The Art of Programming

Vihar Kurama
Co-founder, Caravel.AI

The Story!

Vihar Kurama, 2018

What is Programming?

Vihar Kurama, 2018

A **programming language** is a formal language that specifies a set of instructions that can be used to produce various kinds of output.

Vihar Kurama, 2018

Why?

Vihar Kurama, 2018

Programming is the way that tells a computer what the user (you) want it to do.

Vihar Kurama, 2018

“I think everybody in this country should learn how to program a computer because it teaches you how to think”

Steve Jobs

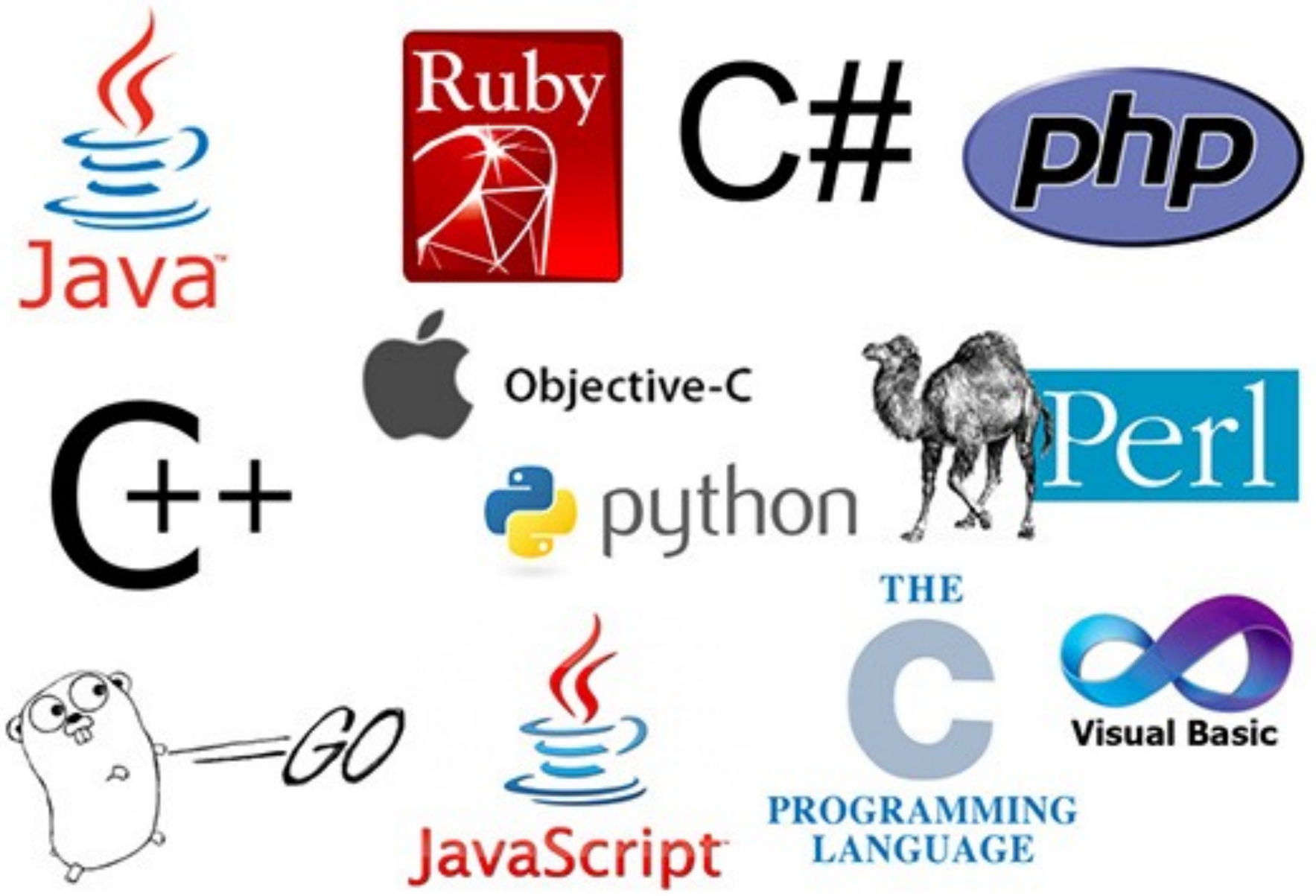
Things that you can do with programming?

Vihar Kurama, 2018

- Communication between Computers
- Operating Systems
- Solving Problems
- Implementing Algorithms
- Heavy Computation
- Building Web Applications
- Data Science
- Machine Learning
- Artificial Intelligence

The Programming Language!

Vihar Kurama, 2018



Vihar Kurama, 2018

Choosing the Right Programming Tool

Vihar Kurama, 2018

Use Case, Portability, Efficient, Secure, Comfortable, Community Support.

