

RYKA|CBT

پدرام شاه صفی
pd.Shahsafi@gmail.com



python

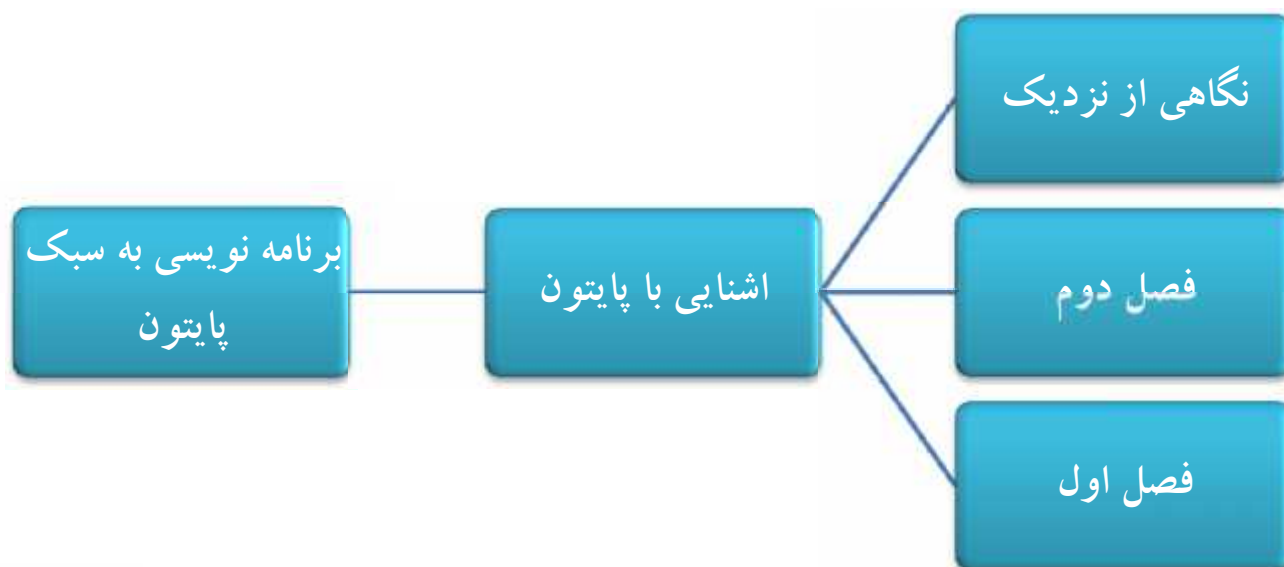


به نام پروردگار دانایی

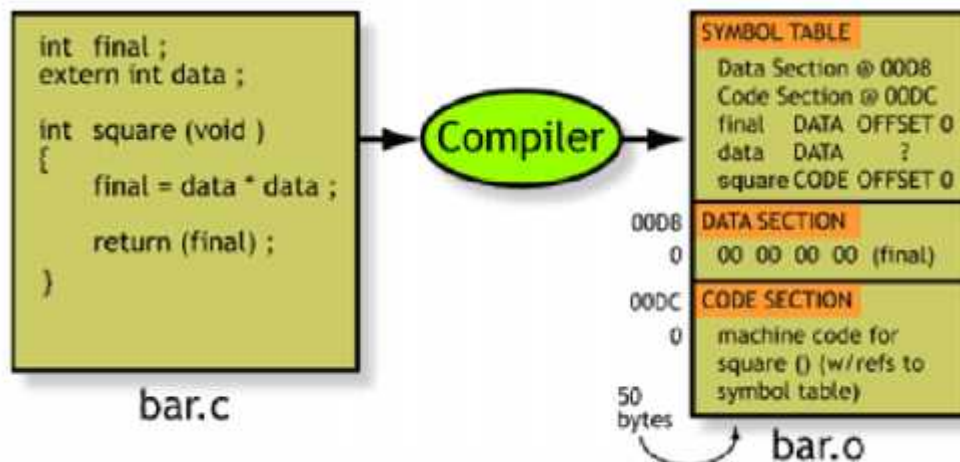
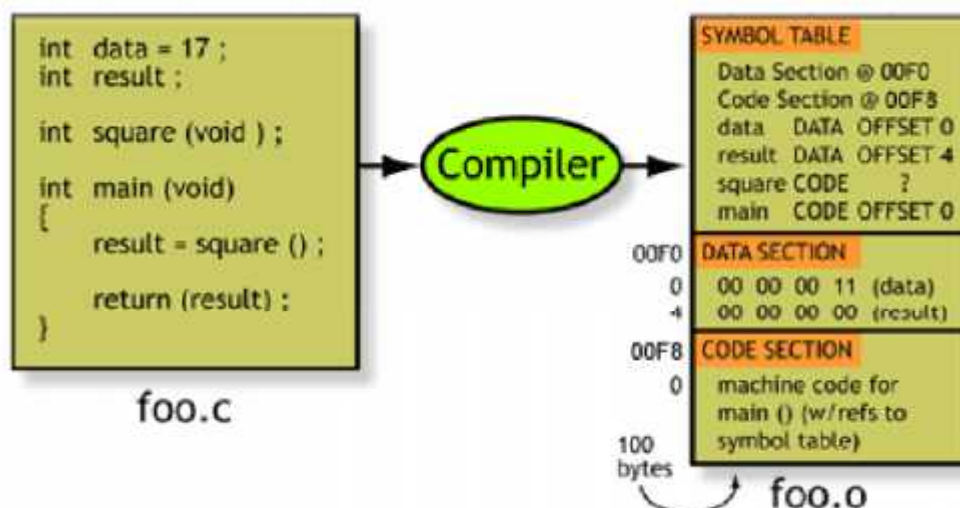
برنامه نویسی به سبک پایتون

