Programming Languages…

* A+
* A++
* A#
* A-0 programming language
* ABAP
* ABC
* ABC ALGOL
* ABLE
* ABSET
* ABSYS
* ACC
* Accent
* ACT-III
* ATOLL - Acceptance, Test Or Launch Language
* Action!
* ACS
* ActionScript
* Actor
* Ada

2000± languages omitted

* YAFL
* Yellow - Rejected prototype for Ada
* Yorick
* Y Language
* Z notation - A program specification language, like UML.
* ZPL
* ZZT-oop
* ZOPL
* ZUG

Befunge!

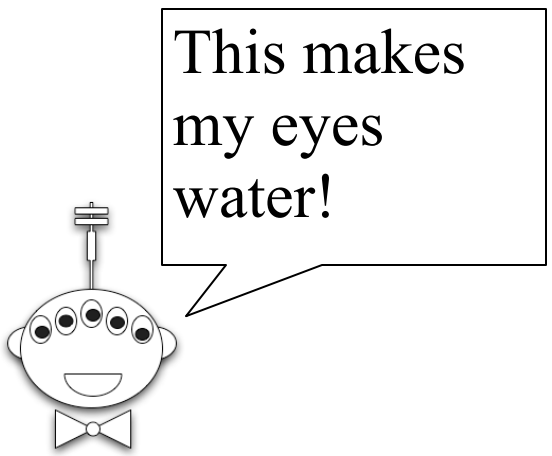
(Funge, Argh!, Befreak, Numberix, Weird)

Code sample: Befunge

Befunge is a synthetic language which may be useful for something one day, but I never hope to see any such day myself. The befunge interpreter starts reading the program at the upper left corner and moves toward the right, but it may be reversed with a **<** or sent upwards **^** and down **v** at will. Arithmetic operations use direct operands and a single stack that reminds me of the unix utility **dc**.

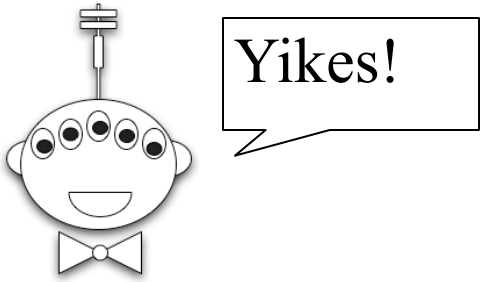
This program prints out the stirring lyrics of the famous camp song "99 Bottles of Beer on the Wall":

```
  9::*\2*+00p0v"."0<
>310p0","     >"llaw eht no "v  >#v_ ^
^_210p0"--:"                  v  ,
 :    v     " of beer"        <  :
 -              >"selttob"00g.^ <     <
 1         >00g1-#^_$" elttob erom enO" ^
      >00g#^_$" selttob erom oN"       ^
 ^_110p0",dnuora ti ssap ,nwod eno ekaT"^
  ^:-1_010p00g1-00pvv:-1g01_@#g00,*25<
     ^              <
```



Intercal

```
DO ,1 <- #13
PLEASE DO ,1 SUB #1 <- #234
DO ,1 SUB #2 <- #112
DO ,1 SUB #3 <- #112
DO ,1 SUB #4 <- #0
DO ,1 SUB #5 <- #64
DO ,1 SUB #6 <- #194
DO ,1 SUB #7 <- #48
PLEASE DO ,1 SUB #8 <- #22
DO ,1 SUB #9 <- #248
DO ,1 SUB #10 <- #168
DO ,1 SUB #11 <- #24
DO ,1 SUB #12 <- #16
DO ,1 SUB #13 <- #214
PLEASE READ OUT ,1
PLEASE GIVE UP
```



1.1 ORIGIN AND PURPOSE

The INTERCAL programming language was designed the morning of May 26, 1972 by Donald R. Woods and James M. Lyon, at Princeton University. Exactly when in the morning will become apparent in the course of this manual. It was inspired by one ambition; to have a compiler language which has nothing at all in common with any other major language. By 'major' was meant anything with which the authors were at all familiar, e.g., FORTRAN, BASIC, COBOL, ALGOL, SNOBOL, SPITBOL, FOCAL, SOLVE, TEACH, APL, LISP, and PL/I. For the most part, INTERCAL has remained true to this goal, sharing only the basic elements such as variables, arrays, and the ability to do I/O, and eschewing all conventional operations other than the assignment statement (FORTRAN "=").

