آزمایشگاه ریاضی (آموزش Sage)

درس دوم:

آشنایی مقدماتی با Python

مدرس: میثم مدنی madani@mehr.sharif.ir

دانشکده علوم ریاضی- دانشگاه صنعتی شریف

ترم اول سال ۱۳۹۱

نسخه ابتدایی (لطفا نظرات خودت و ایرادات فایل را ایمیل کنید)

Python

(Simple & Powerful)

۱- آشنایی و تاریخچه

۲- مفاهیم اولیه و نصب Python

۳- نوشتن اولین برنامه

۴- متغیرها و توابع مهم

۵- کلاس‌ها و شئ گرایی

آشنایی

- یکی از معدود زبان‌های برنامه نویسی است که سادگی و قدرت را در کنار هم دارد.
- با ساده ترین توابع و بدون نیاز به پرداختن به جزئیات (مثلا تعریف متغیر، نامهای تابع طولانی و ...)
- بیشتر تمرکز روی برنامه و الگوریتم است تا ساختارها و توابع و ...
- یک ساختمان داده سطح بالا، کارآمد و یک روش ساده برای برنامه نویسی شئ‌گرا دارد
- Sage بر اساس پایتون نوشته شده است از اینجا است که می‌بینیم این زبان برنامه نویسی قدرتمند است و در ضمن توابع و امکانات این زبان در اغلب موارد در sage نیز قابل استفاده هستند.

تاریخچه

- سال ۱۳۹۰ موسسه تحقیقات بین المللی ریاضیات و کامپیوتر هلند (CWI) توسط Guido van Rossum ساخته شد.
- نام Python را از روی کمدی Monty Python's Flying Circus انتخاب کرد.
- سیستمی است که روی تمام سیستم‌های عامل windows, Unix, OS/2 و ... قابل اجرا است. در صورتی که در ابتدا برای سیستم عامل Amoeba ساخته شده بود

مفاهیم اولیه

- یک **برنامه** دنباله‌ای از دستورات است که چگونگی انجام محاسبات را مشخص می کند. (حل دستگاه، محاسبه ریشه های یک تابع، محاسبات نمادین مانند یافتن و جایگزین کردن یک کلمه)
- **اشکال زدایی:** خطاهای برنامه نویسی را Bug می‌گویند. تصحیح خطا را debugging می گویند.
- انواع خطا

1- syntax error(pprint)

2- runtime error(1/0)

3-semantic error (آب بریز!)

یکی از مهمترین مهارت‌ها در برنامه نویسی یافتن و رفع خطاهاست

نصب Python

نصب در ویندوز

۱- http://www.python.com/download

۲- آخرین نسخه را دانلود کنید.

۳- آن را نصب کنید! (چندتا next).

٤- IDLE (Python GUI) را اجرا کنید.

نصب در لینوکس

نصب هست!

نوشتن اولین برنامه

Python

Print “hello world”