پدرام شاه صفی

بهار ۱۳۹۴



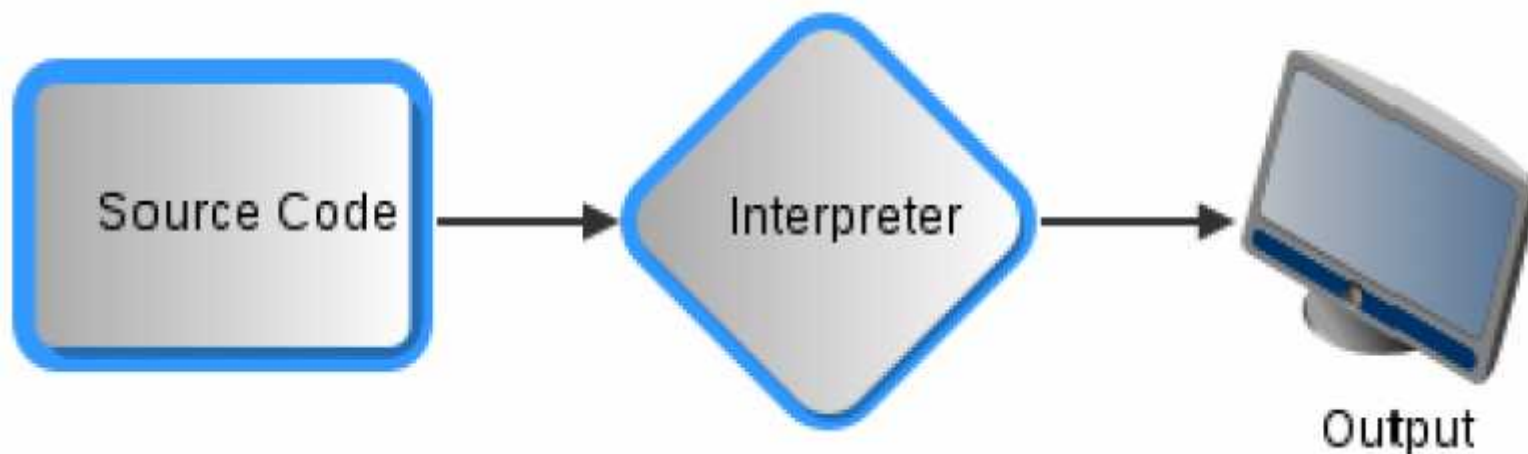
مراحل کامپایلر



- برنامه نوشته میشود.
- کامپایلر آنرا به زبان ماشین تبدیل میکند
- فایل های کد ماشین بهم لینک میشوند
- فایل اجرایی به وجود می آید.

اینترپرتر

- برنامه نوشته میشود.
- برنامه خطایابی یا اجرا میشود.

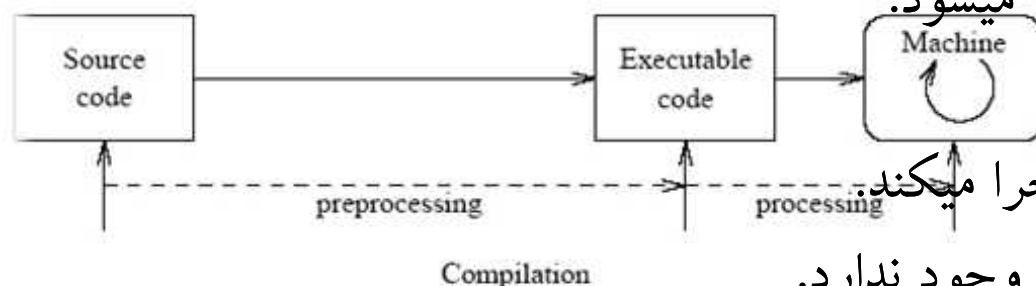


مقایسه

- برنامه بطور کلی یا کامپایل (compiled) میشود یا اینترپرت (interpreted) میشود.

- کامپایلر یک فایل اجرایی میسازد.

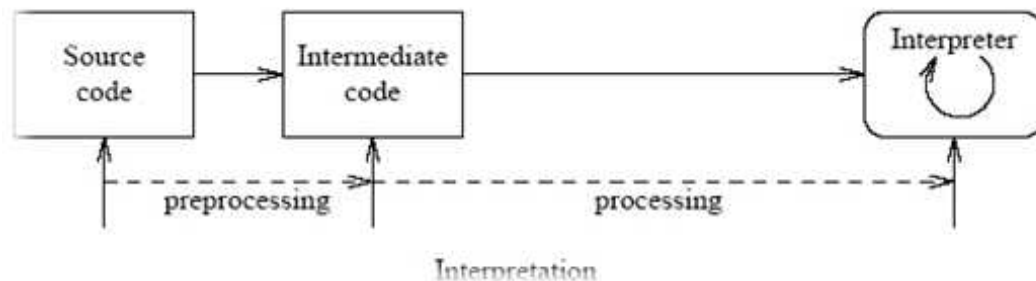
- در دفعه بعدی فایل اجرایی اجرا میشود.



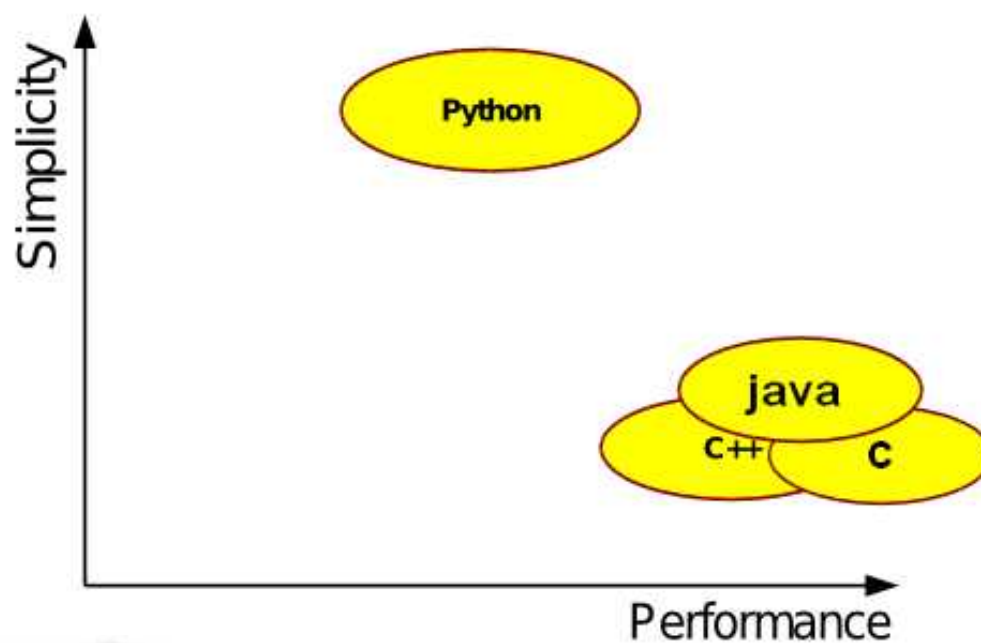
- اینترپرتر خطی از کد را ترجمه و اجرا میکند.

- در دفعه بعدی فایلی ذخیره شده وجود ندارد.

- پس باید دوباره عمل بالا انجام شود.



پایتون vs سی vs جاوا



پایتون vs جاوا

```
import random

def _doquicksort(values, left, right):
    def partition(values, left, right, pivotidx):

        pivot = values[pivotidx]
        values[right], values[pivotidx] = values[pivotidx], values[right]

        storeidx = left
        for idx in range(left, right):
            if values[idx] < pivot:
                values[idx], values[storeidx] = values[storeidx], values[idx]
                storeidx += 1

        values[storeidx], values[right] = values[right], values[storeidx]
        return storeidx

    if right > left:
        pivotidx = random.randint(left, right)
        pivotidx = partition(values, left, right, pivotidx)
        _doquicksort(values, left, pivotidx)
        _doquicksort(values, pivotidx + 1, right)

    return values

def quicksort(mylist):
    return _doquicksort(mylist, 0, len(mylist) - 1)
```

```
import java.util.ArrayList;
import java.util.Random;

public class QuickSort {

    public static final int NUMBERS_TO_SORT = 25;

    public QuickSort() {
    }

    public static void main(String[] args) {
        ArrayList<Integer> numbers = new ArrayList<Integer>();
        Random rand = new Random();
        for (int i = 0; i < NUMBERS_TO_SORT; i++)
            numbers.add(rand.nextInt(NUMBERS_TO_SORT + 1));
        for (int number : numbers)
            System.out.print(number + " ");
        System.out.println("\nBefore quick sort\n\n");
        for (int number : quicksort(numbers))
            System.out.print(number + " ");
        System.out.println("\nAfter quick sort\n\n");
    }

    public static ArrayList<Integer> quicksort(ArrayList<Integer> numbers) {
        if (numbers.size() <= 1)
            return numbers;
        int pivot = numbers.size() / 2;
        ArrayList<Integer> lesser = new ArrayList<Integer>();
        ArrayList<Integer> greater = new ArrayList<Integer>();
        int sameAsPivot = 0;
        for (int number : numbers) {
            if (number == numbers.get(pivot))
                greater.add(number);
            else if (number < numbers.get(pivot))
                lesser.add(number);
            else
                sameAsPivot++;
        }
        lesser = quicksort(lesser);
        for (int i = 0; i < sameAsPivot; i++)
            lesser.add(numbers.get(pivot));
        greater = quicksort(greater);
        ArrayList<Integer> sorted = new ArrayList<Integer>();
        for (int number : lesser)
            sorted.add(number);
        for (int number : greater)
            sorted.add(number);
        return sorted;
    }
}
```





