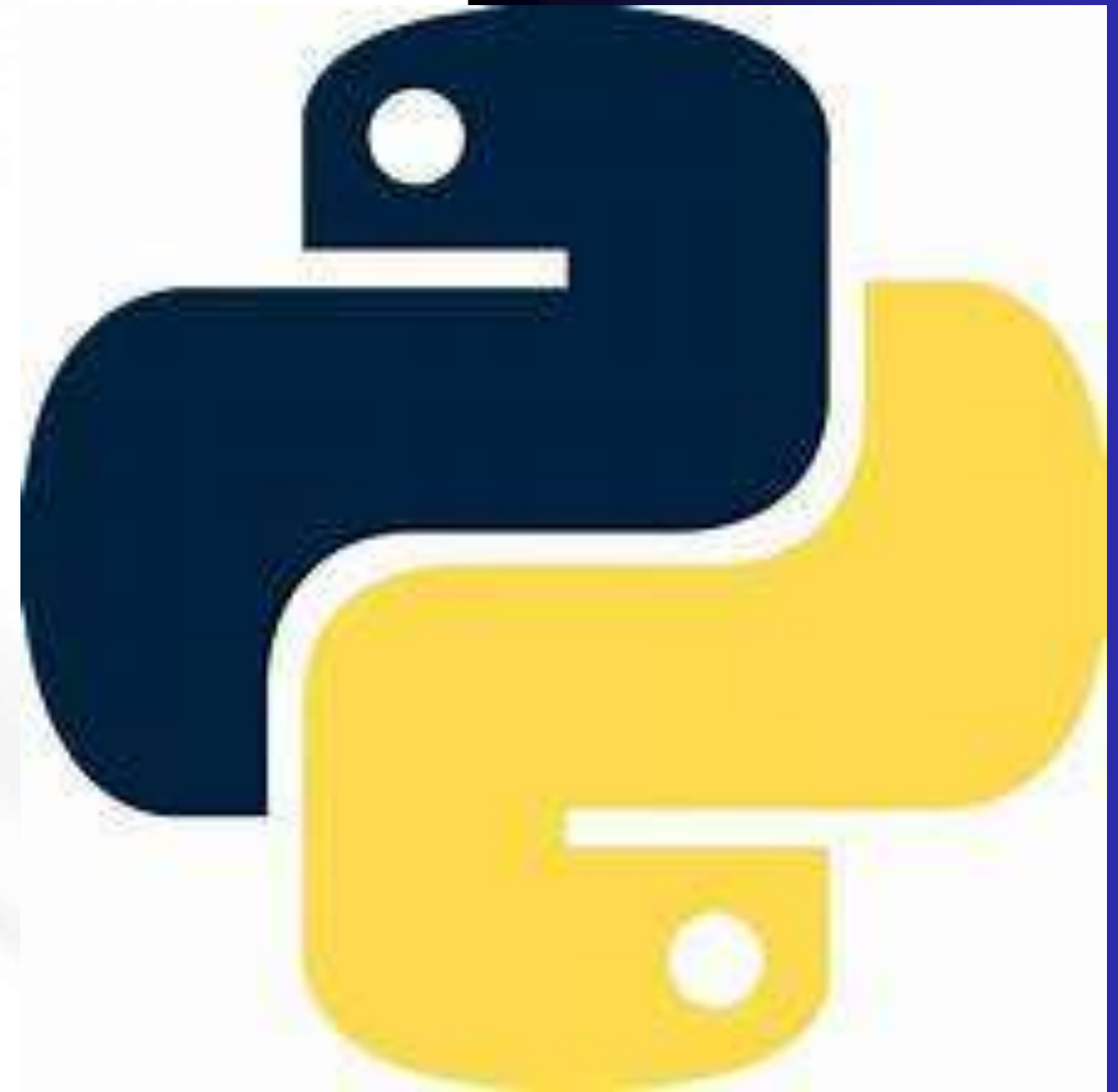


# PYTHON



# WHAT IS PYTHON ..?

**Python is a high-level general purpose programming language.**

**Guido Van Rossum invented Python in 1991.**

**He named it Python because he was highly inspired by a show “Monthly Python’s Flying Circus”.**



**1. General Purpose**

**2. High level**

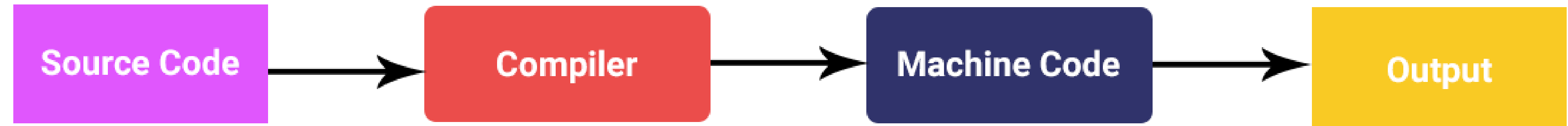
**3. Interpreted**

**4. Dynamically Typed**

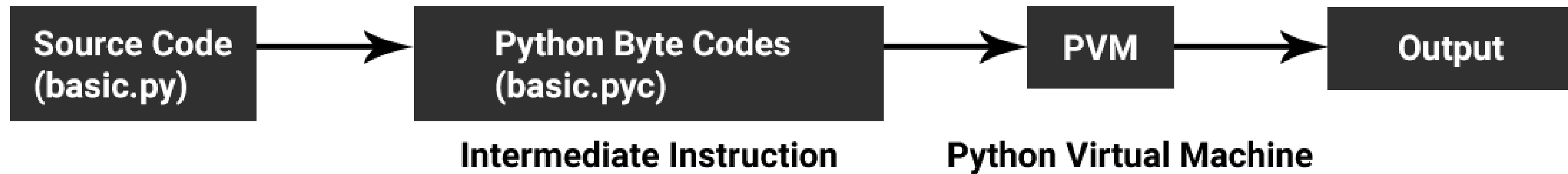
**5. Multi-paradigm**



# Compiler Works



# Interpreter Works

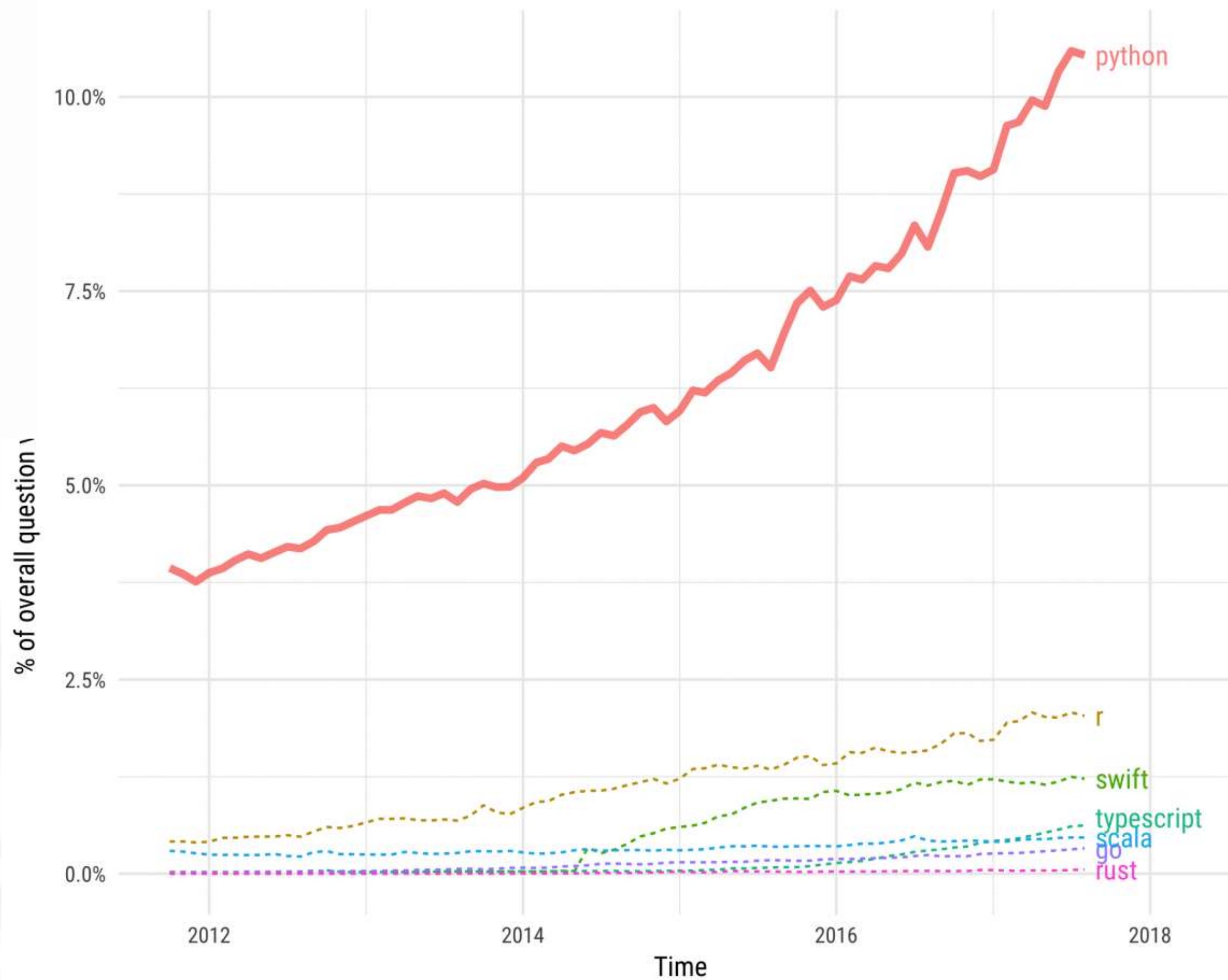






