

۸. منظور از سطح زبان نامه نویسی چیست گزینه های زیر را باشد

۹. سطح زبان یک سطح ویژگی های الگوی زبان را باشد

۱۰. سطح آن (استئصال) می نماید یا به طور خلاصه نزدیک

۱۱. بودن به زبان انسان

۱۲. ۱) لغتی زبان های نویسی سطح بالا (High level)

۱۳. تابلیتی انتشار دارند .

۱۴. (شناخت مفاهیم خودرو و حیوان مفاهیم انتزاعی هستند که مقابل

۱۵. پیاده سازی شده بیست و در واقعیت الگو و طرح هستند

۱۶. بلک جزء های مقابل پیاده سازی مثل بنز و بردن)

۱۷. ۱۷

۱۸. - به زبان انسان نزدیک تر و از حیاتی بسیار درست

۱۹. ۱۸

۲۰. کارایی بسیار دارند

۲۱. صیریت حافظه و مدیریت مسیحیم بردازند را خودشان

۲۲. انجام می دهند . (فارغ از عملیات CPU)

۲۳. نمونه هایی از زبان های سطح بالا :

۲۴. ۲۴

۲۵. ۲۵

۲۶. ۲۶

۲۷. ۲۷

۲۸. ۲۸

۲۹. ۲۹

۳۰. ۳۰

۳۱. ۳۱

۳۲. ۳۲

۳۳. ۳۳

۳۴. ۳۴

۳۵. ۳۵

۳۶. ۳۶

۳۷. ۳۷

۳۸. ۳۸

۳۹. ۳۹

۴۰. ۴۰

۴۱. ۴۱

۴۲. ۴۲

۴۳. ۴۳

۴۴. ۴۴

۴۵. ۴۵

۴۶. ۴۶

۴۷. ۴۷

۴۸. ۴۸

۴۹. ۴۹

۵۰. ۵۰

۵۱. ۵۱

۵۲. ۵۲

۵۳. ۵۳

۵۴. ۵۴

۵۵. ۵۵

۵۶. ۵۶

۵۷. ۵۷

۵۸. ۵۸

۵۹. ۵۹

۶۰. ۶۰

۶۱. ۶۱

۶۲. ۶۲

۶۳. ۶۳

۶۴. ۶۴

۶۵. ۶۵

۶۶. ۶۶

۶۷. ۶۷

۶۸. ۶۸

۶۹. ۶۹

۷۰. ۷۰

۷۱. ۷۱

۷۲. ۷۲

۷۳. ۷۳

۷۴. ۷۴

۷۵. ۷۵

۷۶. ۷۶

۷۷. ۷۷

۷۸. ۷۸

۷۹. ۷۹

۸۰. ۸۰

۸۱. ۸۱

۸۲. ۸۲

۸۳. ۸۳

۸۴. ۸۴

۸۵. ۸۵

۸۶. ۸۶

۸۷. ۸۷

۸۸. ۸۸

۸۹. ۸۹

۹۰. ۹۰

۹۱. ۹۱

۹۲. ۹۲

۹۳. ۹۳

۹۴. ۹۴

۹۵. ۹۵