Press <enter>

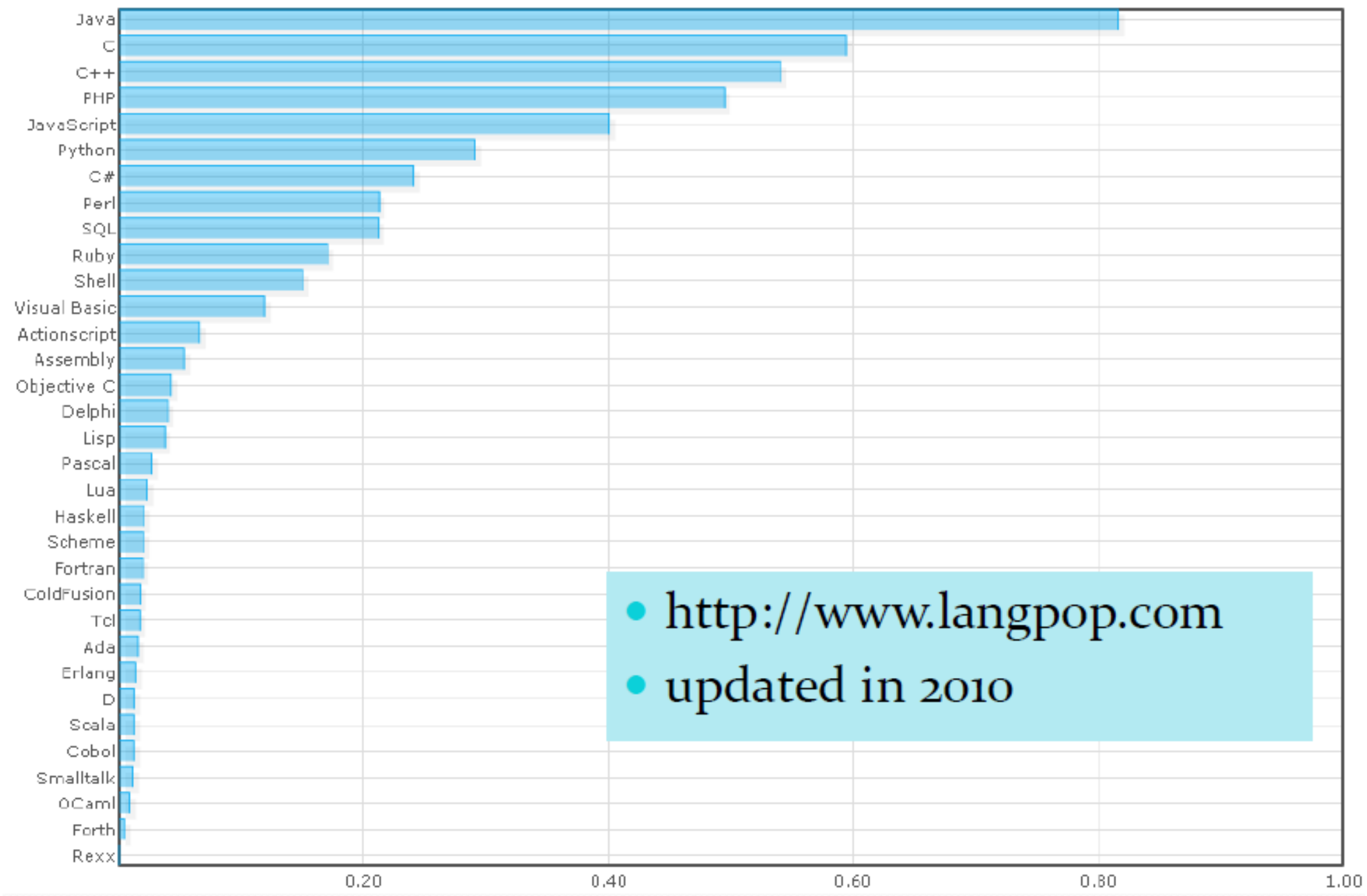
Java

```
public class HelloWorldExample{

  public static void main(String args[]){

System.out.println("Hello World !");
{
{
```

Right click - run


Java
C
C++
PHP
JavaScript
Python
C#
Perl
SQL
Ruby
Shell
Visual Basic
Actionscript
Assembly
Objective C
Delphi
Lisp
Pascal
Lua
Haskell
Scheme
Fortran
ColdFusion
Tcl
Ada
Erlang
D
Scala
Cobol
Smalltalk
OCaml
Forth
Rexx
0.20
0.40
0.60
0.80
1.00
• http://www.langpop.com
• updated in 2010

نوشتن یک برنامه

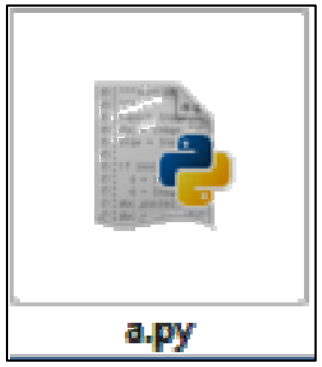
دستورات را می توان به سه صورت وارد کرد.

