

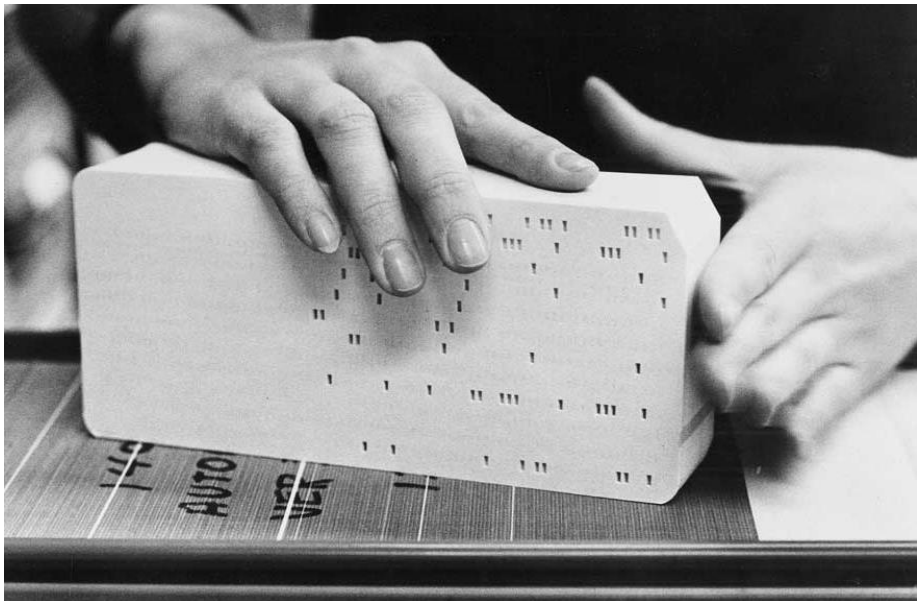
# برنامه سازی پیشرفته (مقدمات پایتون)

صادق اسکندری - دانشکده علوم ریاضی، گروه علوم کامپیوتر

[eskandari@guilan.ac.ir](mailto:eskandari@guilan.ac.ir)