Coding. Crawler



Java

```

public class CrawlerExample {

    public static void main(String[] args) throws IOException {
        PrintWriter textFile = null;
        try {
            textFile = new PrintWriter("result.txt");
            System.out.println("Enter the URL you wish to crawl..");
            System.out.print("> ");
            String myUrl = new Scanner(System.in).nextLine();

            String response = getContentByUrl(myUrl);

            Matcher matcher = Pattern
                .compile("href=\\\"(.+[\\\"\\']+[\\\"\\'])\\\"").matcher(response);
            while (matcher.find()) {
                String url = matcher.group(1);
                System.out.println(url);
                textFile.println(url);
            }
        } finally {
            if(textFile != null) {
                textFile.close();
            }
        }
    }

    private static String getContentByUrl(String myUrl)
        throws IOException {
        URL url = new URL(myUrl);
        URLConnection urlConnection = url.openConnection();
        BufferedReader in = null;
        StringBuilder response = new StringBuilder();
        try {
            in = new BufferedReader(new InputStreamReader
                (urlConnection.getInputStream()));
            String inputLine;
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
        } finally {
            if(in != null) {
                in.close();
            }
        }
        return response.toString();
    }
}

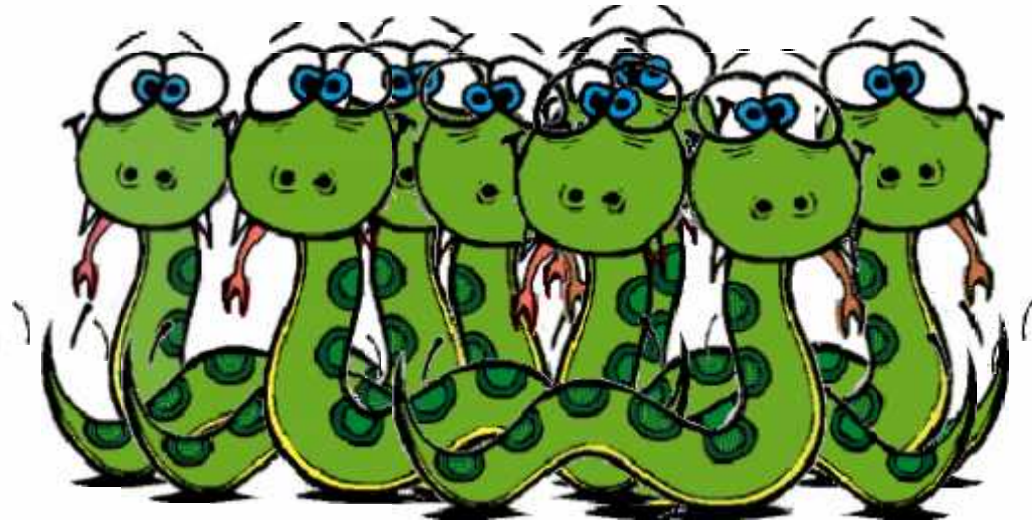
```

Python

```

if __name__ == '__main__':
    with open('result.txt', 'wt') as textfile:
        print("Enter the URL you wish to crawl..")
        myUrl = input("> ")
        for i in re.findall("href=\\\"(.+[\\\"\\']+[\\\"\\'])\\\"",
            urllib.request.urlopen(myUrl).read().decode(), re.I):
            print(i)
            textfile.write(i+'\\n')

```



نگاهی دقیق تر

	پایتون	
حلقه	for x in xrange(1000000): print x	public class ConsoleTest { public static void main(String[] args) { for (int i = 0; i < 1000000; i++) { System.out.println(i); } }}
دیکشنری	for i in xrange(1000): x={} for j in xrange(1000): x[j]=i x[j]	import java.util.Hashtable;public class HashTest { public static void main(String[] args) { for (int i = 0; i < 1000; i++) { Hashtable x = new Hashtable(); for (int j = 0; j < 1000; j++) { x.put(new Integer(i), new Integer(j)); x.get(new Integer(i)); } } }}
کار فایل	f=open('scratch','wb')for i in xrange(1000000): f.write(str(i))f.close()	import java.io.*;public class IOTest{ public static void main(String[] args) { try { File f = new File("scratch"); PrintWriter ps = new PrintWriter(new OutputStreamWriter (new FileOutputStream(f))); for (int i = 0; i < 1000000; i++) { ps.print(String.valueOf(i)); } ps.close(); } catch(IOException ioe) { ioe.printStackTrace(); } }}
لیست	for i in xrange(1000): v=['a','b','c','d','e','f','g'] for j in xrange(1000): v.append(j) v[j]	import java.util.Vector;public class ListTest { public static void main(String[] args) { for (int i = 0; i < 1000; i++) { Vector v = new Vector(); v.addElement("a"); v.addElement("b"); v.addElement("c"); v.addElement("d"); v.addElement("e"); v.addElement("f"); v.addElement("g"); for (int j = 0; j < 1000; j++) { v.addElement(new Integer(j)); v.elementAt(j); } } }}

پایتون vs سی

```
def queensproblem(rows, columns):
    solutions = [[]]
    for row in range(rows):
        solutions = add_one_queen(row, columns, solutions)
    return solutions

def add_one_queen(new_row, columns, prev_solutions):
    return [solution + [new_column]
            for solution in prev_solutions
            for new_column in range(columns)
            if no_conflict(new_row, new_column, solution)]

def no_conflict(new_row, new_column, solution):
    return all(solution[row] != new_column and
               solution[row] + row != new_column + new_row and
               solution[row] - row != new_column - new_row
               for row in range(new_row))

for solution in queensproblem(8, 8):
    print(solution)
```

```
#include <iostream>

using namespace std;

const int N = 8;
int position[N];

// Check if a position is safe
bool isSafe(int queen_number, int row_position)
{
    // Check each queen before this one
    for (int i = 0; i < queen_number; i++)
    {
        // Get another queen's row_position
        int other_row_pos = position[i];

        // Now check if they're in the same row or diagonal
        if (other_row_pos == row_position || // Same row
            other_row_pos == row_position - (queen_number - i) || // Same diagonal
            other_row_pos == row_position + (queen_number - i)) // Same diagonal
            return false;
    }
    return true;
}

// Recursively generates a tuple like [0 3 0 0], then [0 3 0 1] then etc...
void solve(int k)
{
    if (k == N) // We placed N-1 queens (0 included), problem solved!
    {
        // Solution found!
        cout << "Solution: ";
        for (int i = 0; i < N; i++)
            cout << position[i] << " ";
        cout << endl;
    }
    else
    {
        for (int i = 0; i < N; i++) // Generate all possibilities
        {
            // Before putting a queen (the k-th queen) into a row, test it for safeness
            if (isSafe(k, i))
            {
                position[k] = i;
                // Place another queen
                solve(k + 1);
            }
        }
    }
}

int main()
{
    solve(0);
    return 0;
}
```



