



برنامه نویسی با پایتون

سعید محقق / تابستان 1400

05: Functions

کار با تابع در پایتون، توسعه کد و تحویل برنامه به کاربر

■ پروژه 5: توسعه برنامه قرعه کشی (ghore keshi v2)

1. ایجاد تابع برای تایپ زمان دار

2. تشخیص نوع سیستم عامل

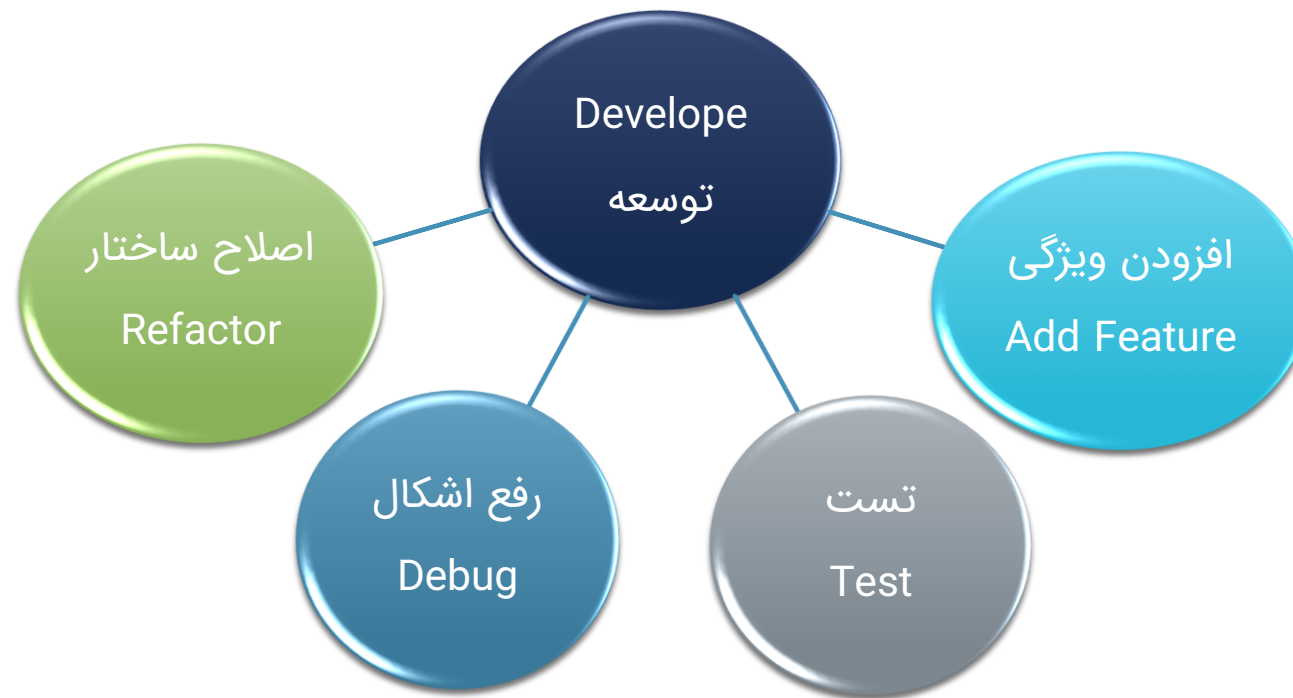
3. پاک کردن صفحه (بر اساس نوع سیستم عامل)

4. اضافه کردن تاریخ

5. تبدیل کد به فایل exe در ویندوز

What will we learn? (Keywords)

- Functions
- Identify System's OS
- Clear Screen (OS-based)
- Datetime module
- Python to Exe (in Windows)



- جدول ascii ← یک کد برای هر کاراکتر (128 کاراکتر از 0 تا 127)

<https://www.rapidtables.com/code/text/ascii-table.html>

جدول ascii

- حالت پیشرفته‌تر ← جدول unicode یا UTF-8 ← قابلیت تعریف تعداد بسیار بیشتر کاراکتر