# یادآوری ...

برنامه نویسی در زمان های قدیم ☹️



Punch Cards



00011010101.....0001

01110010101.....0011

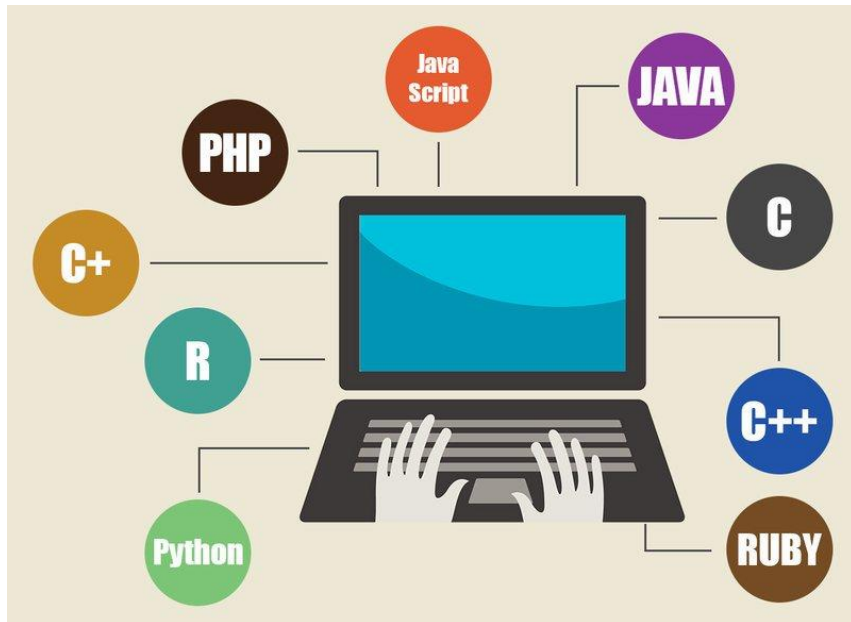
00011010101.....0111

0000000000.....1010

0000000000.....10100

# یادآوری ...

😊 برنامه نویسی امروزه



Programming Language

ترجمه

00011010101.....0001

01110010101.....0011

00011010101.....0111

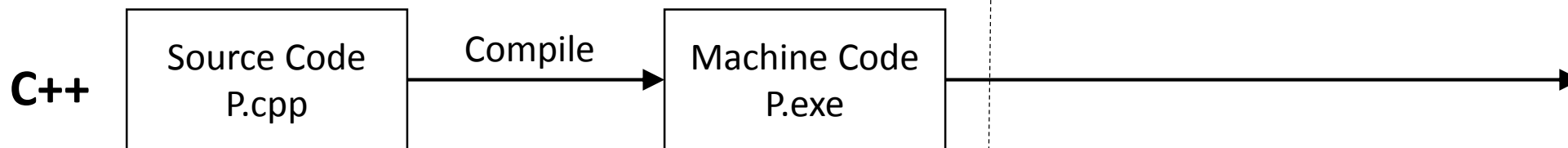
00000000000.....1010

00000000000.....10100

# یادآوری ...

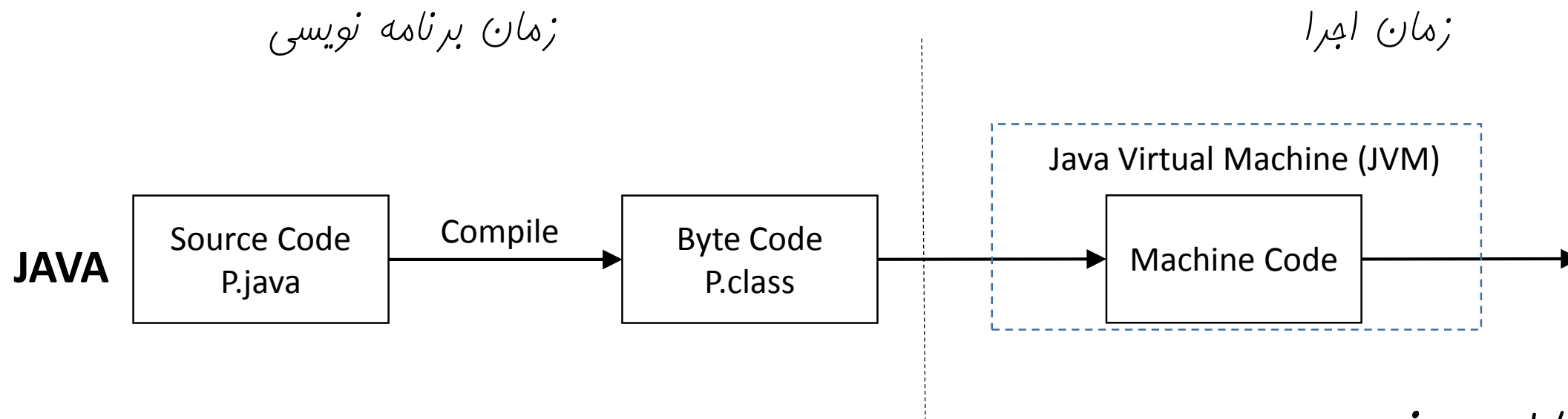
زمان برنامه نویسی  
(کامپیوتر مبدأ)

زمان اجرا  
(کامپیوتر مقصد)



- زبان کامپایلری
- سرعت اجرای بالا 😊
- اگر معماری کامپیوتر مبدأ با معماری کامپیوتر مقصد یکسان نباشد، فضا رخ می دهد ☹️