1.2 ACRONYM

The full name of the compiler is "Compiler Language With No Pronounceable Acronym", which is, for obvious reasons, abbreviated "INTERCAL".

1.3 ACKNOWLEDGMENTS

Python

- Relatively “nice” syntax
- Emerging as language of choice in many fields
- Packages for graphics, audio, scientific computing, …

Python

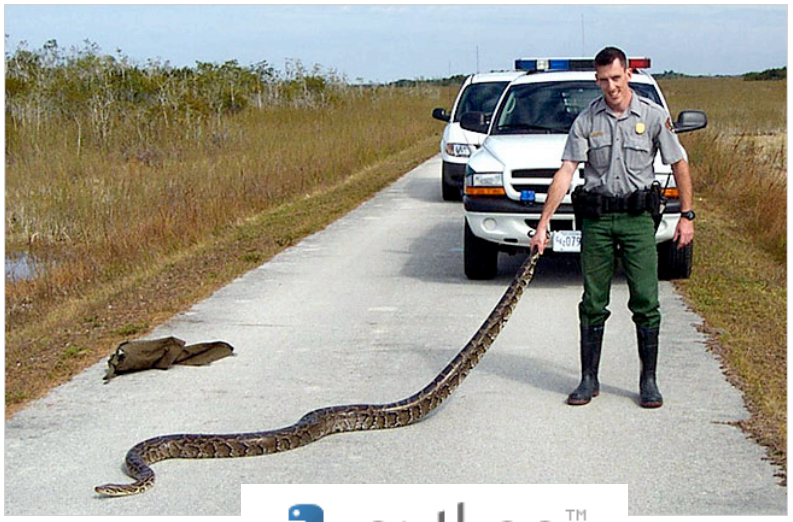
```
print("Hello World!")
```

Java

```
class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World!");
  }
}
```

Befunge

```
>              v
v  ,,,,,"Hello"<
>48*,          v
v,,,,,,"World!"<
>25*,@
```



Hello World…

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

C++

```
Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook.
Ook! Ook. Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook?
Ook! Ook! Ook? Ook! Ook? Ook. Ook. Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook. Ook! Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook! Ook. Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook! Ook.
Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook! Ook? Ook? Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook? Ook! Ook! Ook? Ook! Ook? Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook.
Ook? Ook. Ook? Ook. Ook? Ook. Ook? Ook. Ook! Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook! Ook. Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook.
Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook! Ook!
Ook! Ook. Ook. Ook? Ook. Ook? Ook. Ook. Ook! Ook. Ook! Ook? Ook! Ook! Ook? Ook!
Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook. Ook.
Ook. Ook. Ook. Ook. Ook! Ook.
```

