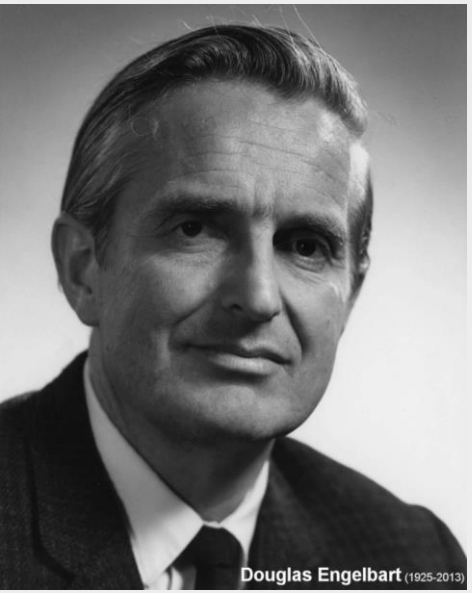
دوره پایتون

هادی حیدری – امیررضا بهجت

پاییز ۱۴۰۰

از لندن تا سیلیکون ولی:



Doug Engelbart

برای دیدن مادر همه نمایش‌ها برروی عکس مقابل کلیک کنید

Ada Lovelace

History of Programming Languages

1843: Ada Lovelace's machine algorithm

1944-45: Plankalkül

1949: Assembly Language

1949: Shortcode

1952: Autocode

1957: FORTRAN

1958: ALGOL (Algorithmic Language)

1958: LISP (List Processor)

1959: COBOL

1964: BASIC

1970: PASCAL

1972: Smalltalk

1972: C

1972: SQL (SEQUEL at the time)