# یادآوری ...



- زبان کامپایلری-مفسری
- سرعت اجرای پایین تر نسبت به زبانهای کامپایلری ☹️
- اگر معماری کامپیوتر مبدأ با معماری کامپیوتر مقصد یکسان نباشد، فضا رخ نمی دهد 😊

# یادآوری ...

زمان برنامه نویسی

زمان اجرا

Python

Source Code  
P.py

Python Virtual Machine (PVM)

Byte Code  
P.pyc

Machine Code

• زبان مفسری

• سرعت اجرای پایین تر ☹️

• قابلیت انتقال کدها 😊

• اگر معماری کامپیوتر مبدأ با معماری کامپیوتر مقصد یکسان نباشد، فضا رخ نمی دهد 😊

- شیء گرا
- رایگان
- مفسری و قابل حمل
- نزدیک به زبان انسان
- زبان همه منظوره



## آپشن ها برای برنامه نویسی در پایتون

- پایتون ۳ یا پایتون ۲

اگرچه این دو نسخه تفاوت چندانی با یکدیگر ندارند،  
ما در این درس از نسخه ۳ استفاده خواهیم کرد.  
ممکن است برخی از برنامه هایی که مینویسیم در نسخه ۲  
با فضا مواجه شوند.

- برنامه نویسی تعاملی یا برنامه نویسی در فایل

در مد تعاملی (Interactive Mode) نتیجه هر دستور  
در همان لحظه مشخص می شود. برای اجرای چندین  
دستورالعمل، ابتدا آنها را در یک فایل با پسوند .py  
نوشته و سپس همه را با هم اجرا می کنیم. (آشنایی با  
این دو حالت، در کلاس حل تمرین 😊)



# مثال

```
1 # convert.py
2 # A program to convert Celsius temps to Fahrenheit
3
4 def main():
5     celsius = eval(input("What is the Celsius temperature? "))
6     fahrenheit = 9 / 5 * celsius + 32
7     print("The temperature is", fahrenheit, "degrees Fahrenheit.")
8
9 main()
```

توضیحات

تعریف تابع

فراخوانی تابع

What is the Celsius temperature? 8  
The temperature is 46.4 degrees Fahrenheit.

فروبی

## نکات اولیه:

توضیحات (Comments): هر فظی که با # شروع شود، توسط مفسر نادیده گرفته می شود.

شناسه ها (Identifiers): هر نامی که کاربر برای بخش های مختلف برنامه خود (متغیرها، کلاسها، توابع و ... انتخاب می کند شناسه نام دارد. یک شناسه در پایتون می تواند شامل کاراکترها (بزرگ و کوچک)، اعداد و فظ زیر (\_) باشد به گونه ای که با عدد شروع نشود و کلمه کلیدی نباشد.

```
False      class      finally    is          return
None       continue  for        lambda     try
True       def       from      nonlocal   while
and        del       global    not        with
as         elif      if         or         yield
assert     else     import    pass
break     except   in        raise
```



کلمات کلیدی در پایتون

## نکات اولیه:

پایتون نسبت به حروف بزرگ و کوچک حساس است (Case Sensitive).

استفاده از ; در انتهای دستورالعمل ها اختیاری است ولی بهتر است استفاده نشود.

```
1 # this function definition starts a new block
2 def add_numbers(a, b):
3     # this instruction is inside the block, because it's indented
4     c = a + b
5     # so is this one
6     return c
7
8 # this if statement starts a new block
9 if it_is_tuesday:
10     # this is inside the block
11     print("It's Tuesday!")
12 # this is outside the block!
13 print("Print this no matter what.")
```

بلاک ها: در جاوا و C++، بلاکهای کد به وسیله علامت های { } مشخص می شوند.

در پایتون از تورفتگی (Indentation) برای مشخص کردن بلاک ها استفاده می شود.