انتخاب با شما

```
x=input("enter something to calculate: ")
print x
```

```
#include<stdio.h>

void main()
{
    char operation;
    float num1, num2, result;

    printf("Simulation of a Simple Calculator\n");
    printf("XXXXXXXXXXXXXXXXXXXXXXXXXXXX\n");
    printf("Enter two numbers \n");
    scanf("%f %f", &num1, &num2);
    fflush(stdin);
    printf("Enter the operator [+,-,*,/] \n");
    scanf("%c", &operation);
    num1 = (float)num1;
    {
        case '+': result = num1 + num2;
            break;
        case '-': result = num1 - num2;
            break;
        case '*': result = num1 * num2;
            break;
        case '/': result = num1 / num2;
            break;
        default: printf("Invalid operations");
            break;
    }
    printf("\n%.2f %c %.2f = %.2f\n", num1, operation, num2, result);
}
```

```
#include<stdio.h>
#include<math.h>
#include<stdlib.h>

public static void main(String[] args)
{
    System.out.println("Enter two decimal numbers");
    Scanner input = new Scanner(System.in);
    float num1, num2;
    num1 = input.nextFloat();
    num2 = input.nextFloat();
    float result;
    System.out.println("Enter your selection: 1 for Addition, 2 for subtraction, 3 for Multiplication, and 4 for Division");
    int choice;
    choice = input.nextInt();
    switch (choice)
    {
        case 1:
            System.out.println(add(num1, num2));
            break;
        case 2:
            System.out.println(sub(num1, num2));
            break;
        case 3:
            System.out.println(mult(num1, num2));
            break;
        case 4:
            System.out.println(div(num1, num2));
            break;
        default:
            System.out.println("Invalid operation");
    }
}

public static float add(float x, float y)
{
    return x + y;
}

public static float sub(float x, float y)
{
    return x - y;
}

public static float mult(float x, float y)
{
    return x * y;
}

public static float div(float x, float y)
{
    return x / y;
}

return result;
```














MODERN LANGUAGE WARS

PHP VS. PYTHON VS. RUBY

Programmers are particular about their language preferences— and most can tell you exactly why they like one over another. Three languages in particular are the rage these days. We compare Ruby, Python, and PHP to see how they stack up.

SOURCES: MJTSAI.COM | LCGEUROPE.COM | SHOOTOUT.ALOTH.DEBIAN.ORG | GITHUB.COM | C2.COM | RUBY-LANG.COM | PYTHON.ORG | "PHP VS. PYTHON VS. RUBY," KLAUS PURER | MONSTER.COM | LINKEDIN.COM | XODIAN.NET

udemy

	PHP	RUBY	PYTHON
CURRENT VERSION	PHP: 5.3.8 AUGUST 23, 2011	RUBY: 1.9.3 OCTOBER 31, 2011	PYTHON: 3.2.2 SEPTEMBER 4, 2011
PURPOSE	 <p>PHP was designed for web development to produce dynamic web pages.</p>	 <p>Ruby was designed to make programming fun and flexible for the programmer.</p>	 <p>Python was designed to emphasize productivity and code readability.</p>
CREATOR & YEAR RELEASED	1995 RASMUS LERDORF	1995 YUKIHIRO "MATZ" MATSUMOTO	1991 GUIDO VAN ROSSUM
INFLUENCED BY	<ul style="list-style-type: none"> C PERL JAVA C++ TCL 	<ul style="list-style-type: none"> ADA C++ CLU DYLAN EIFFEL LISP PERL PYTHON 	<ul style="list-style-type: none"> ABC ALGOL 68 C C++ ICON JAVA LISP PERL
SITES BUILT USING IT	 WIKIPEDIA  UDEMY  FACEBOOK	 TWITTER  HULU  GROUPON	 YOUTUBE  GOOGLE

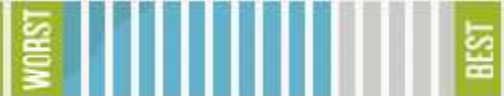
USABILITY



PHP follows a classic approach and is extensively documented.



Programmers describe Ruby code as elegant, powerful, and expressive. It is highly usable because of its principle of least astonishment, enforced to minimize confusion for users.



Python uses strict indentation enforcements. Python is arguably the most readable programming language.

EASE OF LEARNING



PHP is easy to learn for former C programmers.



Ruby is better for a programmer who already knows a language or two.



Python is great for beginners, often recommended by programmers due to the simplicity of its syntax.

BEST FOR

Which language would be best for the small business owner or job seeker?



SMALL BUSINESS

Based on the LinkedIn findings above, those businesses looking for developers will have the easiest time finding a PHP expert, followed by Python, then Ruby.