- ۱- به طور مستقیم در Python shell که در ابتدا باز می شود.

```
>>>print"hi"
hi
```

۲- در یک فایل جدید (با new window) و ذخیره و سپس اجرا Run یا کلید میانبر cnrl+f5

```
print"hi"
```



۳- می توانید با کلید بر روی فایل ذخیره شده مستقیما آن را اجرا کنید.

متغیرها و توابع مهم

۱- از # می توان برای گذاشتن یادداشت استفاده کرد که در زمان اجرا دیده نخواهند شد.

```
#tozihate file …
```

- از دستور help(str) برای کمک می توانید استفاده کنید

```
help(input)
```

- رشته‌ها را در "*"وارد می کنیم ولی برای رشته‌های چند خطی از ''' ******* ''' استفاده می کنیم.

```
print "hi there"
print('''hi
There''')
```

- برای رشته‌های چند خطی از \ هم می توان استفاده کرد.

```
print("hi \
There")
```

- نسخه های جدید پایتون از نوشته‌های فارسی یا عربی و ... نیز حمایت می کند

```
print "سلام"
```

توابع مهم

- abs(number) قدر مطق
- cmath.sqrt(number) محاسبه ریشه دوم حتی برای اعداد منفی
- float(object) یک رشته یا عدد را به یک عدد اعشاری تبدیل می کند
- help() کمک
- input(prompt) گرفتن ورودی از کاربر
- int(object) تبدیل یک رشته یا عدد به یک عدد صحیح
- long(object) تبدیل یک رشته یا عدد به یک عدد صحیح طولانی
- math.ceil(number) کوچکترین عدد صحیح بزرگتر مساوی یک عدد صحیح
- math.floor(number) جزء صحیح
- math.sqrt(number) ریشه دوم بدون در نظر گرفتن اعداد منفی
- pow(x, y[, z]) z بهx^y پیمانه
- raw_input(prompt) گرفتن ورودی به عنوان یک رشته
- repr(object) برگرداندن نمایش رشته ای یک شی
- round(number[, ndigits]) رند یک عدد با دقت دلخواه
- str(object) تبدیل یک مقدار به یک رشته

لیستها

c=[1,'hello',3,4,5,6]

- c[0], c[1], c[2], c[3],c[-1]

  1, 'hello',3,4,6

- c[0:2]

  1, 'hello'

- c[2:]

  [3,4,5,6]

- c[:2]

  [1,'hello']

اندیس گذاری پایتون از صفر شروع می شود

لیستها

- c=[n^3 for n in range (1,6)]

[1, 8, 27, 64, 125]

- cd=[n**3/2 for n in range (1,6)]

[0, 4, 13, 32, 62]

- 7 in cd

False

حلقه ها

- حلقه while

```
x=1
while x<100:
    print x, '\t',x**2
    x=x+3
```

- اتمام while؟

حلقه ها

- حلقه FOR

```
x=1
while x<100:
    print x, '\t',x**2
    x=x+3
```

- اتمام while؟

حلقه ها

- حلقه for

```
for nn in range(20):
    print nn
---------
E=[1,2,2,3,5]
for nn in E:
    print nn
--------
for prog in ["pascal","c++","python"]:
  print "I can learn " + prog
```

حلقه ها

- حلقه while

```
x=1
while x<100:
    print x, '\t',x**2
    x=x+3
```

- حلقه while

لیست ها

- list(s) Converts s to a list.
- s.append(x) Appends a new element, x, to the end of s.
- s.extend(t) Appends a new list, t, to the end of s.
- s.count(x) Counts occurrences of x in s.
- s.index(x [,start [,stop]]) Returns the smallest i where s[i] ==x.
  (start and stop optionally specify the starting and ending index for the search.)
- s.insert(i,x) Inserts x at index i.
- s.pop([i]) Returns the element i and removes it from
- s.remove(x) Searches for x and removes it from s.
- s.reverse() Reverses items of s in place.
- s.sort() sort elements
- min(s)
- max(s)