۹۶. ۹۶

۹۷. ۹۷

۹۸. ۹۸

۹۹. ۹۹

۱۰۰. ۱۰۰

۱۰۱. ۱۰۱

۱۰۲. ۱۰۲

۱۰۳. ۱۰۳

۱۰۴. ۱۰۴

۱۰۵. ۱۰۵

۱۰۶. ۱۰۶

۱۰۷. ۱۰۷

۱۰۸. ۱۰۸

۱۰۹. ۱۰۹

۱۱۰. ۱۱۰

۱۱۱. ۱۱۱

۱۱۲. ۱۱۲

۱۱۳. ۱۱۳

۱۱۴. ۱۱۴

۱۱۵. ۱۱۵

۱۱۶. ۱۱۶

۱۱۷. ۱۱۷

۱۱۸. ۱۱۸

۱۱۹. ۱۱۹

۱۲۰. ۱۲۰

۱۲۱. ۱۲۱

۱۲۲. ۱۲۲

۱۲۳. ۱۲۳

۱۲۴. ۱۲۴

۱۲۵. ۱۲۵

۱۲۶. ۱۲۶

۱۲۷. ۱۲۷

۱۲۸. ۱۲۸

۱۲۹. ۱۲۹

۱۳۰. ۱۳۰

۱۳۱. ۱۳۱

۱۳۲. ۱۳۲

۱۳۳. ۱۳۳

۱۳۴. ۱۳۴

۱۳۵. ۱۳۵

۱۳۶. ۱۳۶

۱۳۷. ۱۳۷

۱۳۸. ۱۳۸

۱۳۹. ۱۳۹

۱۴۰. ۱۴۰

۱۴۱. ۱۴۱

۱۴۲. ۱۴۲

۱۴۳. ۱۴۳

۱۴۴. ۱۴۴

۱۴۵. ۱۴۵

۱۴۶. ۱۴۶

۱۴۷. ۱۴۷

۱۴۸. ۱۴۸

۱۴۹. ۱۴۹

۱۵۰. ۱۵۰

۱۵۱. ۱۵۱

۱۵۲. ۱۵۲

۱۵۳. ۱۵۳

۱۵۴. ۱۵۴

۱۵۵. ۱۵۵

۱۵۶. ۱۵۶

۱۵۷. ۱۵۷

۱۵۸. ۱۵۸

۱۵۹. ۱۵۹

۱۶۰. ۱۶۰

۱۶۱. ۱۶۱

۱۶۲. ۱۶۲

۱۶۳. ۱۶۳

۱۶۴. ۱۶۴

۱۶۵. ۱۶۵

۱۶۶. ۱۶۶

۱۶۷. ۱۶۷

۱۶۸. ۱۶۸

۱۶۹. ۱۶۹

۱۷۰. ۱۷۰

۱۷۱. ۱۷۱

۱۷۲. ۱۷۲

۱۷۳. ۱۷۳

۱۷۴. ۱۷۴

۱۷۵. ۱۷۵

۱۷۶. ۱۷۶

۱۷۷. ۱۷۷

۱۷۸. ۱۷۸

۱۷۹. ۱۷۹

۱۸۰. ۱۸۰

۱۸۱. ۱۸۱

۱۸۲. ۱۸۲

۱۸۳. ۱۸۳

۱۸۴. ۱۸۴

۱۸۵. ۱۸۵

۱۸۶. ۱۸۶

۱۸۷. ۱۸۷

۱۸۸. ۱۸۸

۱۸۹. ۱۸۹

۱۹۰. ۱۹۰

۱۹۱. ۱۹۱

۱۹۲. ۱۹۲

۱۹۳. ۱۹۳

۱۹۴. ۱۹۴

۱۹۵. ۱۹۵

۱۹۶. ۱۹۶

۱۹۷. ۱۹۷

۱۹۸. ۱۹۸

۱۹۹. ۱۹۹

۲۰۰. ۲۰۰

۲۰۱. ۲۰۱

۲۰۲. ۲۰۲

۲۰۳. ۲۰۳

۲۰۴. ۲۰۴

۲۰۵. ۲۰۵

۲۰۶. ۲۰۶

۲۰۷. ۲۰۷

۲۰۸. ۲۰۸

۲۰۹. ۲۰۹

۲۱۰. ۲۱۰

۲۱۱. ۲۱۱

۲۱۲. ۲۱۲

۲۱۳. ۲۱۳

۲۱۴. ۲۱۴

۲۱۵. ۲۱۵

۲۱۶. ۲۱۶

۲۱۷. ۲۱۷

۲۱۸. ۲۱۸

۲۱۹. ۲۱۹

۲۲۰. ۲۲۰