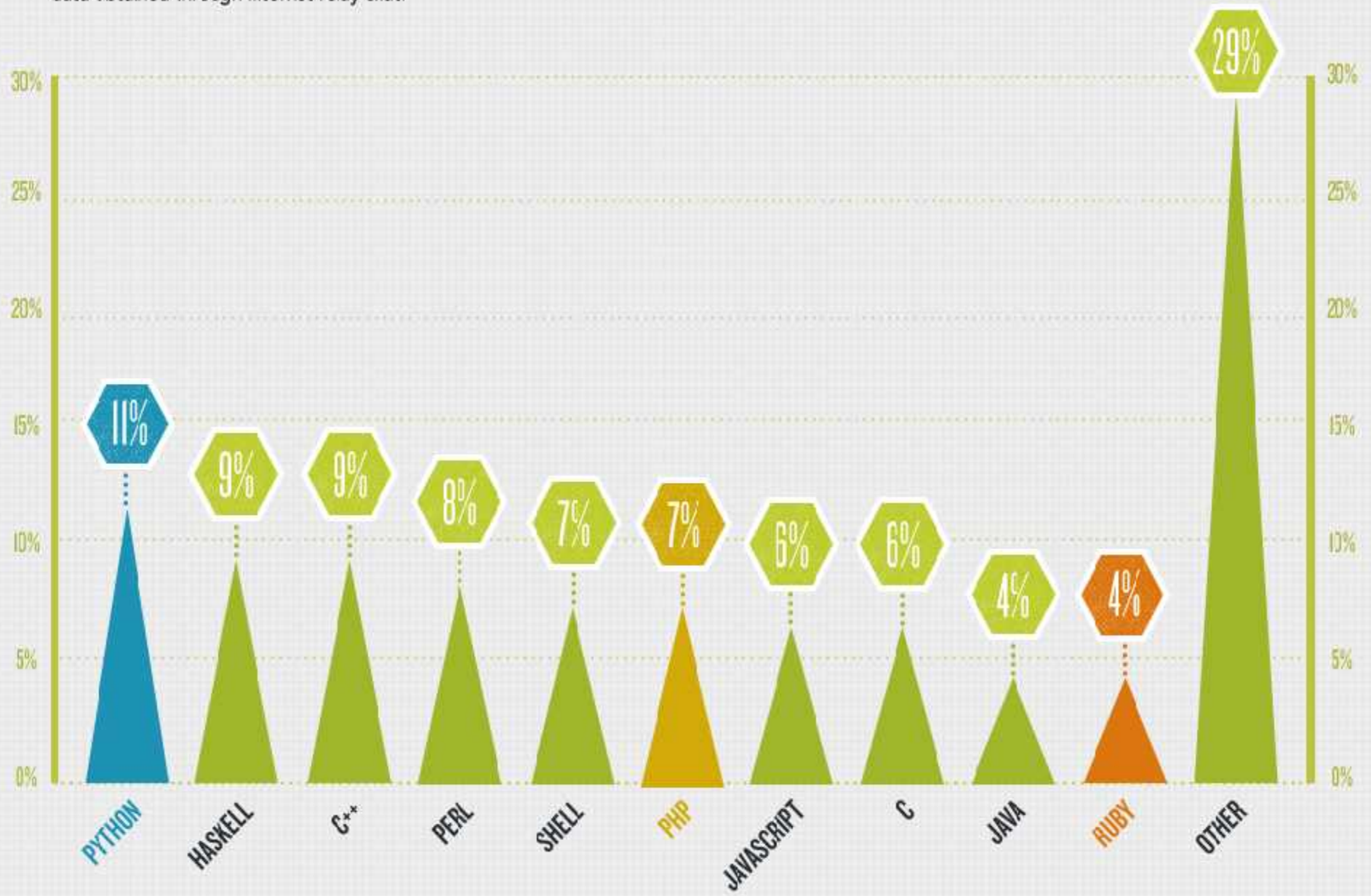


THE JOB SEEKER

Of all three languages, PHP is the most pervasive in the programming world followed by Python. Those two languages would be best to know if you are seeking a job.

MOST-DISCUSSED LANGUAGE

The popularity index below comes from IEEE Spectrum, which used data obtained through internet relay chat.





HOW FAST IS IT?

Using benchmark tests, we compare which languages are the fastest in terms of lines of code and average run time.

LINES OF CODE:

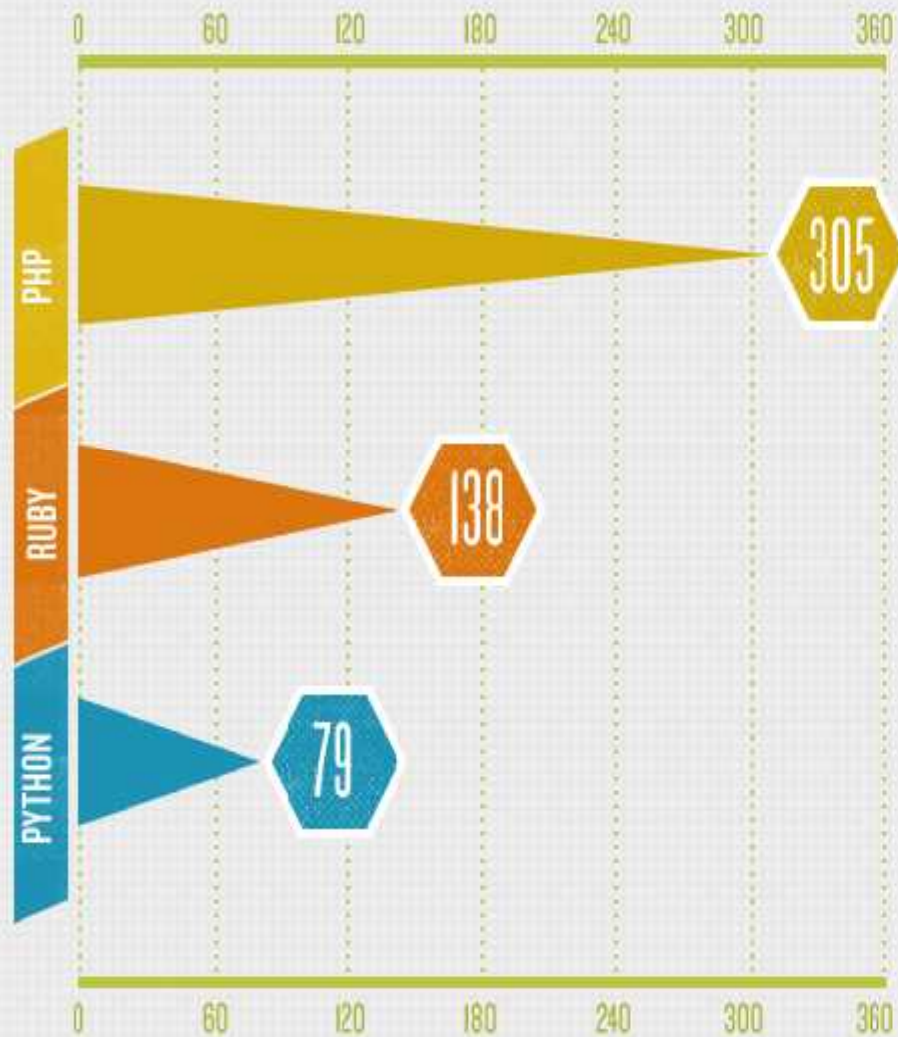


AVERAGE RUN TIME:



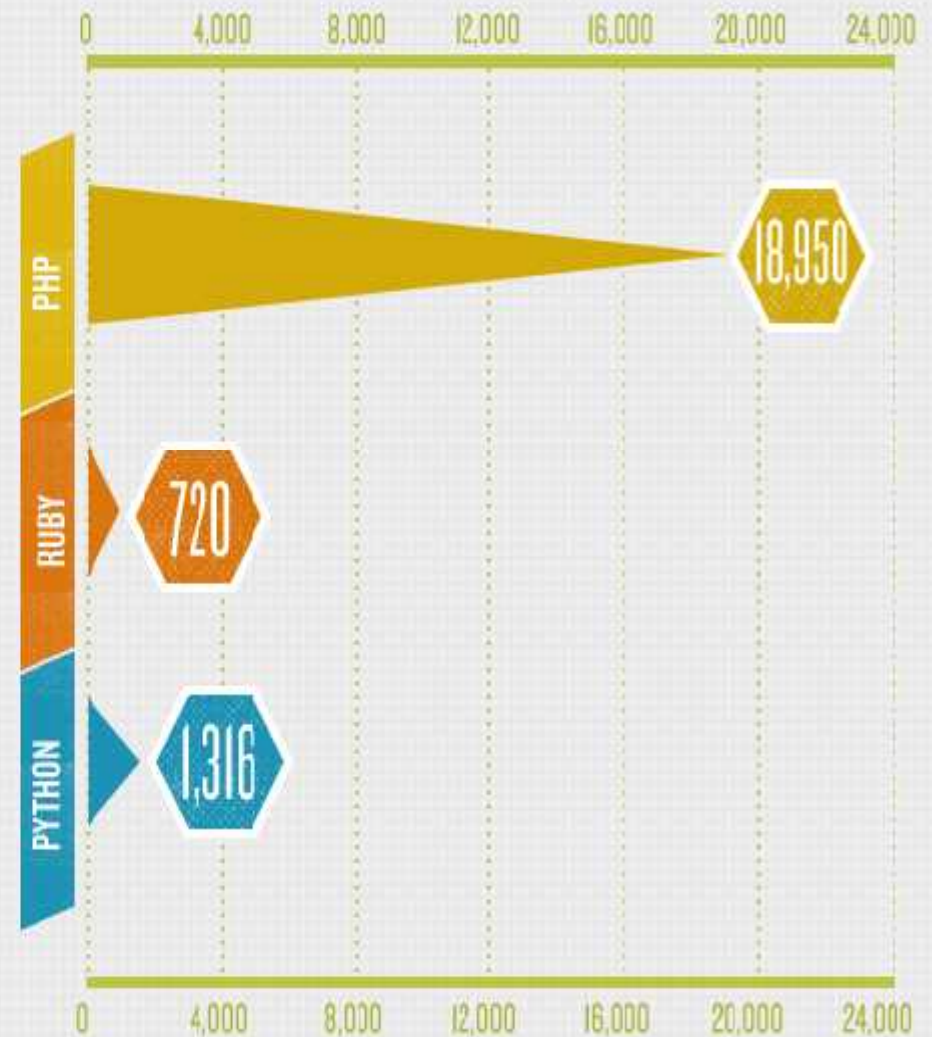
JOB POSTINGS ON MONSTER.COM

Below are the number of job postings as of December 12, 2011.



NUMBER OF DEVELOPERS

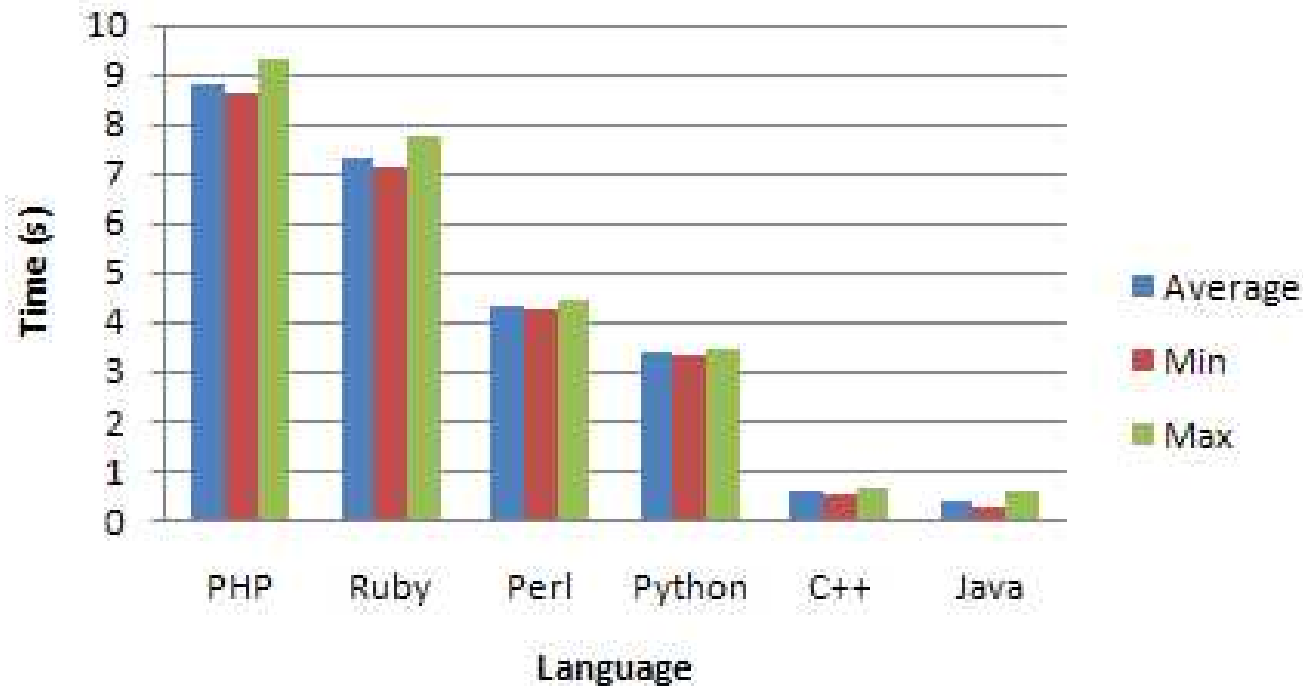
Here are the number of developers on LinkedIn as of December 15, 2011.



*Determined by searching for professionals with "Ruby developer," "Python developer," and "PHP developer."

سرعت کم

Mergesort Performance



- پایتون زبانی پویاست.
- پایتون یک زبان مفسری است.