برخی نکات

```
if a < b:
pass # Do nothing
else:
z = a
----------
+= (a+=5    or     a=a+5)   -=        *=        **=
----------
[a+b+c for a in A for b in B for c in C]
```

عملگرهای مجموعه ای

```
s = set([3,7,8,17])
t = set("Hello")
>>> print t
set(['H', 'e', 'l', 'o'])
a = t | s # اجتماع
b = t & s # اشتراک
c = t – s # تفاضل
d = t ^ s # تفاضل متقارن
t.add('5') #  اضافه کردن
s.remove(9)#9 حذف
```

تابع و چند فایلی

```
# file : div.py
def divide(a,b):
q = a//b # If a and b are integers, q is an integer
r = a - q*b
return (q,r)
----------
import div
a, b = div.divide(2305, 29)
-------
from div import divide
```

کلاس

تعریف کلاس بعد از import و اول فایل نوشته می شود.

```
Class Point:
    pass
```

یک کلاس به نام Point می سازد

اعضای این کلاس را **شی** می گوییم

شی

برای ساخت یک ماشین به یک سری ابزار نیاز داریم و هر ابزاری ویژگی و روش کار خود را دارد.

مثلا خود ماشین.

لاستیک: رنگ، جنس

روش: انتقال نیرو

Car
ATTRIBUTES: make model color wheels seats autobody motor
METHODS: start drive stop

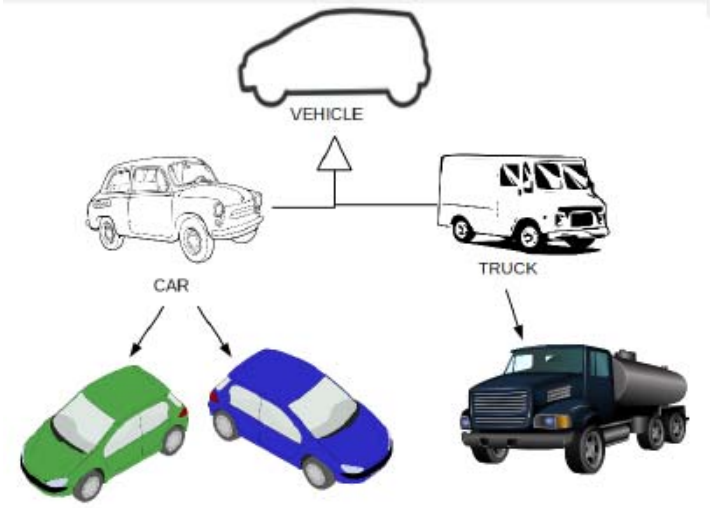
شی

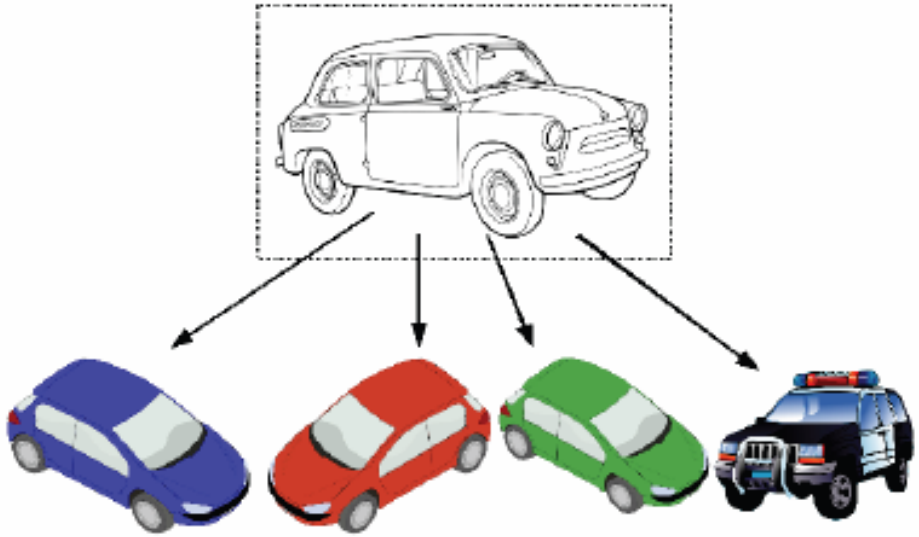
- یک راه ساخت ماشین این است که تمامی قطعات را همانجا تراشکاری، ریخته گری و آماده کرده و یک ماشین را بسازیم!
- راه ساده تر استفاده از ابزارهای از قبل آماده شده است. هر یک از ابزارها را یک شی می گوییم
- کارهای داخل هر کارگاه یک کلاس هستند
- هر شی می تواند توسط خود شخص نوشته شده باشد یا توسط دیگران و از طریق اشتراک گذاری یا خریداری استفاده شوند.