<https://www.rapidtables.com/code/text/unicode-characters.html>

جدول unicode

- برنامه Character Map در ویندوز ← کاراکترهای Unicode و مقدار عددی آنها

■ قانون DRY در برنامه‌نویسی ← Don't Repeat Yourself

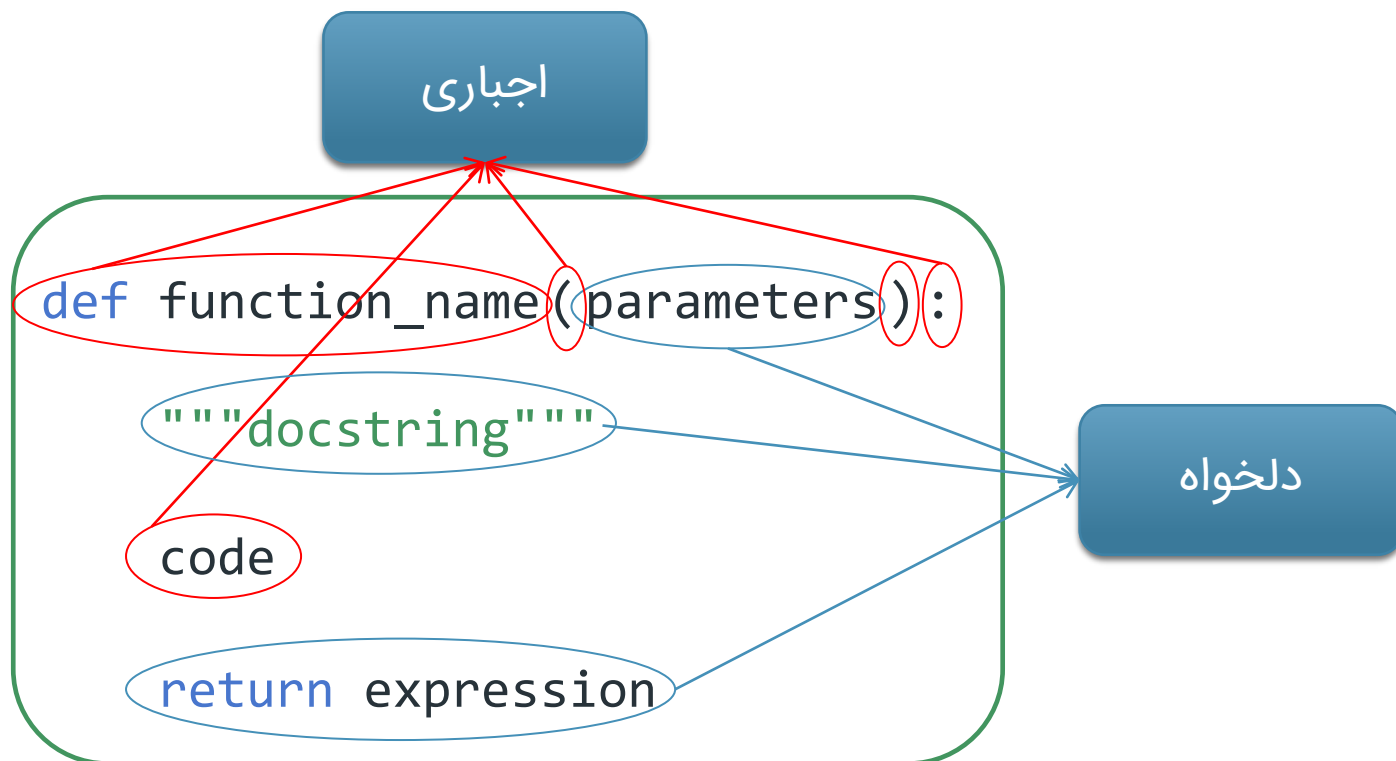
```
print('Hello')
print('-----')
print('=====')
print('-----')
print('Welcome')
print('-----')
print('=====')
print('-----')
...
```

```
def print_line():
    print('-----')
    print('=====')
    print('-----')
```

```
print('Hello')
print_line()
print('Welcome')
print_line()
...
```

■ مثال:

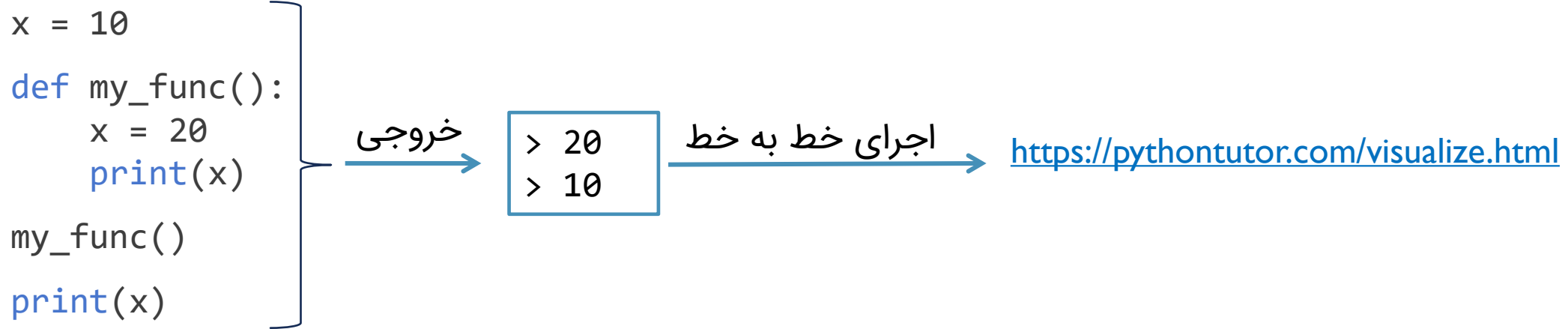
■ افزایش خوانایی کد / تقسیم وظایف / بهبود قابلیت توسعه / ...