شکل برنامه ی پایتونی

```
Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Enter an integer from 1 to 99: 50

guess is high
Enter an integer from 1 to 99: 25

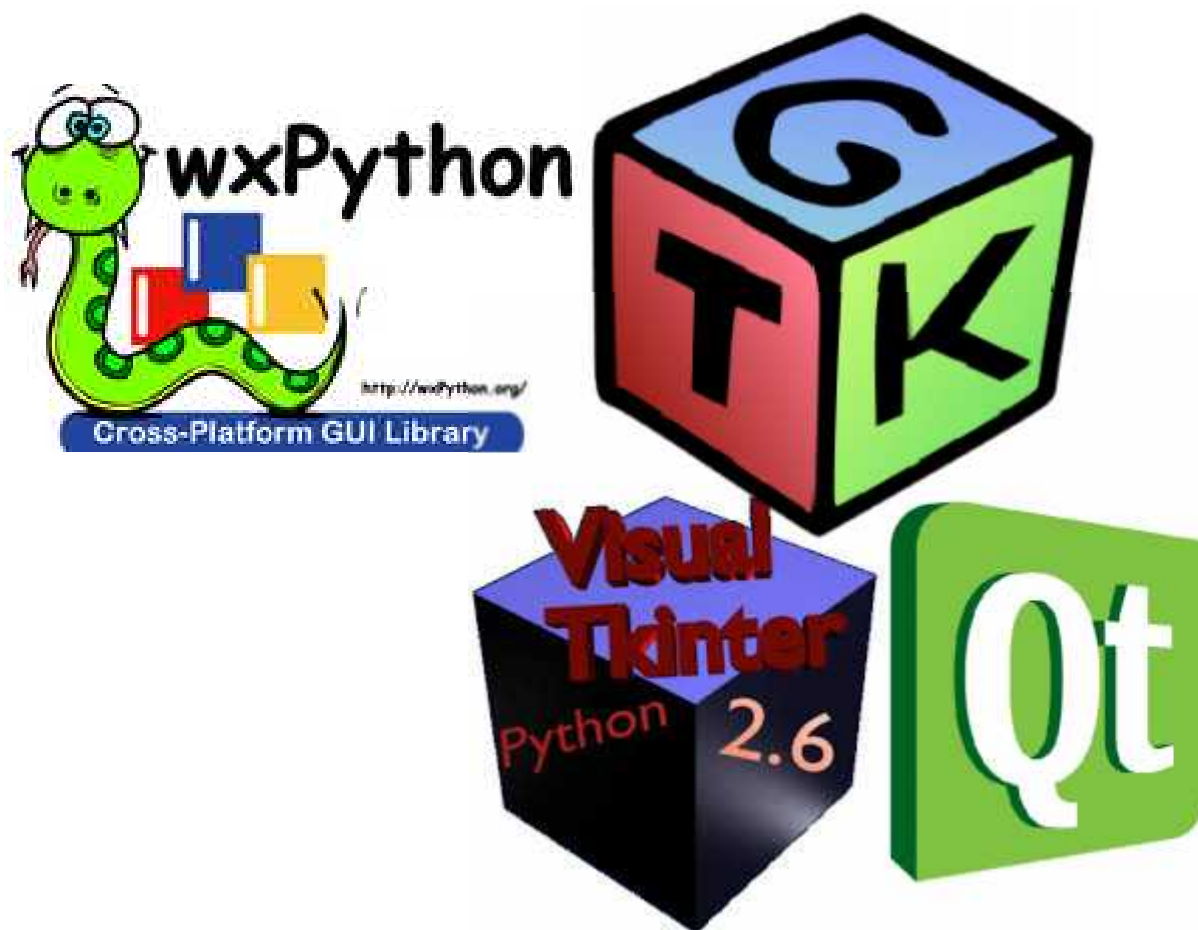
guess is high
Enter an integer from 1 to 99: 12

guess is low
Enter an integer from 1 to 99: 17

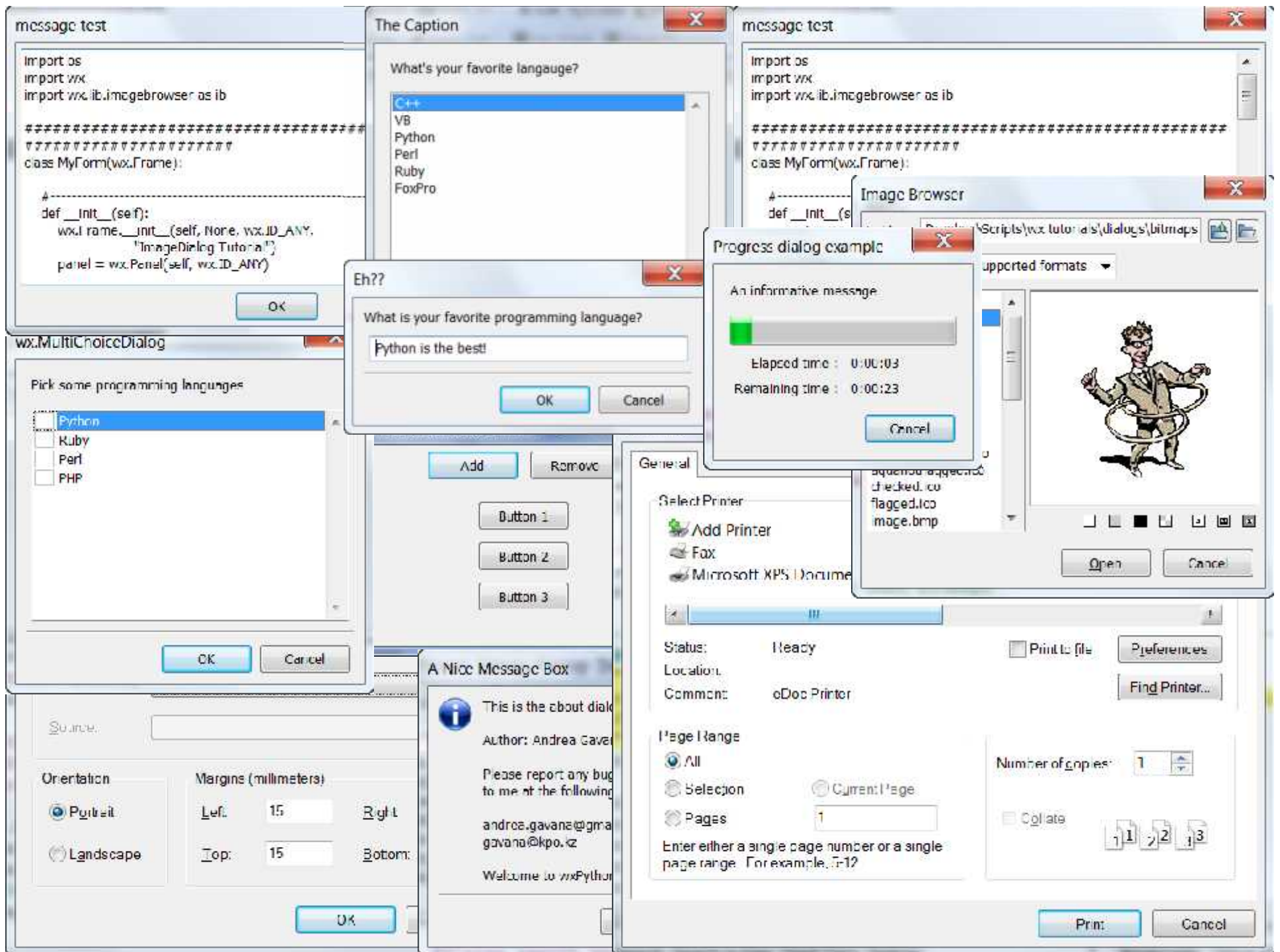
you guessed it!
```

