



# برنامه نویسی با پایتون

سعید محققى

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

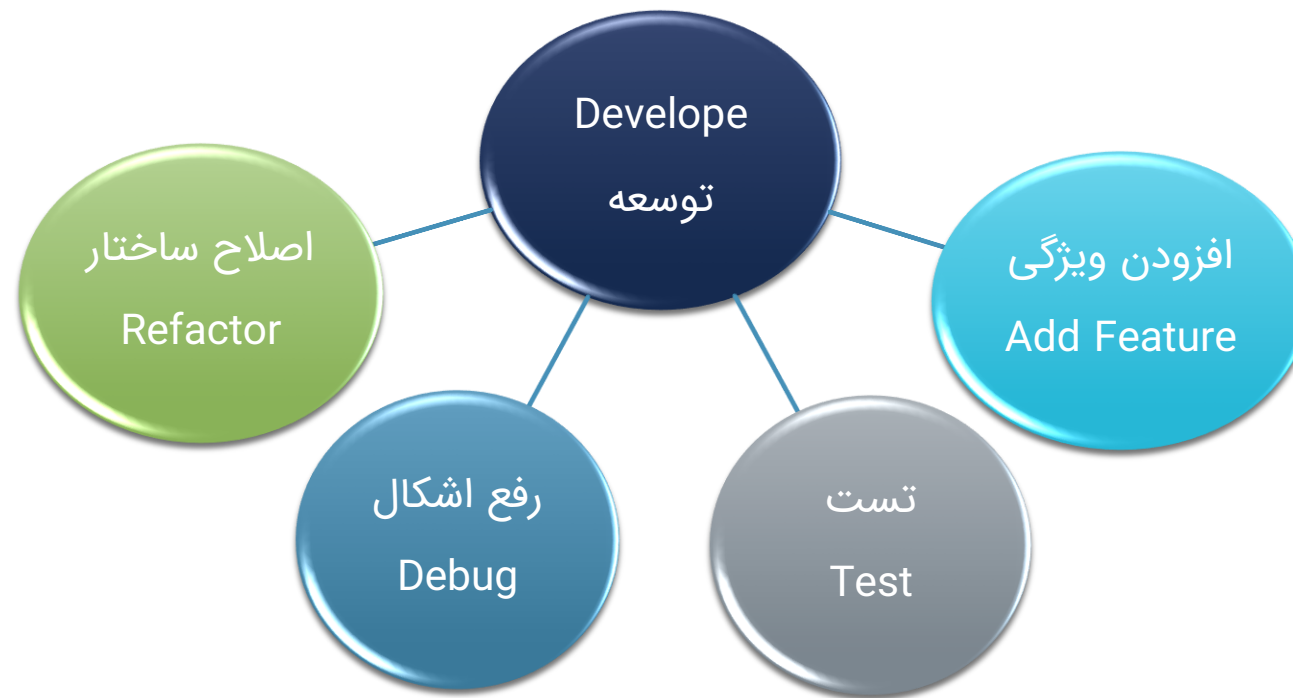
3. پاک کردن صفحه (بر اساس نوع سیستم عامل)

4. اضافه کردن تاریخ (میلادی / شمسی)