Ook

Some things you'll do this semester…

Sequence alignment

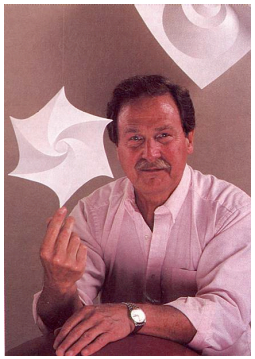
```
ATTATCG
ACATTC
```

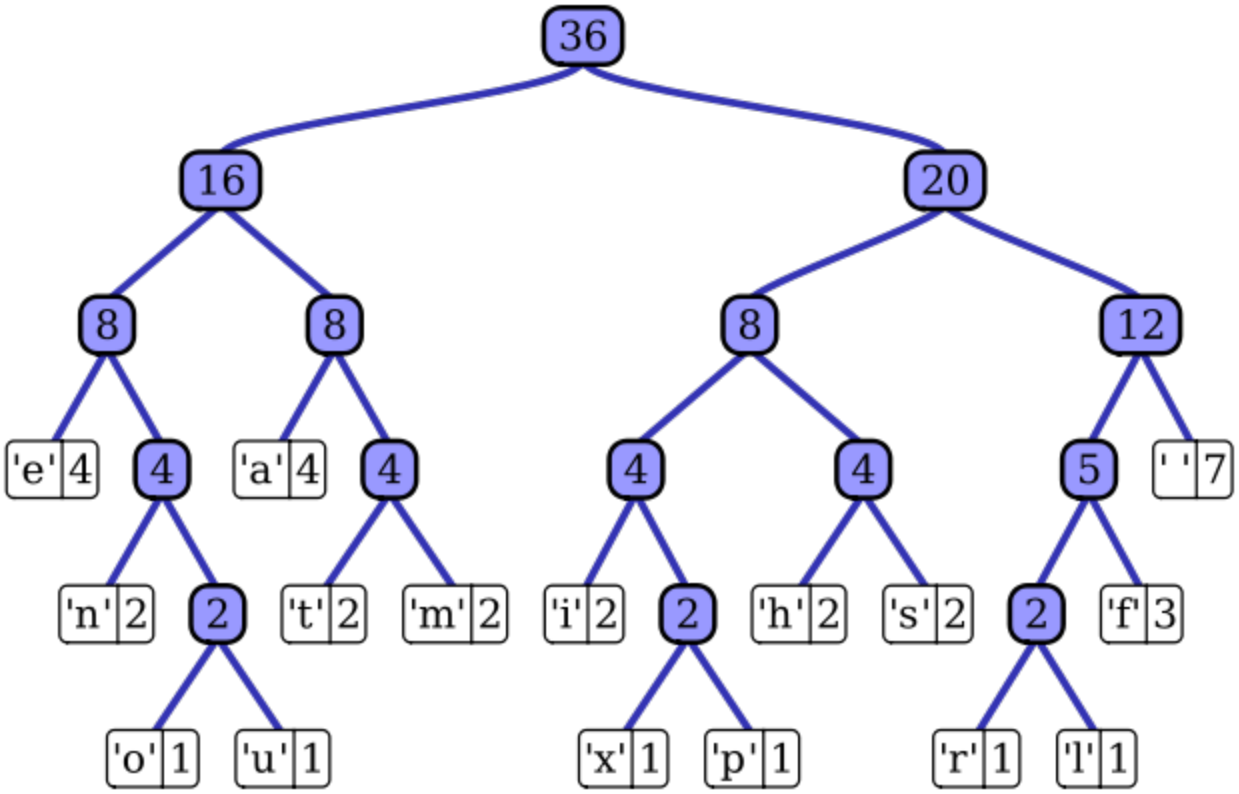
```
Distance is 4
ATTAT-CG
A-CATTC-
```

```
ATTATCG   ->   Delete T
A TATCG   ->   Change T to C
A CAT_CG  ->   Insert T here
A CATTCG  ->   Delete G
A CATTC
```