## ورودی و خروجی

به روش های مختلفی می توان از کاربرد ورودی گرفته و یا خروجی را در اختیار کاربر قرار داد.

```
print("hello world!")  
a = input("enter a number: ")
```

استفاده از ورودی و  
خروجی استاندارد

**تمرین:** برنامه Convert.py را به  
گونه ای تغییر دهید که درجه  
سلیسوس را از یک فایل به نام s.txt  
خوانده و نتیجه را در فایل  
f.txt ذخیره کند.

```
with open('myfile.txt', 'w') as myfile:  
    myfile.write("Hello!")  
  
with open('myfile.txt', 'r') as myfile:  
    data1 = myfile.read()
```

استفاده از فایل



روش های دیگری نیز وجود دارند مانند واسط های گرافیکی کاربرد که در ادامه درس با آنها آشنا خواهیم شد

# انواع داده (data type)

به انواع اطلاعاتی که قابل پردازش توسط زبان برنامه نویسی باشند، نوع داده گفته میشود.

یک نوع داده عبارت است از مجموعه ای از مقادیر به همراه مجموعه ای از عملگرها  
بر روی آن مقادیر

Data type = set of values (domain) + set of operators

Integer =  $Z + \{+, *, /, \dots\}$

انواع داده درون ساخت (Built-in data type)  
Numbers, Strings, Lists, Dictionaries, Tuples, Files, Sets,

انواع داده کلاسی

Student, Teacher, Car, TV, ....

انواع داده در پایتون

# اعداد (Numbers)

انواع عددی در پایتون شامل انواع صحیح (int) و انواع ممیز شناور (float) می باشند.

## int

Operator	Example	Result
+	28 + 10	38
-	28 - 10	18
*	28 * 10	280
//	28 // 10	2
%	28 % 10	8
**	28 ** 10	296196766695424

## float

Operator	Example	Result
+	28.0 + 10.0	38.0
-	28.0 - 10.0	18.0
*	28.0 * 10.0	280.0
/	28.0 / 10.0	2.8
//	28.0 // 10.0	2.0
%	28.0 % 10.0	8.0
**	28.0 ** 10.0	296196766695424.0

شکل های مختلف اعداد ممیز شناور در پایتون:

5.4, 0.0, 4., .5, 3.8e15, 3.8e+15, 2.5e-3

# اعداد (Numbers)

بر خلاف زبانهایی مانند جاوا و C++، طول یک شیء `int` در پایتون محدودیتی ندارد 😊

```
a = 2 ** 940
```

😊 یک عدد  $2^{940}$  رقمی

```
a = Math.pow(2,940)
```

😞 Overflow

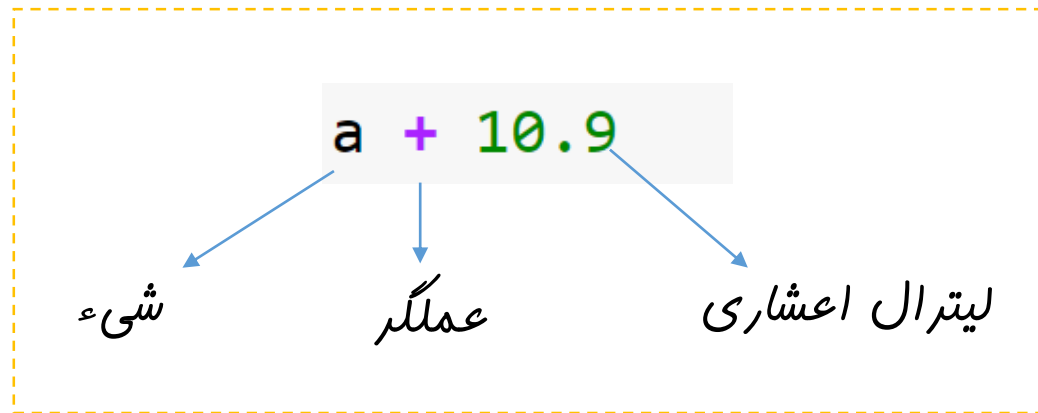
پایتون از استاندارد IEEE754 برای ذخیره اعداد اعشاری استفاده می کند. بنابراین، طول یک شیء `float` در پایتون محدود است 😞

2.2250738585072014e-308

1.7976931348623157e+308

# اعداد (Numbers)

یک عبارت (Expression)، ترکیبی است از لیترالها، اشیاء و عملگرها که دارای یک مقدار است.