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

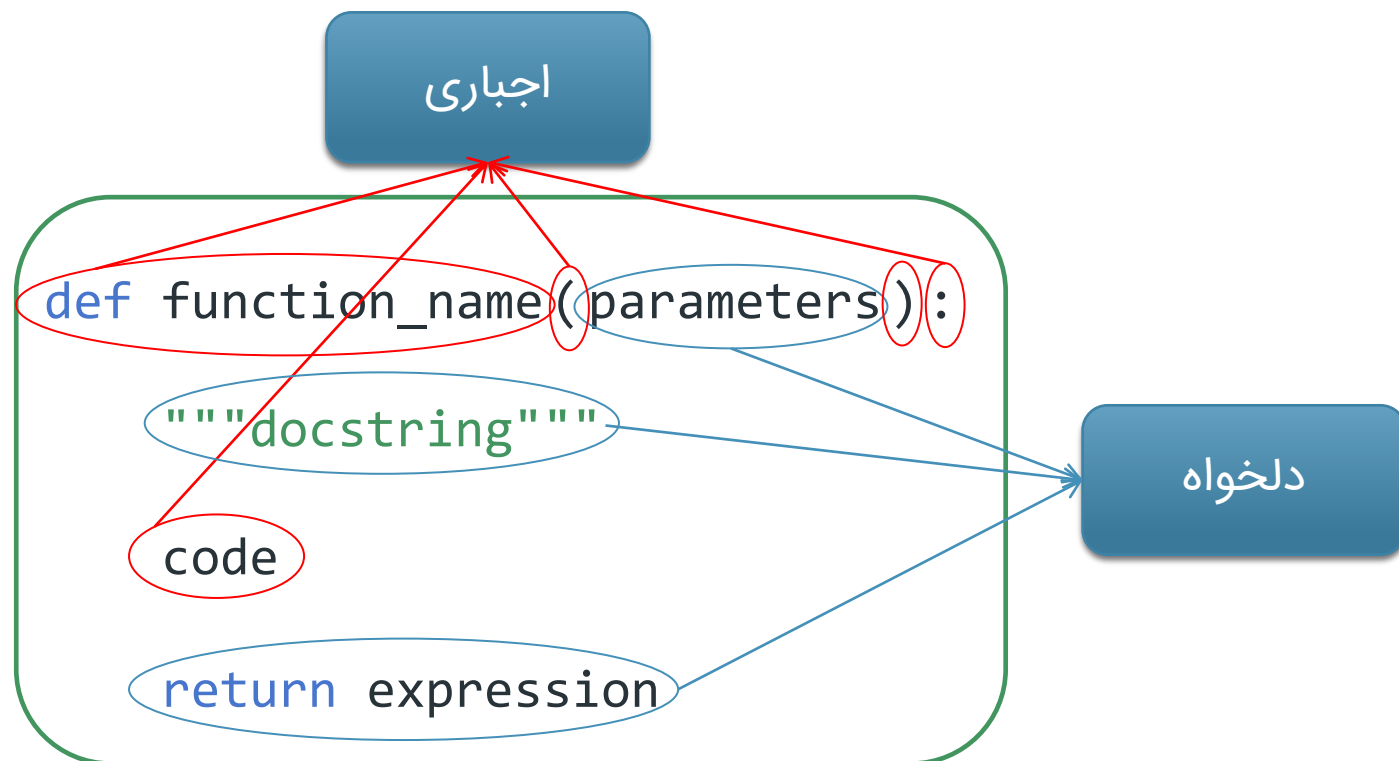
```
print('Hello')
print('-----')
print('=====')
print('-----')
print('Welcome')
print('-----')
print('=====')
print('-----')
...
```

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

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

■ مثال:

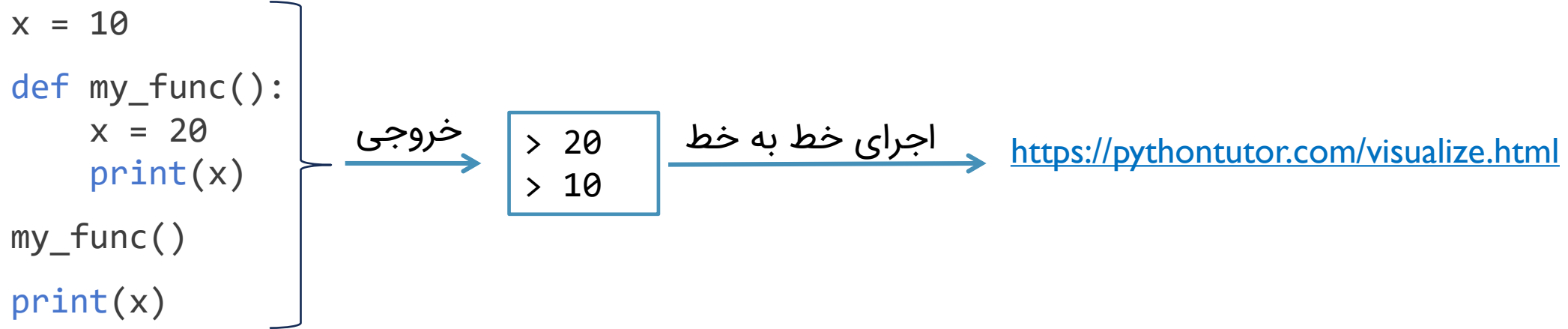
■ افزایش خوانایی کد / تقسیم وظایف / بهبود قابلیت توسعه / ...