# Python compared to smaller, growing technologies

Based on question traffic in World Bank high-income countries



# Top Companies using Python

**NOKIA**



**amazon**



  
**reddit**



**YAHOO!**  
Maps

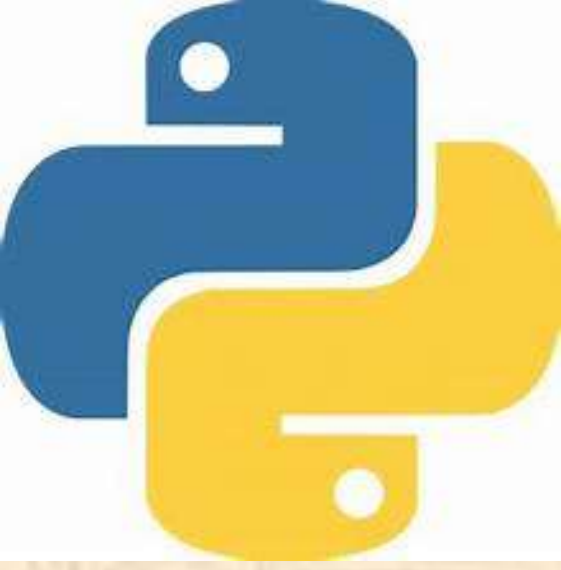


**IBM**

**Quora**

WALT DISNEY  
**FEATURE  
ANIMATION**





# Applications of Python

**Games and  
3D Graphics**



**Software  
Development**



**Database  
Access**



**Web  
Development**

**Business  
Applications**



