name. You can type 'white' to get white or black text for that colour. Use colour help by typing 'color help'

# **Special Thanks!**



Blake Gouthro



JinHo Mo

**Special Thanks To;**  
**Blake Gouthro - C.E.O of**  
**Rosewell Software, Head**  
**Programmer, Head of**  
**Management, Rosewell Software**  
**Email Account Holder,**  
**Graphical Designer, Writer**

**JinHo Mo - Team Member of**  
**Rosewell Software, Programmer,**  
**Writer, Graphical Designer**

## **Notes**

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---