یک عبارت

```
a, b, c = 2, 4, 6  
d = 1 + a * 2 + b * c ** 2  
print(d)  ؟؟؟  
فروچی؟؟؟
```

149

برای تعیین مقدار یک  
عبارت پیچیده تر، نیاز به  
دانستن اولویت عملگرها  
داریم



## اعداد (Numbers)

برای تغییر اولویت عملگرها از پرانتز گذاری استفاده می کنیم 😊

```
a,b,c = 2,4,6
d = (1+a)*2+((b*c)**2)
print(d) ؟؟
```

582

```
x = 5.0 * 2
print(x)
```

10.0

**سوال:** نوع x چیست؟ (10 یا 10.0)

زمانی که در یک عبارت هم مقادیر صحیح و هم مقادیر اعشاری وجود داشته باشد، مقدار صحیح ابتدا تبدیل به مقدار اعشاری می شود. به این رفتار تبدیل نوع ضمنی (Implicit) گفته می شود

## اعداد (Numbers)

```
x = int(5.0) * 2  
print(x)
```

10

گاهی برنامه نویس می خواهد نوع اعشاری را به صریح تبدیل کند. برای این کار باید از تبدیل نوع صریح (Explicit) استفاده کند.

👹 در استفاده از تبدیل نوع صریح دقت کنید زیرا ممکن است بخشی از اطلاعات از بین برود 👹

## اعداد (Numbers)

مثال هایی از چاپ عبارت های عددی

```
print(3)
```

3

```
print(1+2)
```

3

```
print("1 + 2 = " + (1+2))
```

Error

```
print("1 + 2 = " + str((1+2)))
```

1 + 2 = 3

```
print("1 + 2 = %d and 1 + 3 = %d"%(3,4))
```

1 + 2 = 3 and 1 + 3 = 4

## اعداد (Numbers)

مثال هایی از چاپ عبارت های عددی

```
print(3.5)
```

3.5

```
print('1.4 + 1.3 = ' + str(1.4 + 1.3))
```

2.7

```
print('1.400 + 1.301 = ' + str(1.400 + 1.301))
```

1.400 + 1.301 = 2.700999999996

```
print('1.400 + 1.301 = %.3f'%(1.400 + 1.301))
```

1.400 + 1.301 = 2.701

```
print('1.400 + 1.301 = %d'%(1.400 + 1.301))
```

1.400 + 1.301 = 2

```
1 import math # Makes the math library available.
2 def main():
3     print("This program finds the real solutions to a quadratic")
4     print()
5     a = float(input("Enter coefficient a: "))
6     b = float(input("Enter coefficient b: "))
7     c = float(input("Enter coefficient c: "))
8
9     discRoot = math.sqrt(b * b - 4 * a * c)
10    root1 = (-b + discRoot) / (2 * a)
11    root2 = (-b - discRoot) / (2 * a)
12
13    print()
14    print("The solutions are: ", root1, root2 )
15
16 main()
17
```