۲۲۱. ۲۲۱

۲۲۲. ۲۲۲

۲۲۳. ۲۲۳

۲۲۴. ۲۲۴

۲۲۵. ۲۲۵

۲۲۶. ۲۲۶

۲۲۷. ۲۲۷

۲۲۸. ۲۲۸

۲۲۹. ۲۲۹

۲۳۰. ۲۳۰

۲۳۱. ۲۳۱

۲۳۲. ۲۳۲

۲۳۳. ۲۳۳

۲۳۴. ۲۳۴

۲۳۵. ۲۳۵

۲۳۶. ۲۳۶

۲۳۷. ۲۳۷

۲۳۸. ۲۳۸

۲۳۹. ۲۳۹

۲۴۰. ۲۴۰

۲۴۱. ۲۴۱

۲۴۲. ۲۴۲

۲۴۳. ۲۴۳

۲۴۴. ۲۴۴

۲۴۵. ۲۴۵

۲۴۶. ۲۴۶

۲۴۷. ۲۴۷

۲۴۸. ۲۴۸

۲۴۹. ۲۴۹

۲۵۰. ۲۵۰

۲۵۱. ۲۵۱

۲۵۲. ۲۵۲

۲۵۳. ۲۵۳

۲۵۴. ۲۵۴

۲۵۵. ۲۵۵

۲۵۶. ۲۵۶

۲۵۷. ۲۵۷

۲۵۸. ۲۵۸

۲۵۹. ۲۵۹

۲۶۰. ۲۶۰

۲۶۱. ۲۶۱

۲۶۲. ۲۶۲

۲۶۳. ۲۶۳

۲۶۴. ۲۶۴

۲۶۵. ۲۶۵

۲۶۶. ۲۶۶

۲۶۷. ۲۶۷

۲۶۸. ۲۶۸

۲۶۹. ۲۶۹

۲۷۰. ۲۷۰

۲۷۱. ۲۷۱

۲۷۲. ۲۷۲

۲۷۳. ۲۷۳

۲۷۴. ۲۷۴

۲۷۵. ۲۷۵

۲۷۶. ۲۷۶

۲۷۷. ۲۷۷

۲۷۸. ۲۷۸

۲۷۹. ۲۷۹

۲۸۰. ۲۸۰

۲۸۱. ۲۸۱

۲۸۲. ۲۸۲

۲۸۳. ۲۸۳

۲۸۴. ۲۸۴

۲۸۵. ۲۸۵

۲۸۶. ۲۸۶

۲۸۷. ۲۸۷

۲۸۸. ۲۸۸

۲۸۹. ۲۸۹

۲۹۰. ۲۹۰

۲۹۱. ۲۹۱

۲۹۲. ۲۹۲

۲۹۳. ۲۹۳

۲۹۴. ۲۹۴

۲۹۵. ۲۹۵

۲۹۶. ۲۹۶

۲۹۷. ۲۹۷

۲۹۸. ۲۹۸

۲۹۹. ۲۹۹

۳۰۰. ۳۰۰

۳۰۱. ۳۰۱

۳۰۲. ۳۰۲

۳۰۳. ۳۰۳

۳۰۴. ۳۰۴

۳۰۵. ۳۰۵

۳۰۶. ۳۰۶

۳۰۷. ۳۰۷

۳۰۸. ۳۰۸

۳۰۹. ۳۰۹

۳۱۰. ۳۱۰

۳۱۱. ۳۱۱

۳۱۲. ۳۱۲

۳۱۳. ۳۱۳

۳۱۴. ۳۱۴

۳۱۵. ۳۱۵

۳۱۶. ۳۱۶

۳۱۷. ۳۱۷

۳۱۸. ۳۱۸

۳۱۹. ۳۱۹

۳۲۰. ۳۲۰

۳۲۱. ۳۲۱

۳۲۲. ۳۲۲

۳۲۳. ۳۲۳

۳۲۴. ۳۲۴

۳۲۵. ۳۲۵

۳۲۶. ۳۲۶

۳۲۷. ۳۲۷

۳۲۸. ۳۲۸

۳۲۹. ۳۲۹

۳۳۰. ۳۳۰

۳۳۱. ۳۳۱

۳۳۲. ۳۳۲

۳۳۳. ۳۳۳

۳۳۴. ۳۳۴

۳۳۵. ۳۳۵

۳۳۶. ۳۳۶

۳۳۷. ۳۳۷

۳۳۸. ۳۳۸

۳۳۹. ۳۳۹

۳۴۰. ۳۴۰

۳۴۱. ۳۴۱

۳۴۲. ۳۴۲

۳۴۳. ۳۴۳