Spel Cheking…

Huffman Data Compression



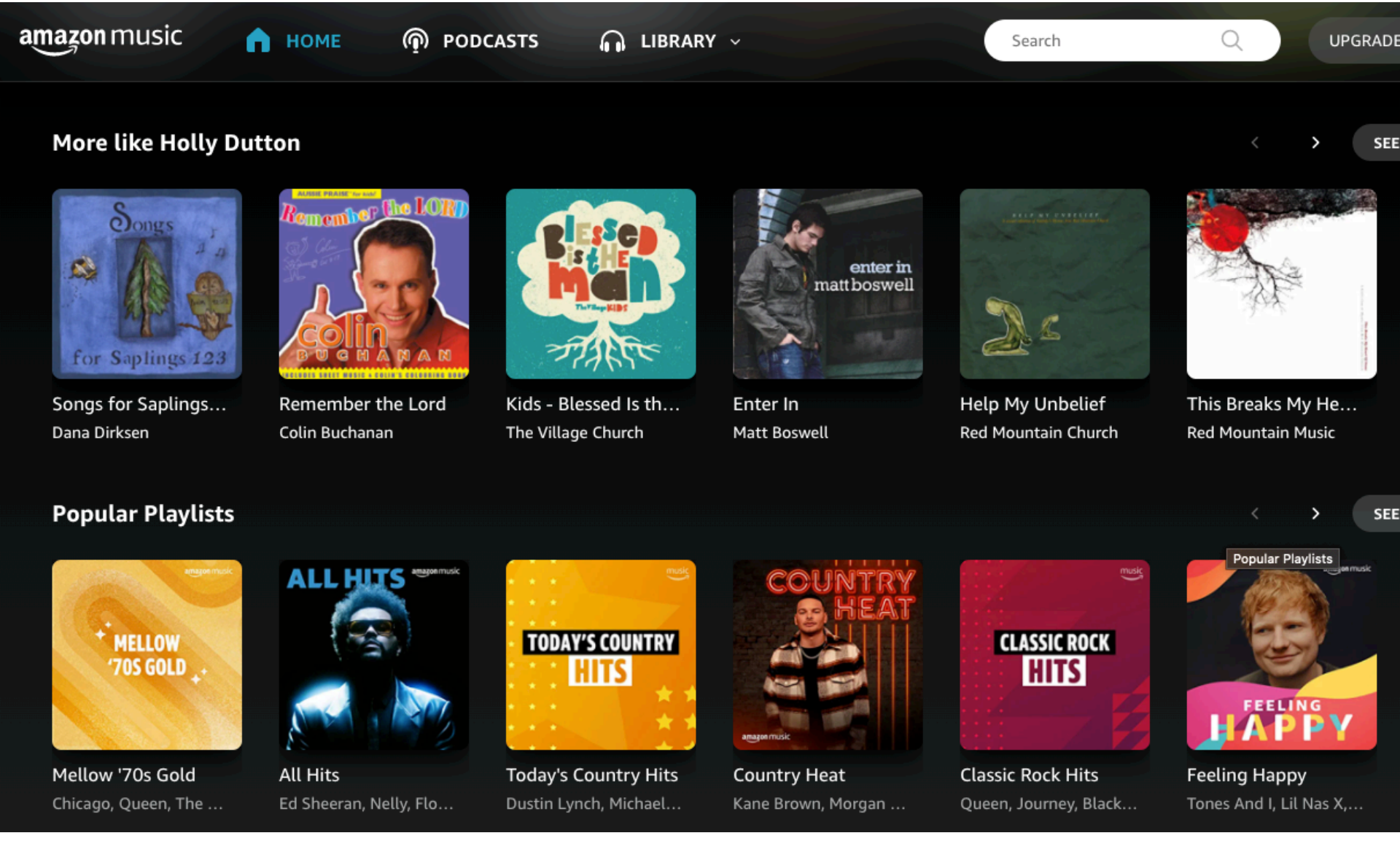


Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110
p	1	10011
r	1	11000
u	1	00111
x	1	10010