Vihar Kurama, 2018

Examples!

Vihar Kurama, 2018

Python and Why?

Vihar Kurama, 2018

- Easy to Learn
- Dynamically Interpreted Language
- Less Code is the best code
- Object Oriented
- Open Source
- Easy to Debug
- Vast Community Support
- Thousands Of Libraries

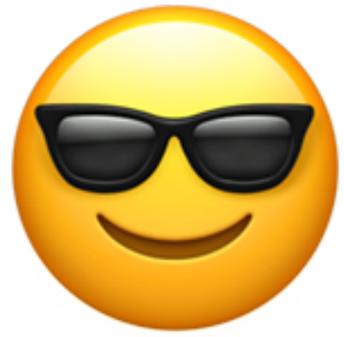
OpenSourced [@]

github.com/python

Vihar Kurama, 2018

Things you can do with Python.

Vihar Kurama, 2018

Infinite 

Vihar Kurama, 2018

Mostly Used For

- Understanding Programming
- Problem Solving
- Web Scraping
- Building Softwares
- Building Web Applications 🌐
- Data Science 📊
- Data Visualisation 📈
- Machine Learning 🎰
- Artificial Intelligence 🚗

Python In Action

Vihar Kurama, 2018

Python for Problem Solving

Vihar Kurama, 2018

How many lines do you think will it take to write a simple sorting algorithm in Python compared to C++ ?

Vihar Kurama, 2018

C++ - 103 Lines of Code

Python - 25 Lines of Code

Python reduces Approx. 75%

Vihar Kurama, 2018

Python

```
def mergeSort(alist):
    print("Splitting ",alist)
    if len(alist)>1:
        mid = len(alist)//2
        lefthalf = alist[:mid]
        righthalf = alist[mid:]

        mergeSort(lefthalf)
        mergeSort(righthalf)

        i=0
        j=0
        k=0
        while i < len(lefthalf) and j < len(righthalf):
            if lefthalf[i] < righthalf[j]:
                alist[k]=lefthalf[i]
                i=i+1
            else:
                alist[k]=righthalf[j]
                j=j+1
            k=k+1

        while i < len(lefthalf):
            alist[k]=lefthalf[i]
            i=i+1
            k=k+1

        while j < len(righthalf):
            alist[k]=righthalf[j]
            j=j+1
            k=k+1
    print("Merging ",alist)
```

C++

```
/* C program for Merge Sort */
#include<stdlib.h>
#include<stdio.h>

// Merges two subarrays of arr[].
// First subarray is arr[l..m]
// Second subarray is arr[m+1..r]
void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 =  r - m;

    /* create temp arrays */
    int L[n1], R[n2];

    /* Copy data to temp arrays L[] and R[] */
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1+ j];

    /* Merge the temp arrays back into arr[l..r]*/
    i = 0; // Initial index of first subarray
    j = 0; // Initial index of second subarray
    k = l; // Initial index of merged subarray
    while (i < n1 && j < n2)
    {
        if (L[i] <= R[j])
        {
            arr[k] = L[i];
            i++;
        }
        else
        {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    /* Copy the remaining elements of L[], if there
       are any */
    while (i < n1)
    {
        arr[k] = L[i];
        i++;
        k++;
    }

    /* Copy the remaining elements of R[], if there
       are any */
    while (j < n2)
    {
        arr[k] = R[j];
        j++;
        k++;
    }
}
```

```
/* l is for left index and r is right index of the
   sub-array of arr to be sorted */
void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        // Same as (l+r)/2, but avoids overflow for
        // large l and h
        int m = l+(r-l)/2;

        // Sort first and second halves
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
}

/* UTILITY FUNCTIONS */
/* Function to print an array */
void printArray(int A[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", A[i]);
    printf("\n");
}

/* Driver program to test above functions */
int main()
{
    int arr[] = {12, 11, 13, 5, 6, 7};
    int arr_size = sizeof(arr)/sizeof(arr[0]);

    printf("Given array is \n");
    printArray(arr, arr_size);

    mergeSort(arr, 0, arr_size - 1);

    printf("\nSorted array is \n");
    printArray(arr, arr_size);
    return 0;
}
```

Vihar Kurama, 2018

It’s more simple and easy.

Vihar Kurama, 2018

Still Not Convinced

Don’t worry we got the sort function!

Vihar Kurama, 2018

```
$python
>>> a = [9,4,2,5,6,7]
>>> a.sort()
>>> a
 [2, 4, 5, 6, 7, 9]
```

Next Topics

Python For Web
Python For DataScience
Python for Machine Learning

Vihar Kurama, 2018

Late Enough!

Let’s Get Started with Python.

Vihar Kurama, 2018