۳۴۴. ۳۴۴

۳۴۵. ۳۴۵

۳۴۶. ۳۴۶

۳۴۷. ۳۴۷

۳۴۸. ۳۴۸

۳۴۹. ۳۴۹

۳۵۰. ۳۵۰

۳۵۱. ۳۵۱

۳۵۲. ۳۵۲

۳۵۳. ۳۵۳

۳۵۴. ۳۵۴

۳۵۵. ۳۵۵

۳۵۶. ۳۵۶

۳۵۷. ۳۵۷

۳۵۸. ۳۵۸

۳۵۹. ۳۵۹

۳۶۰. ۳۶۰

۳۶۱. ۳۶۱

۳۶۲. ۳۶۲

۳۶۳. ۳۶۳

۳۶۴. ۳۶۴

۳۶۵. ۳۶۵

۳۶۶. ۳۶۶

۳۶۷. ۳۶۷

۳۶۸. ۳۶۸

۳۶۹. ۳۶۹

۳۷۰. ۳۷۰

۳۷۱. ۳۷۱

۳۷۲. ۳۷۲

۳۷۳. ۳۷۳

۳۷۴. ۳۷۴

۳۷۵. ۳۷۵

۳۷۶. ۳۷۶

۳۷۷. ۳۷۷

۳۷۸. ۳۷۸

۳۷۹. ۳۷۹

۳۸۰. ۳۸۰

۳۸۱. ۳۸۱

۳۸۲. ۳۸۲

۳۸۳. ۳۸۳

۳۸۴. ۳۸۴

۳۸۵. ۳۸۵

۳۸۶. ۳۸۶

۳۸۷. ۳۸۷

۳۸۸. ۳۸۸

۳۸۹. ۳۸۹

۳۹۰. ۳۹۰

۳۹۱. ۳۹۱

۳۹۲. ۳۹۲

۳۹۳. ۳۹۳

۳۹۴. ۳۹۴

۳۹۵. ۳۹۵

۳۹۶. ۳۹۶

۳۹۷. ۳۹۷

۳۹۸. ۳۹۸

۳۹۹. ۳۹۹

۴۰۰. ۴۰۰

۴۰۱. ۴۰۱

۴۰۲. ۴۰۲

۴۰۳. ۴۰۳

۴۰۴. ۴۰۴

۴۰۵. ۴۰۵

۴۰۶. ۴۰۶

۴۰۷. ۴۰۷

۴۰۸. ۴۰۸

۴۰۹. ۴۰۹

۴۱۰. ۴۱۰

۴۱۱. ۴۱۱

۴۱۲. ۴۱۲

۴۱۳. ۴۱۳

۴۱۴. ۴۱۴

۴۱۵. ۴۱۵

۴۱۶. ۴۱۶

۴۱۷. ۴۱۷

۴۱۸. ۴۱۸

۴۱۹. ۴۱۹

۴۲۰. ۴۲۰

۴۲۱. ۴۲۱

۴۲۲. ۴۲۲

۴۲۳. ۴۲۳

۴۲۴. ۴۲۴

۴۲۵. ۴۲۵

۴۲۶. ۴۲۶

۴۲۷. ۴۲۷

۴۲۸. ۴۲۸

۴۲۹. ۴۲۹

۴۳۰. ۴۳۰

۴۳۱. ۴۳۱

۴۳۲. ۴۳۲

۴۳۳. ۴۳۳

۴۳۴. ۴۳۴

۴۳۵. ۴۳۵

۴۳۶. ۴۳۶

۴۳۷. ۴۳۷

۴۳۸. ۴۳۸

۴۳۹. ۴۳۹

۴۴۰. ۴۴۰

۴۴۱. ۴۴۱

۴۴۲. ۴۴۲

۴۴۳. ۴۴۳

۴۴۴. ۴۴۴

۴۴۵. ۴۴۵

۴۴۶. ۴۴۶

۴۴۷. ۴۴۷

۴۴۸. ۴۴۸

۴۴۹. ۴۴۹

۴۵۰. ۴۵۰

۴۵۱. ۴۵۱

۴۵۲. ۴۵۲

۴۵۳. ۴۵۳

۴۵۴. ۴۵۴

۴۵۵. ۴۵۵

۴۵۶. ۴۵۶

۴۵۷. ۴۵۷

۴۵۸. ۴۵۸

۴۵۹. ۴۵۹

۴۶۰. ۴۶۰

۴۶۱. ۴۶۱

۴۶۲. ۴۶۲

۴۶۳. ۴۶۳

۴۶۴. ۴۶۴

۴۶۵. ۴۶۵

۴۶۶. ۴۶۶

۴۶۷. ۴۶۷