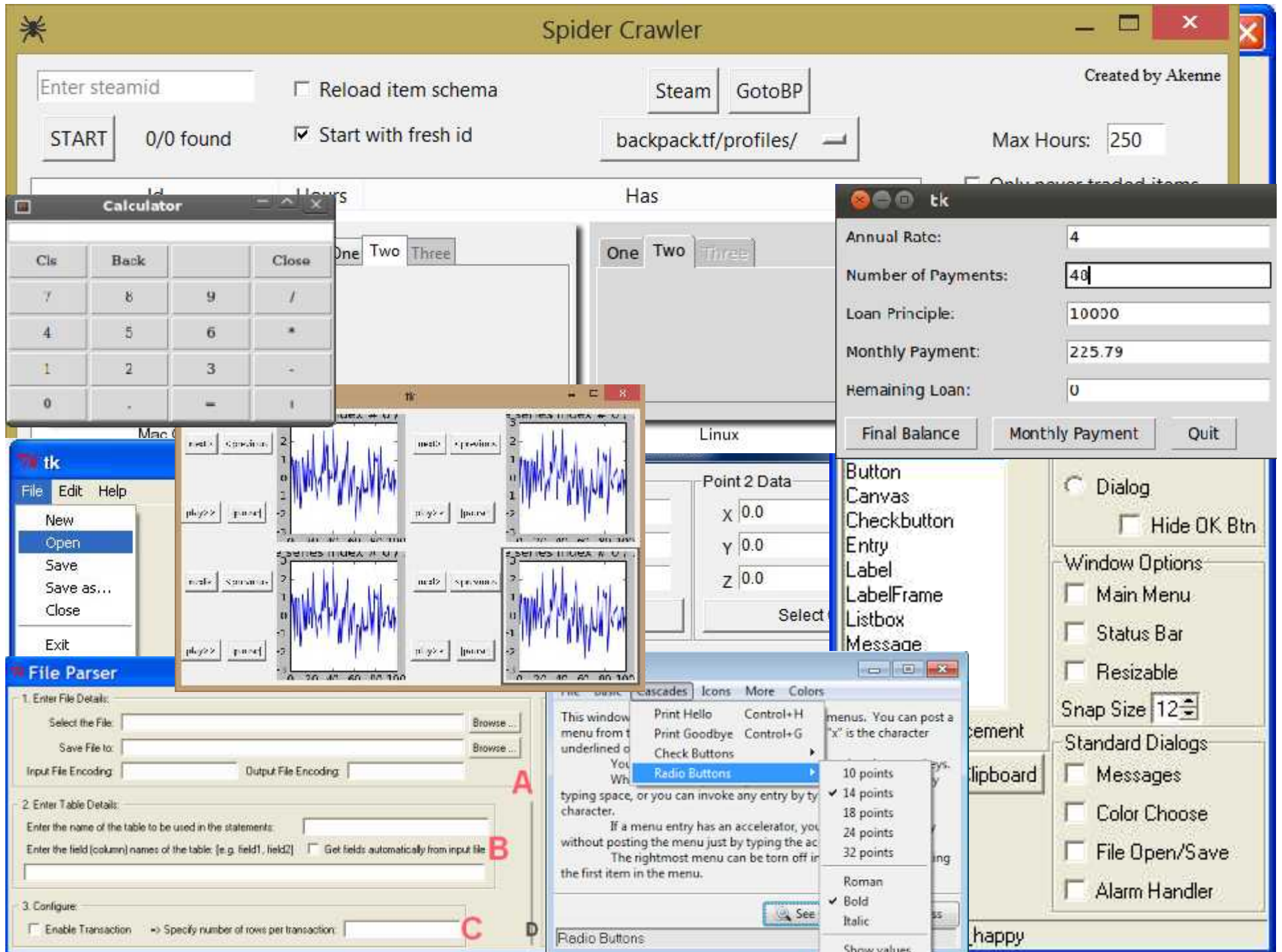


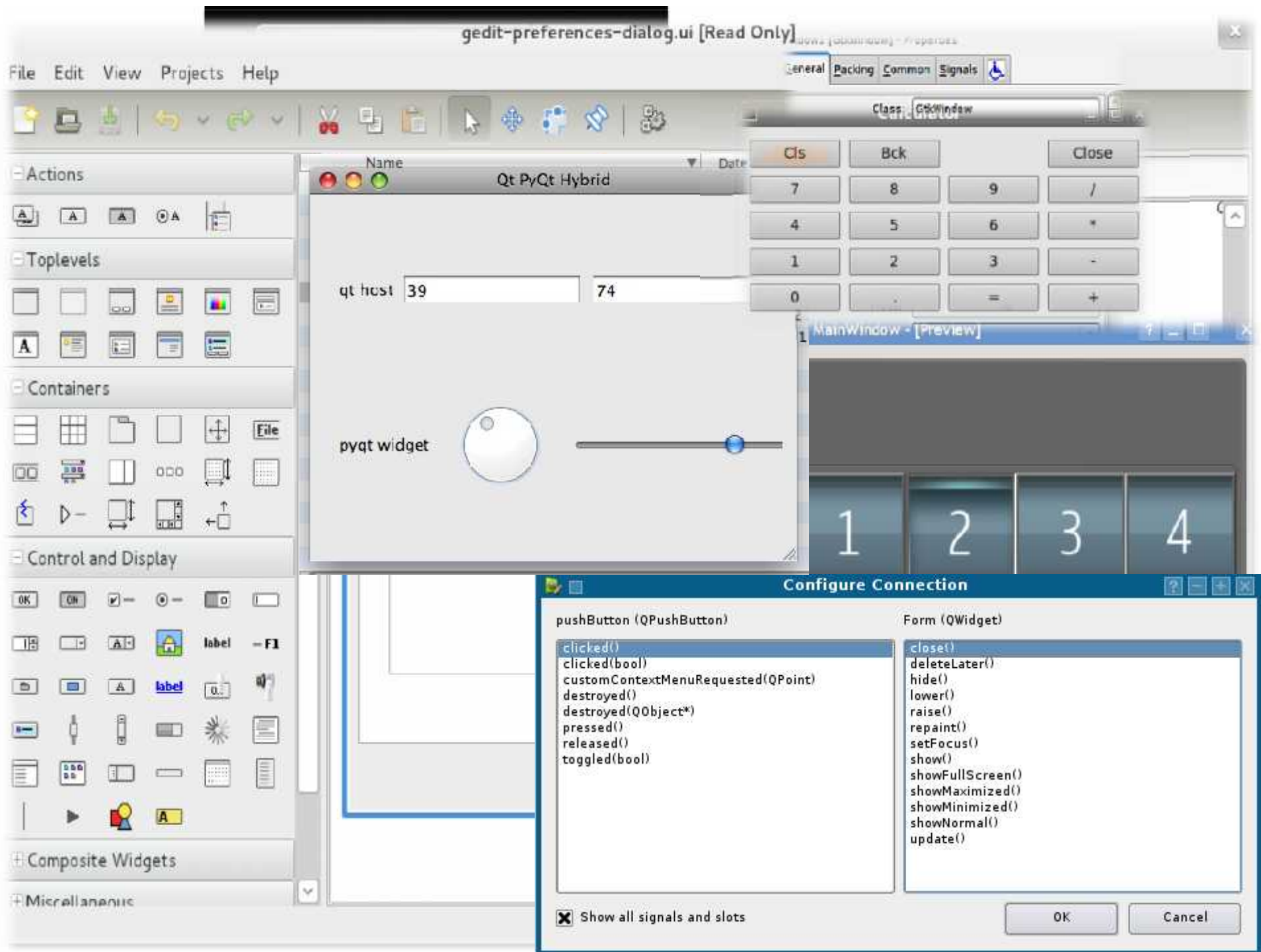
گرافیک



Code less.
Create more.
Deploy everywhere.







GNOME Shell Extension Preferences

Extension SettingsCenter

Global

Menu Label

Settings Center

Apply

Replace System Settings (Only if found)

☐ OFF

Menu Items

Advanced Settings

☒ ON ☐ Up Down Del

Desktop Config Editor

☒ ON ☐ Up Down Del

Gnome Config Editor

☒ ON ☐ Up Down Del

Session Properties

☒ ON ☐ Up Down Del

Extensions Preferences

☒ ON ☐ Up Down Del

Passwords and Keys

☐ OFF ☐ Up Down Del

NVidia Settings

☐ OFF ☐ Up Down Del

Add Menu

Label

Command

Add



Alex Gartner

Available

Notifications

☒ ON ☐

System Settings

Settings Center

Advanced Settings

Desktop Config Editor

Gnome Config Editor

Session Properties

Extensions Preferences

Log Out

Lock

Suspend

Power Off



R**Y****K****A** | **CBT**