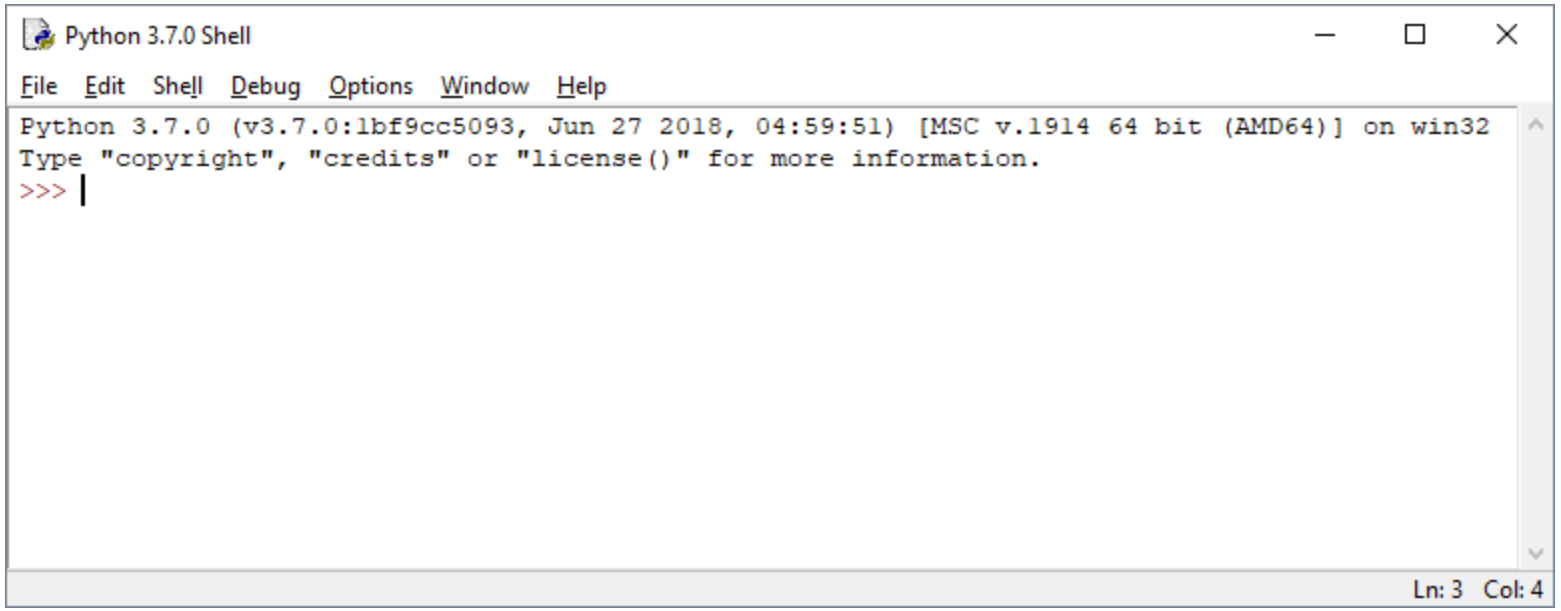
Connect 4 AI

Alien Intelligence?

Music Recommender System

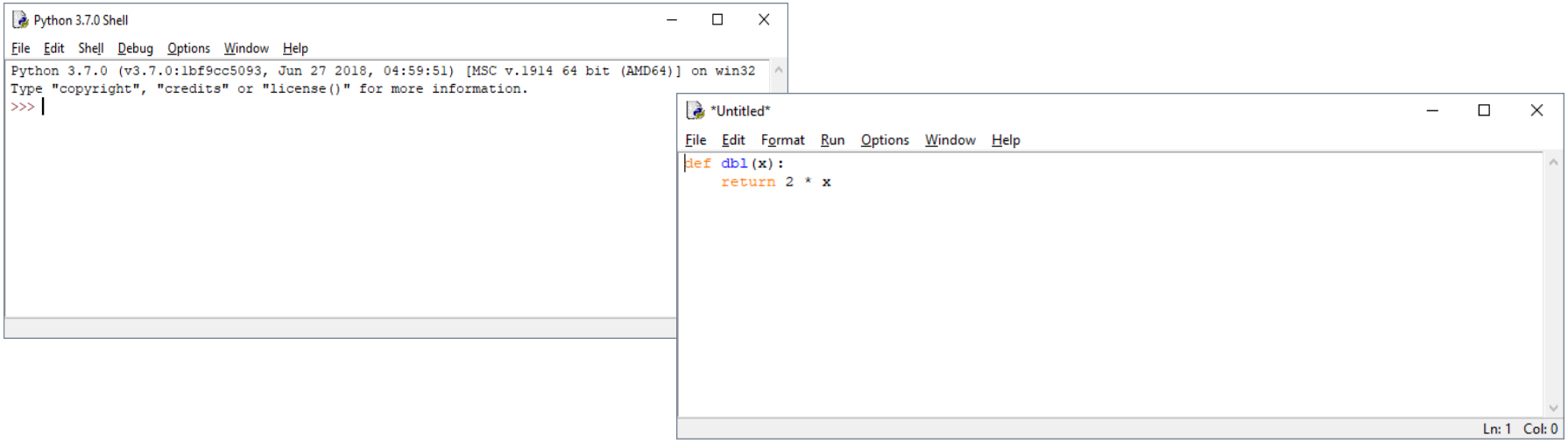


Python and IDLE



Shell window: Interacting with Python!

