

Pink Floyd Assembly

Create a recipe for your very own Pink Floyd song! Packed with references to the band's best (and goofiest) lyrics!

Basic Instructions			
mneumonic	type / syntax	binary	instruction
fly	J-TYPE fly target	OPCODE: 0b000010	jump
ltf (learn to fly)	J_TYPE ltf target	OPCODE: 0b000011	jal
rick	R_TYPE rick \$t1, \$t2, \$t3	OPCODE: 0b000000 FUNCT: 0b000001	add
nick	I_TYPE nick \$t1, \$t2, -100	OPCODE: 0b000001	addi
stay	I_TYPE stay \$t1, -100(\$t2)	OPCODE: 0b000101	sw
sit	I_TYPE sit \$t1,-100(\$t2)	OPCODE: 0b000111	sb
wot	I_TYPE wot \$t1, -100(\$t2)	OPCODE: 0b001001	lw
echoes	R_TYPE echoes	OPCODE: 0b000000 FUNCT: 0b111111	syscall
run	R_TYPE run \$t1, \$t2	OPCODE: 0b000000 FUNCT: 0b100001	move
diff	I_TYPE diff \$t1, \$t2, \$t3	OPCODE: 0b000000 FUNCT: 0b000010	sub

'Songbite' Instructions			
mneumonic	syntax	binary	instruction
syd	R_TYPE syd	OPCODE: 0b000000 FUNCT: 101001	First & Second Call (\$s1<2): Print rand syd lyric Subsequent Calls: Print rand lyric about syd and how he is gone \$s0 ++ \$s1 ++
rog	R_TYPE rog	OPCODE: 0b000000 FUNCT: 101010	Rand rog quote \$s0 ++ \$s2 ++ Does nothing if ((s3 - \$s2) >= 5
gilmie	R_TYPE gilmie	OPCODE: 0b000000 FUNCT: 101011	Rand min, rand sec, "{min}:{sec} GUITAR SOLO" \$s0 ++ \$s3 ++ Does nothing if ((s2 - \$s3) >= 5
money	R_TYPE money	OPCODE: 0b000000 FUNCT: 101100	bass line
quit	R_TYPE quit	OPCODE: 0b000000 FUNCT: 101101	You can't quit! "THE SHOW MUST GO ON"
parry	R_TYPE parry	OPCODE: 0b000000 FUNCT: 101110	soulful sax solo
dogs	R_TYPE dogs	OPCODE: 0b000000 FUNCT: 101111	barking sounds
pigs	R_TYPE pigs	OPCODE: 0b000000 FUNCT: 1100000	Lyrics addressed to the pigs
stone	R_TYPE stone	OPCODE: 0b000000 FUNCT: 101000	"Dragged down by the stone" Stone repeats, then its echo repeats after 10 more songbite calls

'Songbite' Instructions			
mneumonic	syntax	binary	instruction
wall	R_TYPE wall	OPCODE: 0b000000 FUNCT: 110110	\$s5 = 1 Only "rog", "stone", and "quit" songbites work, because roger has built his wall and can no longer hear anyone but himself. This can be reversed by calling "trial" Print lyric depending on whether wall is already up or not
trial	R_TYPE trial	OPCODE: 0b000000 FUNCT: 110111	Does nothing if (\$s5==0) aka roger has not built his wall. Otherwise, set \$s5 = 0 and print lyrics detailing roger's trial

Used Registers \$

\$s0: Songbite total counter

Increments by 1 each time a Songbite Instruction successfully prints to console. Does not increment when logic prevents an Instruction from printing anything.

\$s1: syd counter

Increments by 1 each time a *syd* Instruction successfully prints to console. Does not increment when logic prevents the Instruction from printing anything. Is used to keep track of when to switch from *syd* quotes to *syd* references.

\$s2: rog counter

Increments by 1 each time a *rog* Instruction successfully prints to console. Does not increment when logic prevents the Instruction from printing anything. Is used to keep track of the absolute difference between the number of times *rog* and *gilmie* instructions have been printed.

\$s3: gilmie counter

Increments by 1 each time a *gilmie* Instruction successfully prints to console. Does not increment when logic prevents the Instruction from printing anything. Is used to keep track of the absolute difference between the number of times *rog* and *gilmie* instructions have been printed.

\$s4: stone start count

Is set to the current value of \$s0 when *stone* Instruction is called. Is used to keep track of when to print the *stone* echo.

\$s5: wall { 0:F, 1:T }

Begins set to 0. Is set to 1 when *wall* is called and set to 0 when *trial* is called. When set to 1, only *rog*, *quit*, and *stone* Songbite Instructions are able to successfully print to console.

Sample Code

Program A:

```
stone
rog
gilmie
```

This is a very short program demonstrating the advanced functionality of a few Instructions. The program will begin by printing a songbite that leads into the repetition of "STONE" in the console. Then, we have 10 'rog' Instructions which each print a random Roger lyric. On the 10th call, the program notices that it has now printed 10 songbites since it first repeated "STONE", so it decides to echo that command by repeating "stone" in the console. (This is a reference to how the repetition of the word stone on the 'Animals' album bleeds through into the song on the mirror side of the LP). Lastly, we call the 'gilmie' instruction to show how it does... nothing. Because the 'rog counter' got more than 5 ahead of the 'gilmie counter' David's instruction gets disabled for the rest of the program.

Program B:

```
gilmie
rog
gilmie
gilmie
gilmie
gilmie
rog
gilmie
gilme
gilmie
rog
```

This program demonstrates the functionality of the Roger/David feud in reverse. This time, by the final 'gilmie' instruction, the difference between the two rockstar's instruction counts has grown great enough to disable the 'rog' instruction. Therefore, every songbite in this program successfully prints except for the last one. This program shows how this language forces its programmers to balance the amount they reference each of the band's now estranged frontrunners.

Program C:

```
wall  
rog  
dogs  
pigs  
quit  
parry  
money  
trial
```

This short program begins with a songbite detailing Roger's construction of his metaphorical wall, then prints a random Roger lyric. Next it prints out "THE SHOW MUST GO ON" before finally printing the songbite detailing Roger's faux trial. The 'wall' instruction starting the program off is the reason half the instructions do nothing, those being 'dogs', 'pigs', 'parry', and 'money'. After the wall is raised, the only songbite instructions which can be metaphorically "heard" in the console are those intended to be from the point of view of Roger. These include: 'rog', 'stone', 'quit', 'wall', and 'trial'. If 'wall' was called in the middle of this program again before 'trial', it would not change the status of the wall but it would print an alternate songbite which references the wall having already been built. This program ends with a call to 'trial', which after printing the songbite details, subsequently allows any songbite to be "played". However no new songbites come (in this program) since this is the end of the very short program.