■ شکل کلی تابع

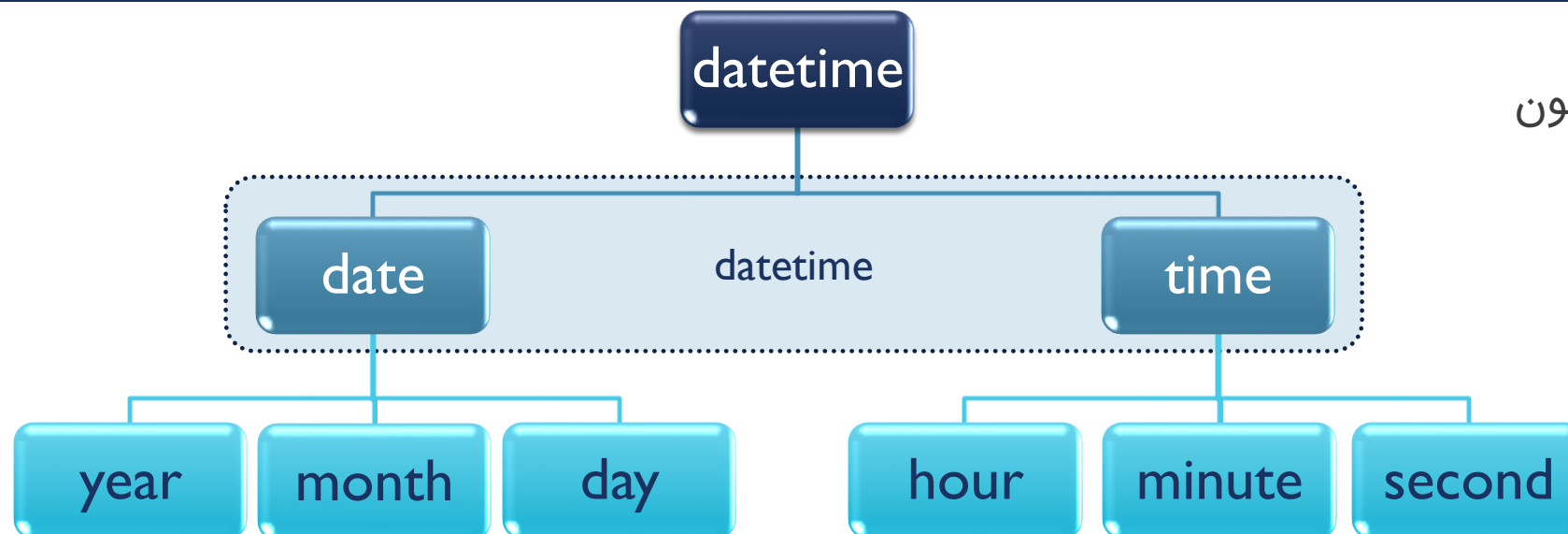
■ تابع حداقلی `def nothing():`
`pass`

مفهوم Scope و متغیرهای Local و Global در پایتون



- متغیرهایی که درون تابع تعریف می‌شوند Local هستند ← فقط درون تابع معتبر هستند / بعد از اجرای تابع از بین می‌روند
- متغیرهایی که در کد اصلی تعریف می‌شوند، Global هستند و در تمام برنامه معتبر هستند
- مثال: محاسبه قیمت بعد از اعمال مالیات

■ کار با تاریخ و زمان در پایتون



```
>>> from datetime import date, time, datetime
>>> today = date.today()
>>> today
datetime.date(2020, 1, 24)
>>> now = datetime.now()
>>> now
datetime.datetime(2020, 1, 24, 14, 4, 57, 10015)
```

تاریخ و زمان حال

- کار با تاریخ و زمان در پایتون

```
>>> from datetime import date, time, datetime
>>> date(year=2020, month=1, day=31)
datetime.date(2020, 1, 31)
>>> time(hour=13, minute=14, second=31)
datetime.time(13, 14, 31)
>>> datetime(year=2020, month=1, day=31, hour=13, minute=14, second=31)
datetime.datetime(2020, 1, 31, 13, 14, 31)
```

تاریخ و زمان انتخابی

<https://realpython.com/python-datetime/>

کار با datetime

- کار با تاریخ و زمان در پایتون

```
from datetime import datetime
import pandas as pd
```

```
date1 = datetime(2022,2,25)
```

```
date2 = date1 + pd.DateOffset(months=11)
```

```
date3 = date1 + pd.DateOffset(days=4)
```

```
date4 = date1 + pd.DateOffset(hours=25)
```

جابجایی تاریخ و زمان

```
print(date1)
```

```
print(f'{date2.year}/{date2.month}/{date2.day}')
```

```
print(date3.strftime('%D'))
```

```
print(date3.strftime('%Y/%m/%d %H:%M:%S'))
```

نمایش تاریخ و زمان

ساخت فایل EXE از کد پایتون (در ویندوز)

The screenshot shows the 'Auto Py To Exe' application window. It has a title bar with the application name and standard window controls. The interface includes a GitHub logo and 'Help Post' link in the top right. A language dropdown menu is set to 'English'. The 'Script Location' section has a text field containing 'C:/Users/Saeed/Desktop/test.py' and a 'Browse' button. Below this, the 'Onefile' section has two buttons: 'One Directory' (selected) and 'One File'. The 'Console Window' section has two buttons: 'Console Based' (selected) and 'Window Based (hide the console)'. There are four checked checkboxes: 'Icon (-iicon)', 'Additional Files (-add-data)', 'Advanced', and 'Settings'. The 'Current Command' section shows a text field with the command 'pyinstaller --noconfirm --onedir --console "C:/Users/Saeed/Desktop/test.py"'. At the bottom is a large blue button labeled 'CONVERT .PY TO .EXE'. Persian annotations with arrows point to specific elements: 'مسیر فایل اصلی پایتون' points to the script location field; 'انتخاب آیکن برنامه' points to the 'Icon' checkbox; 'دستور نهایی کنسول' points to the 'Current Command' field; 'نوع برنامه خروجی' points to the 'One Directory' button; and 'انتخاب فایل‌های جانبی' points to the 'Additional Files' checkbox.

■ نصب پکیج auto-py-to-exe

> `pip install auto-py-to-exe`

■ اجرای auto-py-to-exe از cmd

> `auto-py-to-exe`

■ برنامه رمزنگاری (cipher app)

1. انتخاب حالت رمزنگار یا رمزشکن توسط کاربر

2. انتخاب الگوریتم توسط کاربر

3. دریافت متن از کاربر

4. دریافت پارامترهای مورد نیاز از کاربر

5. انجام عملیات رمزنگاری یا رمزشکنی

6. تبدیل کد به فایل exe در ویندوز

What will we learn? (Keywords)

- Functions
- Ascii Codes
- Ciphertext (Encyption / Decryption)
- UI / UX