**Network  
Programming**





## GAME DEVELOPMENT



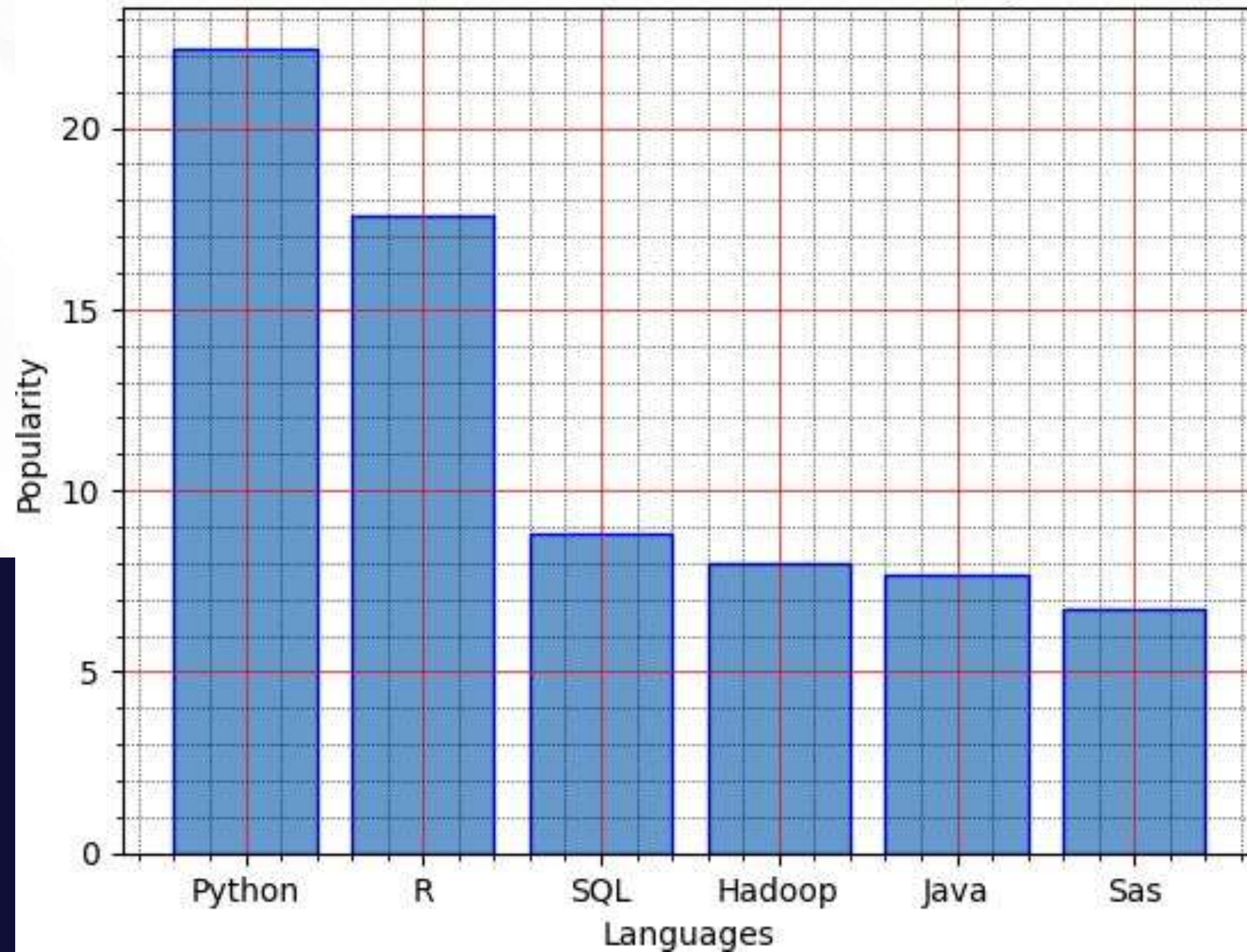


SKETCH  
PROGRAM  
APPLICATION  
MOBILE  
PAGE  
PEOPLE  
LAPTOP  
DATA  
TECHNOLOGY  
NOTEBOOK  
ONLINE  
OFFICE  
DESIGN  
DESIGNER  
SOFTWARE  
DIGITAL  
VISION  
SEO  
INFORMATION  
RESPONSIVE  
MEDIA  
DEVICE  
COMMUNICATION  
HTML  
WORK  
WWW  
CONTENT  
DEVELOPER  
DISPLAY  
CODE  
SCREEN  
CODING  
STRATEGY

**COMPUTER**  
**WEB**  
**DEVELOPMENT**  
**BUSINESS**  
**INTERNET**  
**PROGRAMMING**  
**WEBSITE**



The Six Most Common Data Science Skills in Job Posting





ABOUT

# KEYWORDS IN PYTHON

1

as the  
names  
of

2

Keywords are some predefined and reserved words in Python

function

s

classes

# code

import keyword

print(keyword.kwlist)

# IDENTIFIERS IN PYTHON

**IDENTIFIER IS A USER-DEFINED NAME GIVEN TO A VARIABLE, FUNCTION, CLASS, MODULE, ETC.**

**THE IDENTIFIER IS A COMBINATION OF CHARACTER DIGITS AND AN UNDERSCORE.**

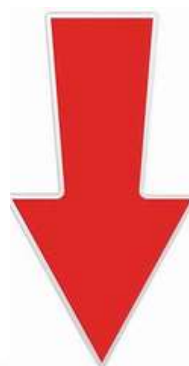
**THEY ARE CASE-SENSITIVE**

**01**



It cannot be a reserved python keyword.