This program finds the real solutions to a quadratic

Enter coefficient a: 10

Enter coefficient b: 20

Enter coefficient c: 2

The solutions are: -0.10557280900008408 -1.894427190999916

## اعداد (Numbers)

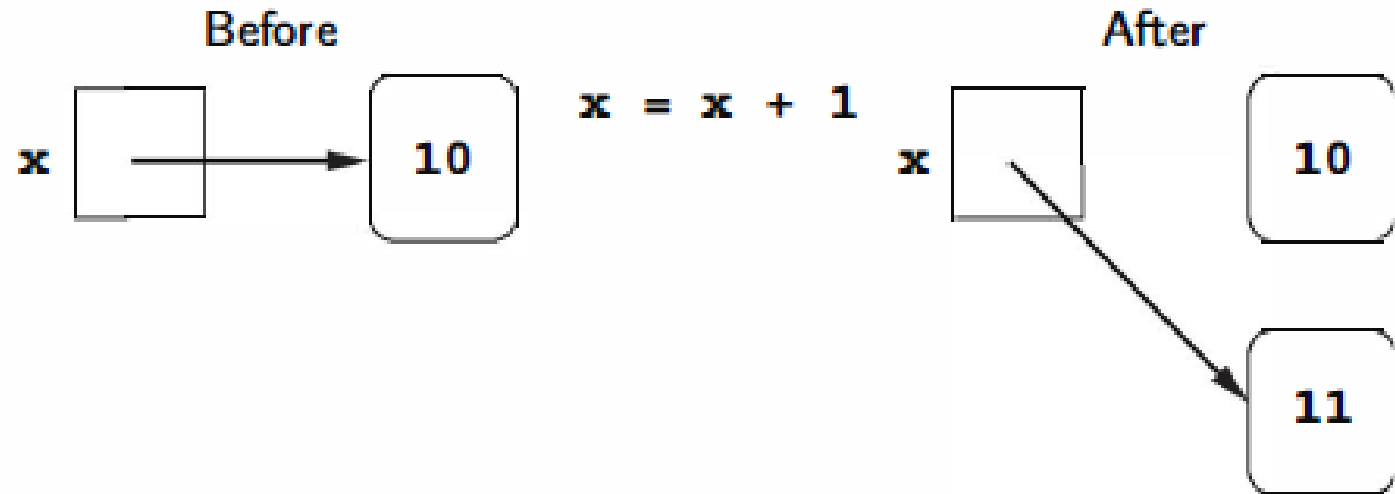
استفاده از کتابخانه math

# عملگر تخصیص

یکی از ویژگی‌های مهم پایتون، تخصیص حافظه پویا به اشیاء (متغیرها) است 🦋

```
1 x = 10
2 print(id(x))
3 x = x + 1
4 print(id(x))
```

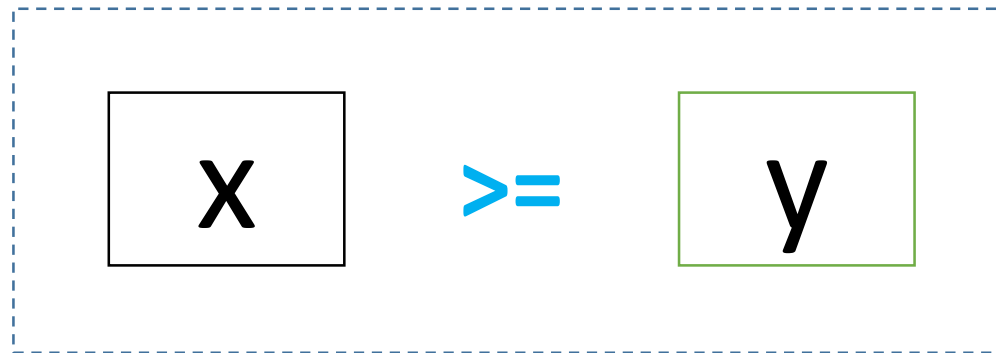
8791396426416  
8791396426448



`id(x)`: آدرس کلمه ای که  
`x` به آن اشاره می کند

## عملگرهای رابطه ای

یک عملگر رابطه ای، دو مقدار را با یکدیگر مقایسه می کند. ( $=$ ،  $\neq$ ،  $>$ ،  $<$ ،  $\geq$ ،  $\leq$ )



True or False (Boolean)