Python and IDLE



Shell window: Interacting with Python

Editor: Writing your own functions!

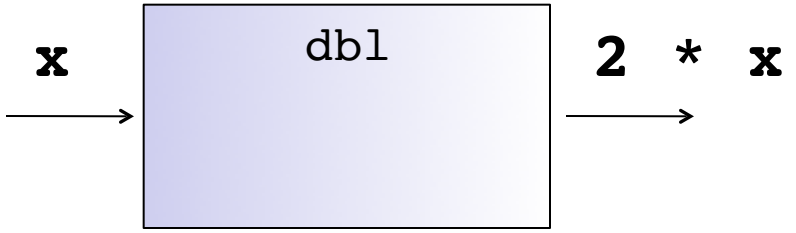
Invoke through “File” and then “New Window”

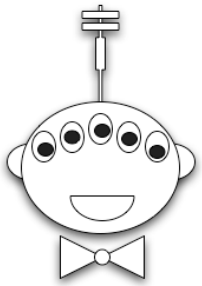
Save

Hit F5 to run

Defining your own functions!

```
def dbl(x):
    return 2 * x
```

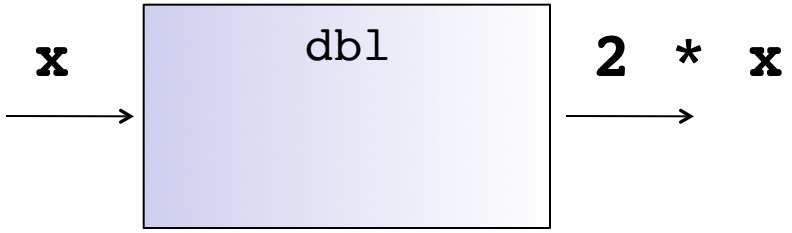




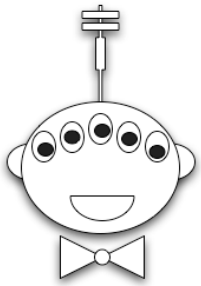
Notice the indentation. This is done using “tab” and it’s absolutely necessary!

Defining your own functions!

```
def dbl(x):
   return 2 * x
```



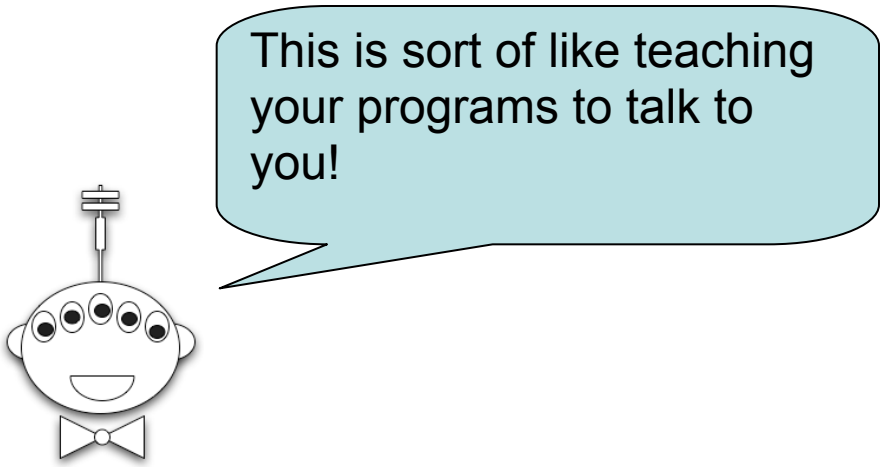

```
def dbl(myInput):
   myOutput = 2 * myInput
   return myOutput
```



Notice the indentation. This is done using “tab” and it’s absolutely necessary!