۴۶۸. ۴۶۸

۴۶۹. ۴۶۹

۴۷۰. ۴۷۰

۴۷۱. ۴۷۱

۴۷۲. ۴۷۲

۴۷۳. ۴۷۳

۴۷۴. ۴۷۴

۴۷۵. ۴۷۵

۴۷۶. ۴۷۶

۴۷۷. ۴۷۷

۴۷۸. ۴۷۸

۴۷۹. ۴۷۹

۴۸۰. ۴۸۰

۴۸۱. ۴۸۱

۴۸۲. ۴۸۲

۴۸۳. ۴۸۳

۴۸۴. ۴۸۴

۴۸۵. ۴۸۵

۴۸۶. ۴۸۶

۴۸۷. ۴۸۷

۴۸۸. ۴۸۸

۴۸۹. ۴۸۹

۴۹۰. ۴۹۰

۴۹۱. ۴۹۱

۴۹۲. ۴۹۲

۴۹۳. ۴۹۳

۴۹۴. ۴۹۴

۴۹۵. ۴۹۵

۴۹۶. ۴۹۶

۴۹۷. ۴۹۷

۴۹۸. ۴۹۸

۴۹۹. ۴۹۹

۵۰۰. ۵۰۰

۵۰۱. ۵۰۱

۵۰۲. ۵۰۲

۵۰۳. ۵۰۳

۵۰۴. ۵۰۴

۵۰۵. ۵۰۵

۵۰۶. ۵۰۶

۵۰۷. ۵۰۷

۵۰۸. ۵۰۸

۵۰۹. ۵۰۹

۵۱۰. ۵۱۰

۵۱۱. ۵۱۱

۵۱۲. ۵۱۲

۵۱۳. ۵۱۳

۵۱۴. ۵۱۴

۵۱۵. ۵۱۵

۵۱۶. ۵۱۶

۵۱۷. ۵۱۷

۵۱۸. ۵۱۸

۵۱۹. ۵۱۹

۵۲۰. ۵۲۰

۵۲۱. ۵۲۱

۵۲۲. ۵۲۲

۵۲۳. ۵۲۳

۵۲۴. ۵۲۴

۵۲۵. ۵۲۵

۵۲۶. ۵۲۶

۵۲۷. ۵۲۷

۵۲۸. ۵۲۸

۵۲۹. ۵۲۹

۵۳۰. ۵۳۰

۵۳۱. ۵۳۱

۵۳۲. ۵۳۲

۵۳۳. ۵۳۳

۵۳۴. ۵۳۴

۵۳۵. ۵۳۵

۵۳۶. ۵۳۶

۵۳۷. ۵۳۷

۵۳۸. ۵۳۸

۵۳۹. ۵۳۹

۵۴۰. ۵۴۰

۵۴۱. ۵۴۱

۵۴۲. ۵۴۲

۵۴۳. ۵۴۳

۵۴۴. ۵۴۴

۵۴۵. ۵۴۵

۵۴۶. ۵۴۶

۵۴۷. ۵۴۷

۵۴۸. ۵۴۸

۵۴۹. ۵۴۹

۵۵۰. ۵۵۰

۵۵۱. ۵۵۱

۵۵۲. ۵۵۲

۵۵۳. ۵۵۳

۵۵۴. ۵۵۴

۵۵۵. ۵۵۵

۵۵۶. ۵۵۶

۵۵۷. ۵۵۷

۵۵۸. ۵۵۸

۵۵۹. ۵۵۹

۵۶۰. ۵۶۰

۵۶۱. ۵۶۱

۵۶۲. ۵۶۲

۵۶۳. ۵۶۳

۵۶۴. ۵۶۴

۵۶۵. ۵۶۵

۵۶۶. ۵۶۶

- ۸ زبان و محایب زبان سطح بالا
- ۹ یادیگاری ساده تر
- ۱۰ مناسب برای لیم و دلب و ...
- ۱۱ امنیت بالاتر

۱۲ سمعت لست \leftarrow (زبان ماسن) همانند زبان مادری می‌باشد

۱۳ نسل سنت اند حافظه و سنت اضطرار

۱۴ نامناسب بـ(برای نوشتن از نسل ویندوز، و ...)