■ شکل کلی تابع

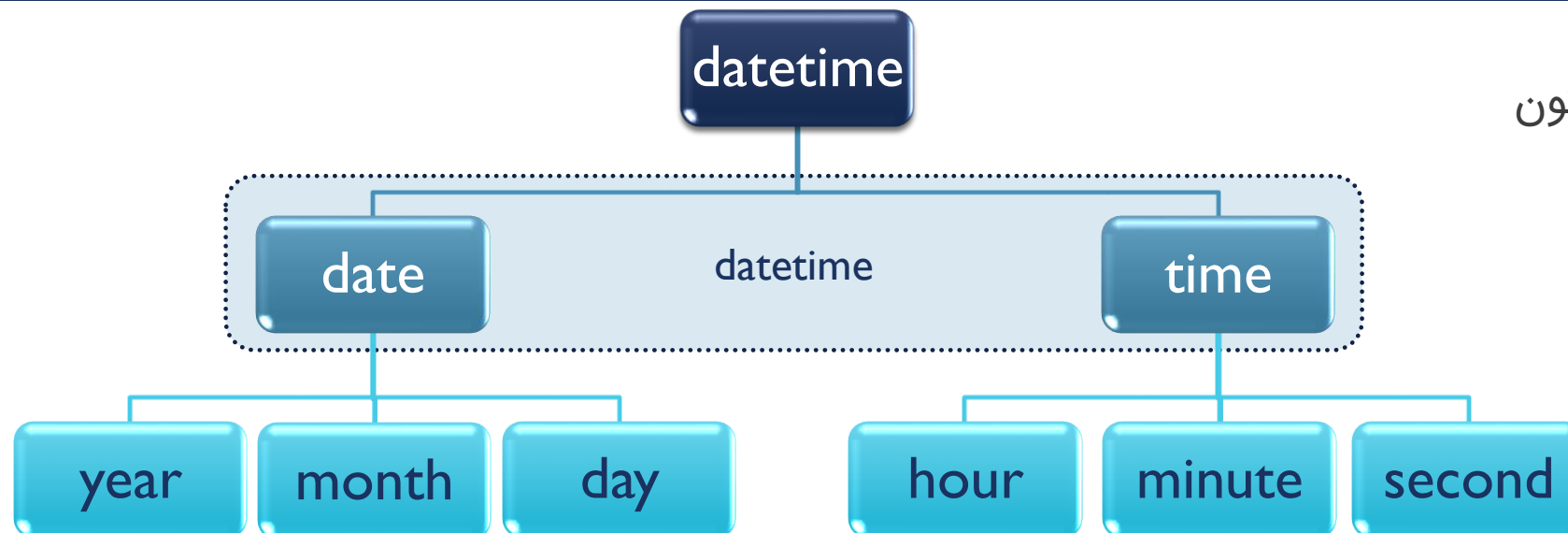
■ تابع حداقلی  
`def nothing():`  
`pass`

## 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, datetime
```

```
today = date.today()
```

```
today → datetime.date(2021, 11, 24)
```

```
now = datetime.now()
```

```
now → datetime.datetime(2021, 11, 24, 16, 45, 56, 974712)
```

تاریخ و زمان حال



- کار با تاریخ و زمان در پایتون

```
from datetime import date, datetime
```

```
d1 = date(2020, 1, 31)
```

*d1 → datetime.date(2020, 1, 31)*

```
d2 = datetime(2020, 1, 31, 13, 14, 31)
```

*d2 → datetime.datetime(2020, 1, 31, 13, 14, 31)*

تاریخ و زمان انتخابی

```
print(d1)
```

→ 2020-01-31

```
print(f'{d1.year}/{d1.month}/{d1.day}')
```

→ 2020/1/31

```
print(d2.strftime('%Y/%m/%d %H:%M:%S'))
```

→ 2020/01/31 13:14:31

```
print(d2.strftime('%c'))
```

→ Fri Jan 31 13:14:31 2020

نمایش تاریخ و زمان

- 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

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

کار با datetime

- کار با تاریخ و زمان شمسی در پایتون ← پکیج persiantools

> pip install persiantools

نصب در CMD

<https://pypi.org/project/persiantools/>

راهنمای استفاده

## 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'. Four Persian annotations with arrows point to specific parts of the interface: 'مسیر فایل اصلی پایتون' points to the script location field; 'انتخاب آیکن برنامه' points to the 'Icon' checkbox; 'انتخاب فایل‌های جانبی' points to the 'Additional Files' checkbox; and 'دستور نهایی کنسول' points to the 'Current Command' text field.

Auto Py To Exe

GitHub Help Post

Language: English

Script Location

C:/Users/Saeed/Desktop/test.py Browse

Onefile (--onedir / --onefile)

One Directory One File

Console Window (--console / --windowed)

Console Based Window Based (hide the console)

Icon (-iicon)

Additional Files (-add-data)

Advanced

Settings

Current Command

pyinstaller --noconfirm --onedir --console "C:/Users/Saeed/Desktop/test.py"

CONVERT .PY TO .EXE

مسیر فایل اصلی پایتون

انتخاب آیکن برنامه

انتخاب فایل‌های جانبی

دستور نهایی کنسول

■ 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