



Python lecture 3

جلسه قبل

- نحوه تعریف و مقداردهی متغیر ها
- معرفی نوع داده رشته
- ورودی و خروجی (دستورات input, print)
- عملگر های مقایسه‌ای
- عملگر های منطقی بر روی نوع داده boolean
- بلاک بندی کدها
- انواع دستورات شرطی
- حلقه تکرار while
- حلقه تکرار for



امروز

- دستور break
- رشته‌ها
- آرایه
- فایل
- سوکت
- Banner grabbing
- راه اندازی و استفاده مقدماتی از git

دستور break

- خروج اضطراری از یک بلاک به بلاک بالاتر

```
a=10
b=20
while a<100:
    while b<100:
        if (b%50==0):
            break
        print(b)
        b+=1
    a+=10
```



Jump out of this while loop

رشته‌ها

- با استفاده از عملگر [] می‌توان به عناصر یک رشته دسترسی پیدا کرد

اندیس‌ها از صفر شروع می‌شوند → `my_string='p y t h o n'`

indexes : 0 1 2 3 4 5

Neg index: -6 -5 -4 -3 -2 -1

`print(my_string[0])` → "p"

`print(my_string[1])` → "y"

`print(my_string[2])` → "t"

`print(my_string[-1])` → "n"

`print(my_string[-2])` → "o"

`print(my_string[-3])` → "h"

- با استفاده از اندیس منفی می‌توان به آخر رشته دسترسی پیدا کرد

رشته‌ها

- دسترسی به یک زیر رشته : [start : stop : step]
- در صورتی که فقط شروع و خاتمه داشته باشیم به طور پیشفرض step=1
- می‌توان فقط از ":" استفاده نمود و شروع و خاتمه مشخص نکرد

```
s="abcd"  
print(s[1:5:2])  
print(s[1:3])  
print(s[:])  
print(s[::-1]) #reverse the string  
print(s[5:0:-2])
```

رشته‌ها

- رشته‌ها غیر قابل تغییر هستند (immutable)