1980/81: Ada

1983: C++

1983: Objective-C

1987: Perl

1990: Haskell

1991: Python

1991: Visual Basic

1993: Ruby

1995: Java

1995: PHP

1995: JavaScript

2000: C#

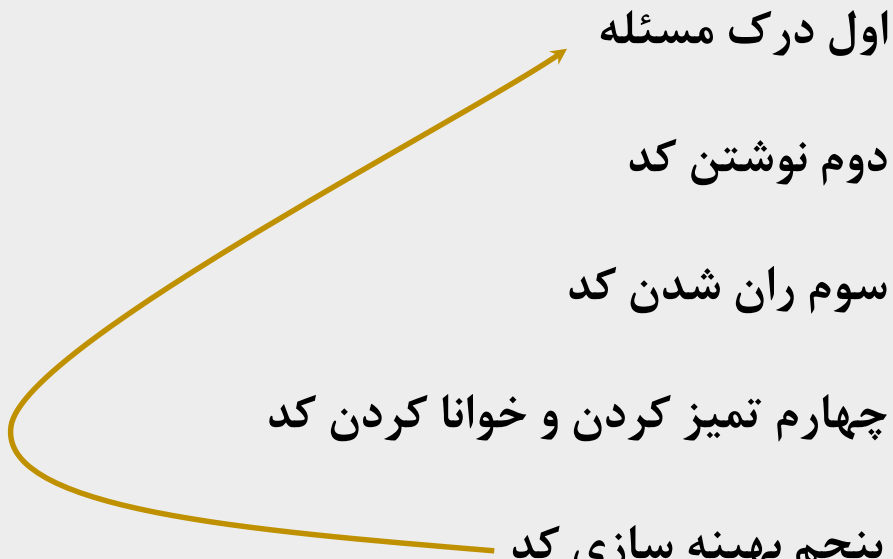
2003: Scala

2003: Groovy

2009: Go

2014: Swift

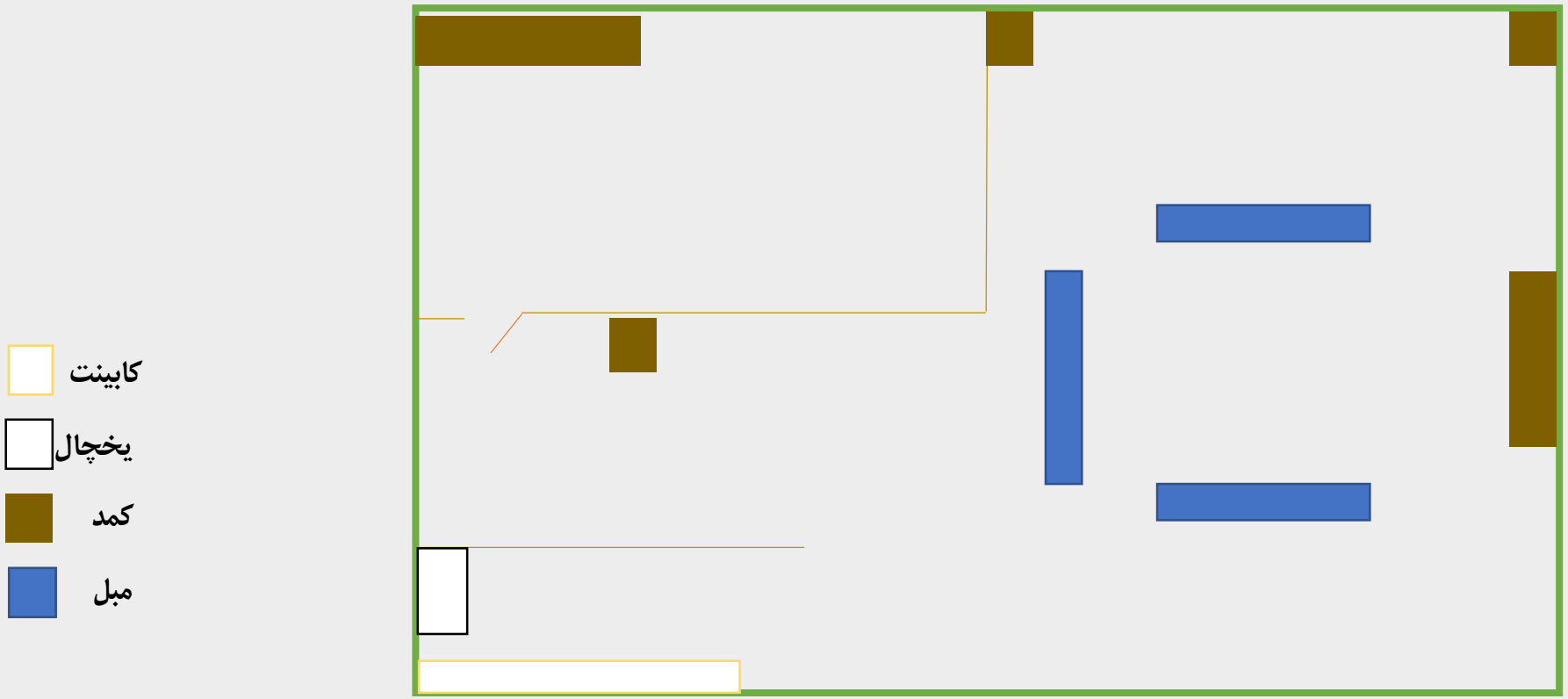
مراحل برنامه نویسی:

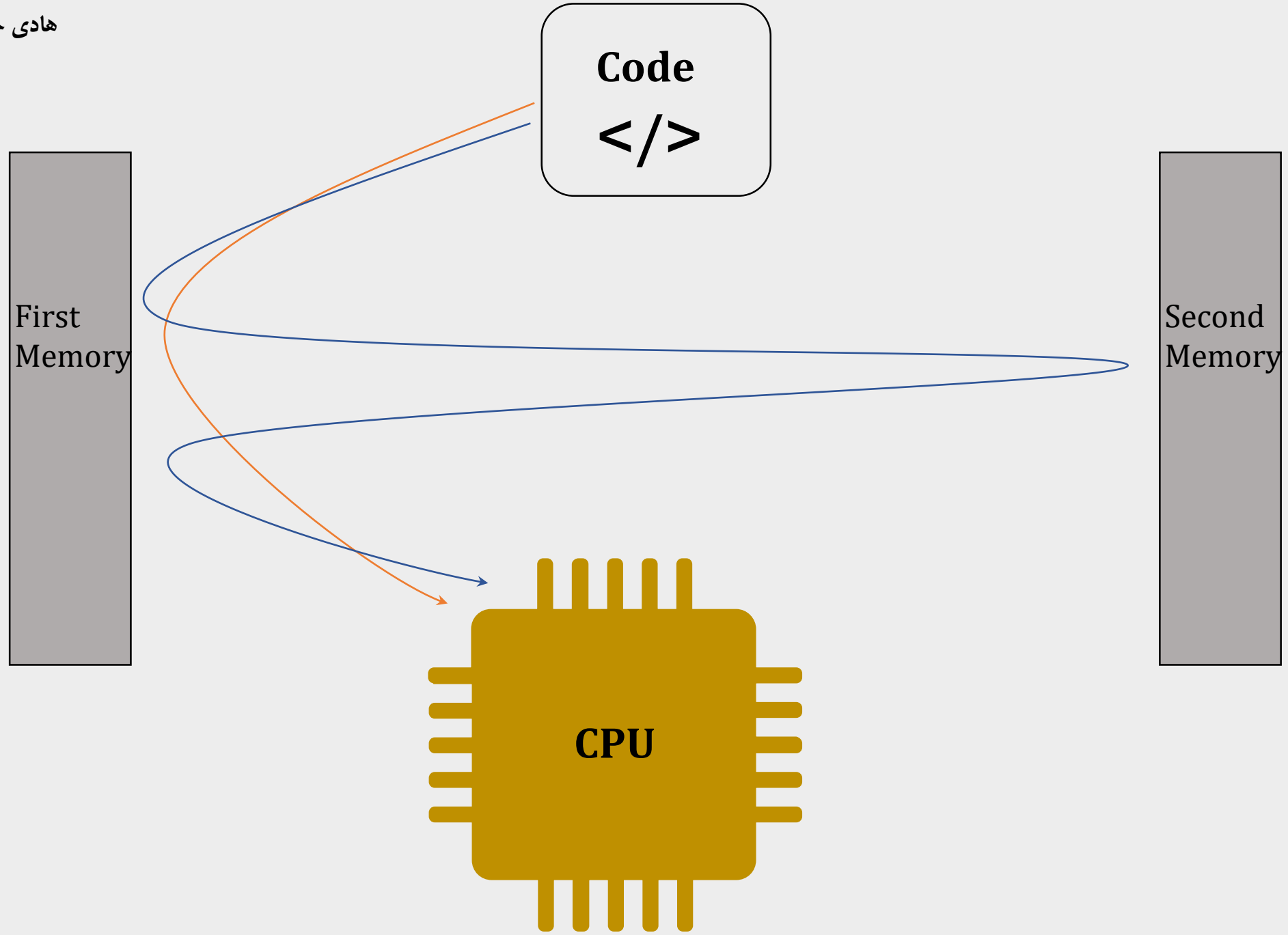


$$f(x|\mu\ ,\sigma^2) = \frac{1}{\sqrt{2\ \pi\ \sigma^2}}\ e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

فهم برنامه نویسی:

فرض کنید یک دوست با معرفت ولی احمق دارید. دوست شما تعداد محدودی کلمه بلد است و تقریبا قبل از گفتن هر کلمه‌ای باید آن را به دوستتان معرفی کنید.
حالا فرض کنید داخل خانه ای با دوستتان هستید و می‌خواهید به او بگویید برای شما سیب از یخچال بیاورد.




Code
</>
First
Memory
Second
Memory
CPU

Python is ……..

Open Source and Free — پایتون زبانی متن باز و رایگان است، هرچند که می‌توان با پایتون برنامه متن بسته(**Close Source**) نیز تولید کرد

Dynamic — پویا بودن به معنی این است که نیازی به تعریف نوع(**Type**) متغیر(**Variables**) نیست و در زمان اجرا(**Run Time**) متغیر مشخص می‌شود

General Purpose — پایتون چند منظوره یعنی در حوزه‌های مختلفی می‌توان از آن استفاده کرد

Multi-Paradigm — هرچند که در پایتون هر چیزی در پایتون یک شی(**Object**) است، اما می‌توان در پایتون از پارادایم‌های مختلف برنامه نویسی استفاده کرد

High-Level — سطح بالا بودن یک زبان اصولا به سادگی و راحتی کد نویسی در آن بر می‌گردد که پایتون در این زمینه بی نظیر است