شی

مزایای شی گرایی

- بهینه شدن ساختار برنامه
- عدم نیاز به دوباره نویسی کدها
- کپسوله سازی (اشیا کمترین ارتباط را با هم داشته باشند تا در صورت نیاز فقط یکی را نیاز باشد تغییر دهیم.)





شی

```
class sage ():
    def __init__(self):
        print "Welcome to sage"
        self.a()
    def a (self):
        print '2 * 2 = 4'
d = sage()
d
```

شی

```
class Person(object):
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def is_old(self):
        return self.age > 40
class A(object):
 def sayHello(self):
  print "Hello A!"
class B(A):
 pass
#-------------------------------
person = Person('G. H. Hardy', 70)
print person.is_old()
#verasat
b = B()
b.sayHello()
```

تفاوتهای شئ گرا و تابع گرا

آیا کارهایی هست که بتوان با شی گرایی انجام داد ولی با تابع گرایی نتوان انجام داد؟

سرعت کار چگونه است.

بسته Numpy

Numeric Python مجموعه ای از ماژول هاست ، برای انجام عملیات ریاضی مخصوصا روی آرایه ها

نصب بسته از

http://sourceforge.net/projects/numpy

دانلود و نصب کنید.

Numpy

Arithmetic add, subtract, multiply, divide, remainder

Powers and Logs power, exp, log

Comparision Logic equal, not_equal, greater, greater_equal, less, less_equal, minimum, maximum

Trigonometry sin,cos,tan,sinh,cosh,tanh,arcsin, arccos, arctan, arctanh

Bitwise bitwise_and, bitwise_or, bitwise_xor, bitwise_not,

تمرینات

- (تحویلی) برنامه ای بنویسید که یک عدد n را دریافت کند و تا n مرحله اعداد فیبوناتچی را نمایش دهد.
- (تحویلی) برنامه ای بنویسید که ۱۰ رشته را دریافت کند و آنها را به ترتیب حروف الفبا مرتب کند.
- کلیدهای میانبر در python IDLE را بیابید.
- تفاوت برنامه نویسی شی گرا و تابعی را با جزئیات بیان کنید

پروژه

- در مورد بسته pygame گزارشی تهیه کنید (نصب، اصول و ...) و یک بازی نسبتا ساده را با آن بنویسید (ایده بازی از خودتان باشد!)
- گزارشی در مورد اسکریپت نویسی در پایتون تهیه کنید.
- گزارشی از بسته های پرکاربرد python ارائه دهید.
- یک دیکشنری در python بنویسید که به تعداد دلخواه رشته و ترجمه آن را بگیرد و در زمان تایپ اولین حرف تمامی رشته های آغاز شده با آن و با نوشتن دو حرف تمامی رشته های آغاز شده با آن رشته و ... را در زیر آن نمایش دهد.

تمرینات جلسه دوم

- تفاوت محاسباتی شی گرا و تابعی را با جزئیات بیان کنید

پروژه جلسه دوم

-