**02**



It should not contain white space.

**03**



It can be a combination of A-Z, a-z, 0-9, or underscore.

**04**



It should not contain any special character other than an underscore ( \_ ).



# Operators in Python

>> << != & + ÷  
\* % // >= - ×





# Python Operators

**Arithmetic Operators**



+, -

**Relational Operators**



>, <

**Assignment Operators**



=, +=

**Logical Operators**



and, or

in,  
not in



**Membership Operators**

is, is not

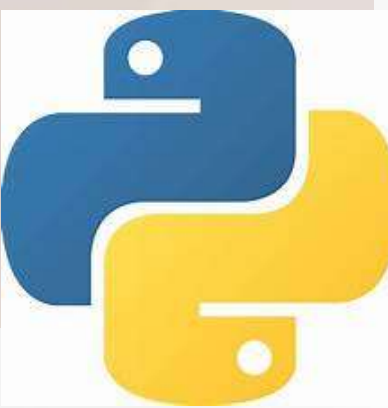


**Identity Operators**

&, ^



**Bitwise Operators**







# **PYTHON ARITHMETIC OPERATORS**



Operators	Meaning	Example	Result
+	Addition	$4 + 2$	6
-	Subtraction	$4 - 2$	2
*	Multiplication	$4 * 2$	8
/	Division	$4 / 2$	2
%	Modulus operator to get remainder in integer division	$5 \% 2$	1
**	Exponent	$5 ** 2 = 5^2$	25
//	Integer Division/ Floor Division	$5 // 2$ $-5 // 2$	2 -3



# Assignment Operators

Assignment operator	Sample expression	Explanation	Assigns
Assume : int c=3, d=5,e=4,f=6,g=12			
+=	c+=7	c=c+7	10 to c
-=	d-=4	d=d-4	1 to d
*=	e*=5	e=e*5	20 to e
/=	f/=3	f=f/3	2 to f
%=	g%=9	g=g%9	3 to g

# Relational Operators

Operators	Meaning	Example	Result
<	Less than	$5 < 2$	False
>	Greater than	$5 > 2$	True
<=	Less than or equal to	$5 <= 2$	False
>=	Greater than or equal to	$5 >= 2$	True
==	Equal to	$5 == 2$	False
!=	Not equal to	$5 != 2$	True



# Logical operators

```
>>> a, b, c = 10, 20, 30
```

```
>>> (a > b) and (b < c)
False
```

```
>>> (a < b) and (b < c)
True
```

```
>>> (a > b) or (b < c)
True
```

Operator	Description
a and b	Logical AND If both operands are True then it returns True
a or b	Logical OR If one of the operands is True then it returns True
not	Logical NOT





## Python Identity Operators :

Identity Operators are used to check the address reference of two variable is same or not.

Operator	Description
is	It is evaluated to be true if the reference present at both sides points to the same object. a = 10 , b = 10 , a is b returns true.
is not	It is evaluated to be true if the reference present at both side do not point to the same object. a = 10 , b = 20 a is not b returns true

# Membership Operators

Sign	Name	Syntax	Explanation
in	-	A in B	Gives true if value is in string
not in	-	A not in B	Gives true if value is in string



## Types of Bitwise Operators

Operator	Name	Example	Result
&	Bitwise AND	6 & 3	2
	Bitwise OR	10   10	10
^	Bitwise XOR	2 ^ 2	0
~	Bitwise 1's complement	~9	-10
<<	Left-Shift	10 << 2	40
>>	Right-Shift	10 >> 2	2

# DATA-TYPES IN PYTHON

```
graph TD; A[DATA-TYPES IN PYTHON] --> B[NUMERIC]; A --> C[BOOLEAN]; A --> D[DATA COLLECTION]; B --> E[INTEGER]; B --> F[FLOAT]; B --> G[COMPLEX]; D --> H[LISTS]; D --> I[TUPLES]; D --> J[DICTIONARIES]; D --> K[SETS]; D --> L[STRINGS];
```

## NUMERIC

INTEGER

FLOAT

COMPLEX

## BOOLEAN

## DATA COLLECTION

LISTS

TUPLES

DICTIONARIES

SETS

STRINGS