Python Data Types

انواع داده	دسته بندی
str	Text Type | نوع متنی
Int // float // complex	Numeric Types | انواع عددی
list // tuple // range	Sequence Types | انواع دنباله
dict	Mapping Type | نوع نگاشتی
Set // frozenset	Set Types | انواع مجموعه
bool	Boolean Type | انواع بولی
bytes // bytearray // memoryview	Binary Types | انواع باینری

Python Arithmetic Operators

عملگرهای محاسباتی پایتون

Arithmetic operators are used with numeric values to perform common mathematical operations:

Operator	Name	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

Python Assignment Operators

عملگر های تخصیص پایتون

Assignment operators are used to assign values to variables:

Operator	Example	Same As
=	x = 5	x = 5
+=	x += 3	x = x + 3
-=	x -= 3	x = x - 3
*=	x *= 3	x = x * 3
/=	x /= 3	x = x / 3
%=	x %= 3	x = x % 3
//=	x //= 3	x = x // 3
**=	x **= 3	x = x ** 3
&=	x &= 3	x = x & 3
|=	x |= 3	x = x | 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

Python Comparison Operators

عملگر های مقایسه و سنجش پایتون

Comparison operators are used to compare two values:

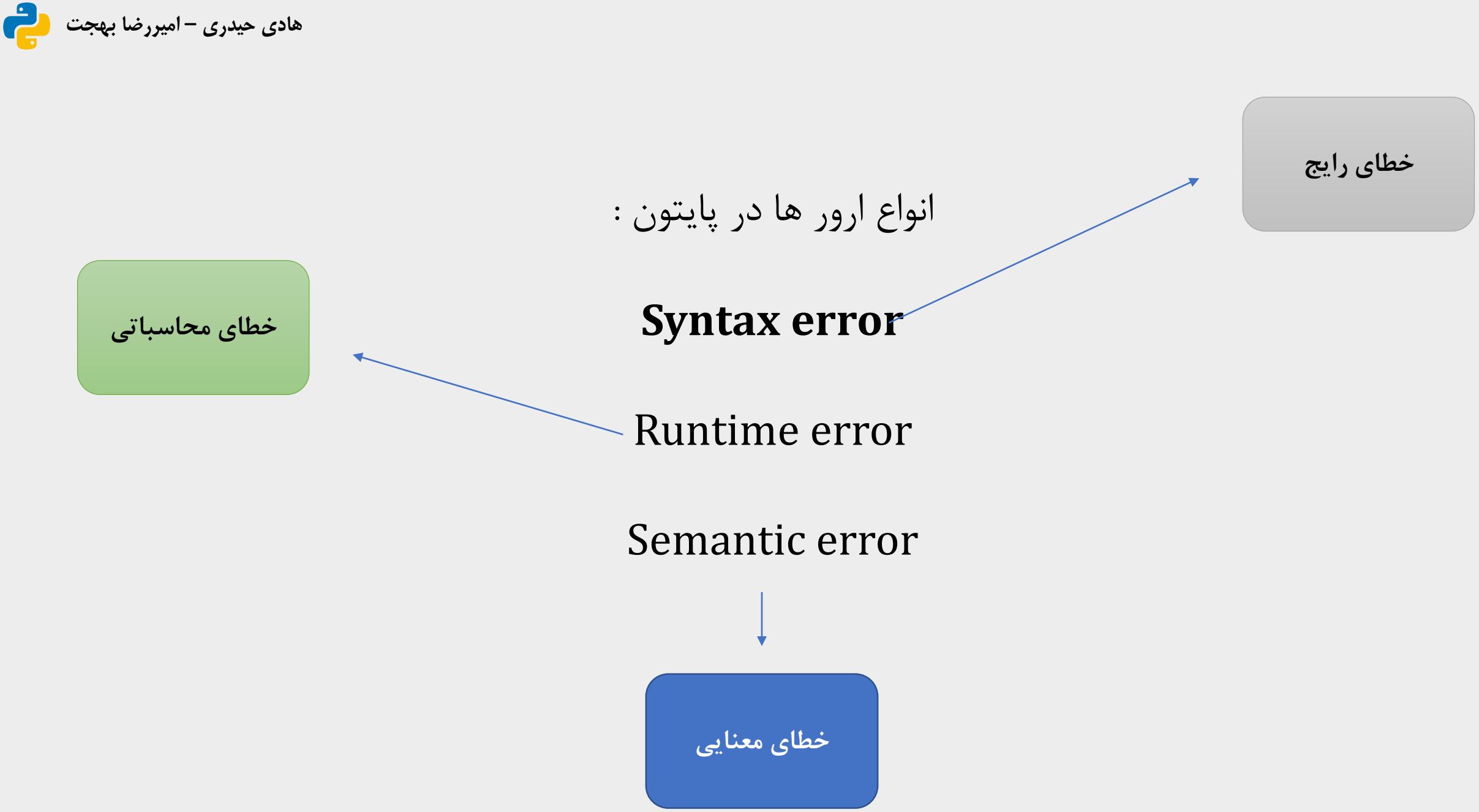
Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Python Logical Operators
Python Identity Operators