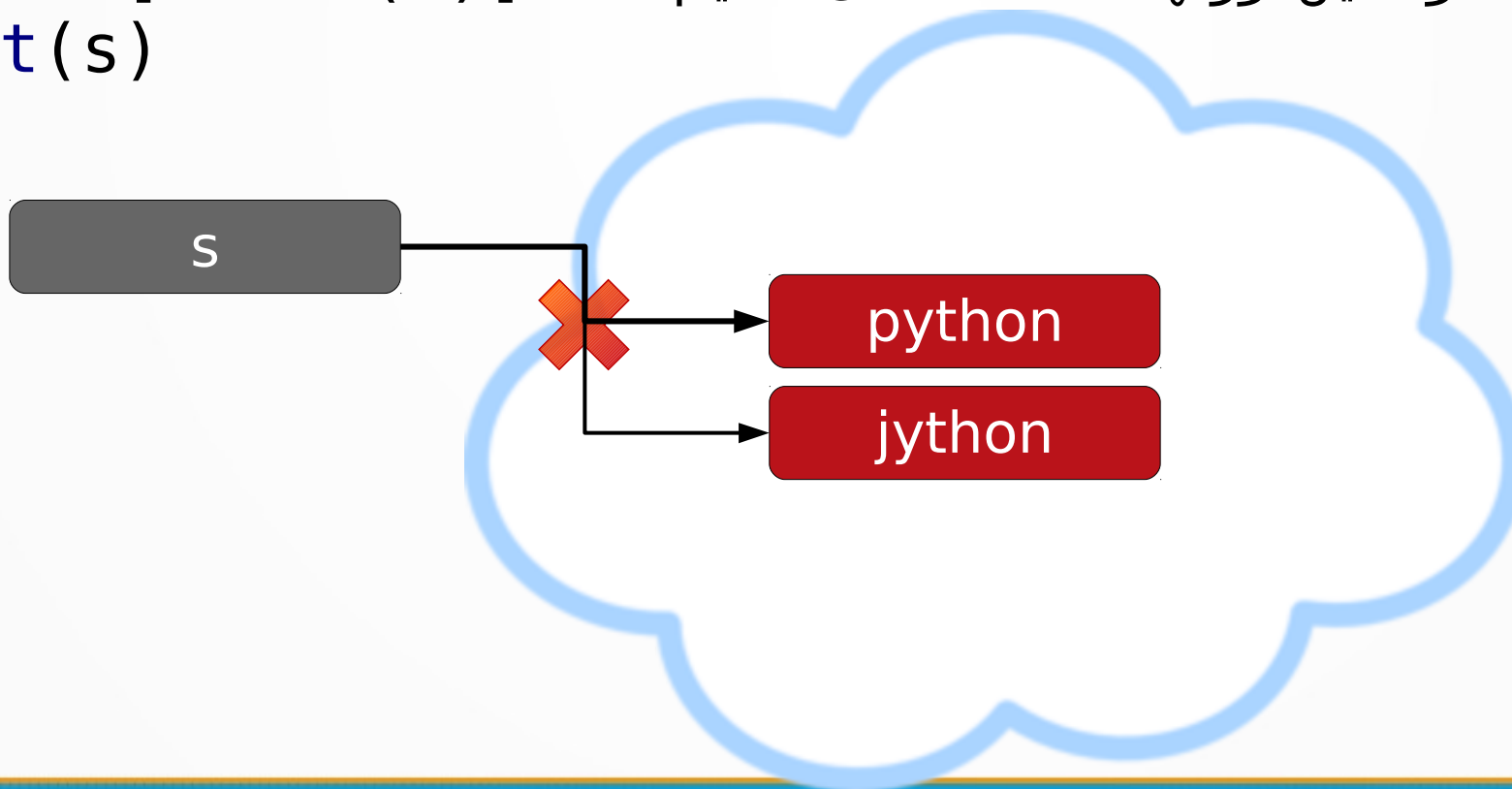
```
s="python"
```

پایتون اجازه تغییر در رشته را نمی‌دهد
بجای دستور فوق از این روش استفاده می‌کنیم

```
s='j'+s[1:len(s)]
```



```
print(s)
```



آرایه

- دنباله ای از عناصر (هم نوع یا غیر هم نوع)
- نحوه تعریف آرایه

```
emails=list()
```

```
emails.append(عنصری که می‌خواهیم به آرایه اضافه شود)
```

- نحوه دسترسی به عناصر آرایه با استفاده از اندیس عنصر انجام می‌شود

```
emails[0]
```


فایل

- با استفاده از تابع open می‌توان یک فایل را باز نمود
- با استفاده از دستور read می‌توان محتوای فایل باز شده را خواند
- با استفاده از تابع split می‌توان رشته خوانده شده را به یک آرایه تبدیل نمود

```
file = open('emails', 'r')
file_contents = file.read()
list_of_emails = file_contents.split('\n')
''''''
```

```
show emails that contains outlook
''''''
```

```
for email in list_of_emails:
    if 'outlook' in email:
        print(email)
```

- A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent to. An endpoint is a combination of an IP address and a port number.

برنامه نویسی سوکت چیست؟

- برقراری ارتباط بین دو node (کامپیوتر و ...) در شبکه
- در معماری client/server یک node بر روی یک پورت منتظر ارسال پیام می ماند که این node سرور نام دارد
- نود دیگر که کلاینت نام دارد درخواست خود را به سمت سرور می فرستد و بسته به نوع درخواست پاسخ دریافت می کند و یا پاسخی دریافت نمی کند

ماژول سوکت

- شامل توابعی است که به برنامه نویس امکان برقراری ارتباطات شبکه را می‌دهد
- نحوه استفاده از این ماژول:

```
import socket
socket.setdefaulttimeout(5)
s=socket.socket()
s.connect(('127.0.0.1',21))
ans = s.recv(1024)
s.close()
print(ans) → 220 symm FTP server
(Version6.4/OpenBSD/Linux-ftpd-0.17) ready
```

استفاده از banner grabbing در تست نفوذ

- با در اختیار داشتن لیستی از سرویس هایی که آسیب پذیر اند می توان بررسی کرد که آیا سرویس مد نظر ما آسیب پذیر است یا خیر

لیستی از سرویس های آسیب پذیر

- FreeFloat Ftp Server (Version 1.00)
- 3Com 3CDaemon FTP Server Version 2.0
- Ability Server 2.34
- Sami FTP Server 2.0.2
- ...

Git

- امکان دنبال کردن تغییرات اعمال شده در کد و ذخیره و بازیابی کد های قبلی
- امکان دنبال کردن تغییراتی که توسط افراد مختلف بر روی کد اعمال شده توسط مسئول پروژه
- امکان مدیریت تغییراتی که توسط افراد مختلف بر روی کد انجام شده است



استفاده از git

- ۱- با استفاده از دستور cd ترمینال وارد فولدر پروژه خود می‌شویم

- ۲- دستور git init را در ترمینال وارد می‌کنیم

- ۳- با دو دستور زیر برنامه نویس که خودمان هستیم را به گیت معرفی می‌کنیم

```
git config user.name "Mona Lisa"
```

- ۳- برای اینکه چند فایل را به ورژن کنترلر گیت معرفی کنیم دستور زیر را در ترمینال وارد می‌کنیم

```
git add all
```

- ۴- با استفاده از دستور زیر تغییرات خود را ذخیره می‌کنیم

استفاده از git

- ۴- با استفاده از دستور زیر تغییرات خود را ذخیره می‌کنیم

`git commit -am "message"` → commit all changed files with message