Docstrings!

```
def dbl(x):
    """This function takes a number x as input
    and returns 2 * x"""
    return 2 * x
```



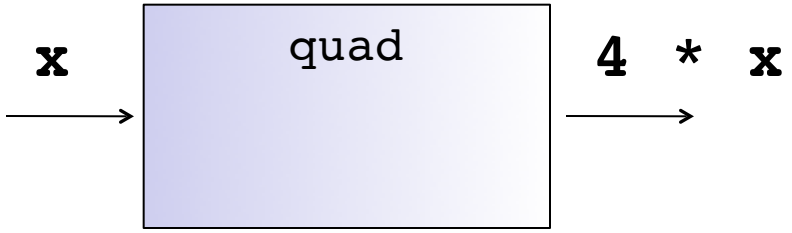
Docstrings… and comments

```
# Doubling program
# Authors: Ran Libeskind-Hadas
# Date:    August 27, 2011

def dbl(x):
  """This function takes a number x as input
  and returns 2 * x"""
  return 2 * x
```

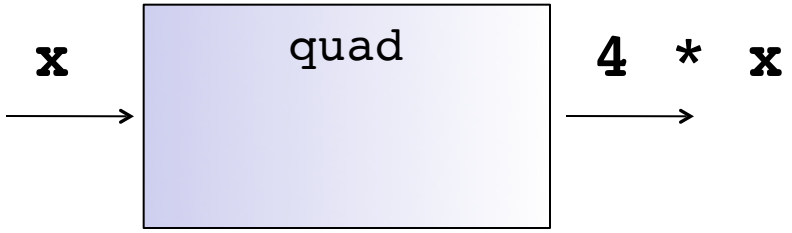
Composition of functions

```
def quad(x):
    return 4 * x
```

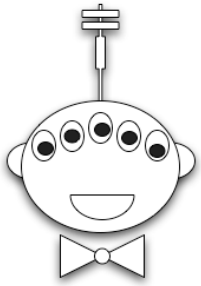


Composition of functions

```
def quad(x):
    return 4 * x
```

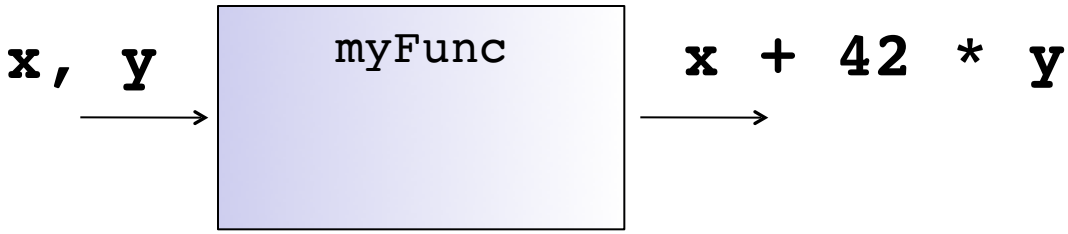


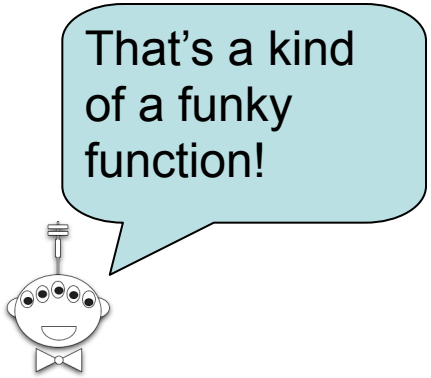

```
def quad(x):
    return dbl(dbl(x))
```



Doubly cool! (draw the boxes)

Multiple inputs...






```
# myFunc
# Authors: Ran Libeskind-Hadas
# Date:    August 27, 2012

def myFunc(x, y):
  """returns x + 42 * y"""
  return x + 42 * y
```