عملگر های منطقی پایتون
عملگر های شناسایی پایتون

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

Operator	Description	Example
is	Returns True if both variables are the same object	x is y
is not	Returns True if both variables are not the same object	x is not y


خطای رایج
انواع ارور ها در پایتون :
Syntax error
خطای محاسباتی
Runtime error
Semantic error
خطای معنایی

Keywords of python

and	A logical operator
as	To create an alias
assert	For debugging
break	To break out of a loop
class	To define a class
continue	To continue to the next iteration of a loop
def	To define a function
del	To delete an object
elif	Used in conditional statements, same as else if
else	Used in conditional statements
except	Used with exceptions, what to do when an exception occurs
False	Boolean value, result of comparison operations
for	To create a for loop
from	To import specific parts of a module

Keywords of python

if	To make a conditional statement
import	To import a module
in	To check if a value is present in a list, tuple, etc.
is	To test if two variables are equal
lambda	To create an anonymous function
None	Represents a null value
not	A logical operator
or	A logical operator
return	To exit a function and return a value
True	Boolean value, result of comparison operations
try	To make a try...except statement
while	To create a while loop

if

1. If
2. Elif
3. else
4. Short hand if
5. Short hand elif
6. And keyword
7. Or keyword

فرمت کلی :

```
If ---- :
    print()
else :
    print()
```

elif

اگر ---- گزاره درست بود :

این --- کار رو انجام بده

در غیر این صورت:

این کار --- انجام بده

حل ۷ مثال کاربردی برای if

While loop

for loop

تا زمانی که --- هست:

این --- کار رو انجام بده

تاااا وقتی که گزاره بالا صحیحه بعدش قطع کن و تمام.

برای --- در رنج یا لیست ---:

این --- کار رو انجام بده

تاااا وقتی که تعداد مشخصی دور در حلقه انجام شود.

حل 4 مثال کاربردی برای حلقه ها

۱. مثال کاربردی while
۲. مثالی دیگر از while
۳. مثال کاربردی for
۴. مثالی دیگر از for

3 مثال مهم برای تشکیل حلقه های تو در تو

Nasted

1n . if

2n . While

۱و۲ . دو مثال عددی برای تفهیم تو در تو
۳ و ۴ . اول بودن عدد

def

زمانی که به تابعی نیاز داشته باشیم به وسیله

def

اون رو مینویسیم با فرم رو به رو و بعد از اتمام کار

از بلوک بیرون میاییم و تابع را فراخوانی میکنیم

```
def function () :
        return ---
        print ----
        x+=1000
        ----
function()
```

Run the prg

حل 10 مثال کاربردی برای def

dict

ساخت یک دیکشنری کاری ساده محسوب می‌شود زیرا تنها کافی است که مقادیر داخل دو آکولاد قرار بگیرند و توسط کاما از هم جدا شوند. یک عنصر دارای یک کلید است و مقدار متناظر آن به صورت جفت «کلید: مقدار» بیان می‌شود. در حالی که مقادیر می‌توانند از هر نوع داده‌ای باشند و قابل تکرار نیز هستند

Key-value

{“amir” : “esm”}

حل 6 مثال کاربردی برای dict()