۱۵ کـ(زبان های برنامه‌نویسی سطح پایین) (low level)

۱۶ قابلیت ایجاد نوادرانه

۱۷ توسط مائیکن خوانده می‌شود و به زبان انسان نزدیک نیست.

۱۸ نیاز به صریحیت حافظه دارد. روابط استردی سخت افول

۱۹ و قابلیت انتقال سنت اند ریلر نیست)

۲۰ نویزه های از زبان سطح پایین

۲۱ کـ(زبان ماشین) (Machine language)

۲۲ اصفهان نوع عدد صفر (خاموش) و یک (روشن) بـ(بـ(

۲۳ نویزه استورات دارم اسیفابه می‌شود

۲۴ لـ(زبان اسیبلر) (Assembler)

* (زبان باینری)

۷

پندارشیده
فروردين

۱۳۹۹

26 March 2020
۱ شعبان ۱۴۴۱

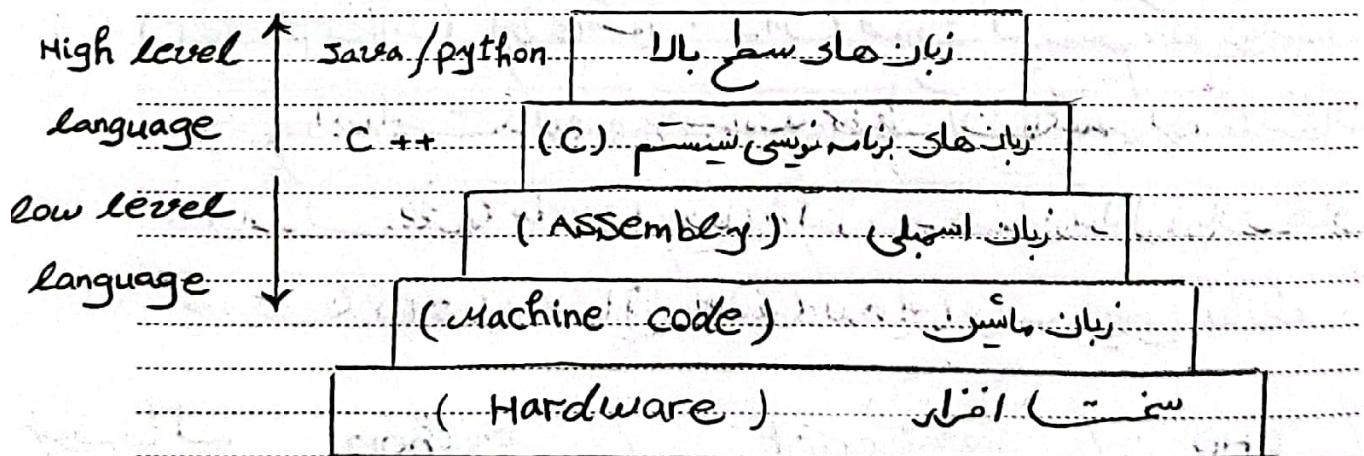
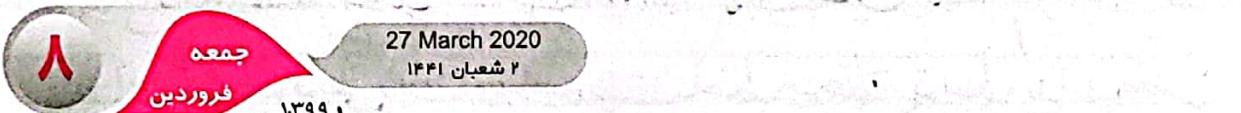
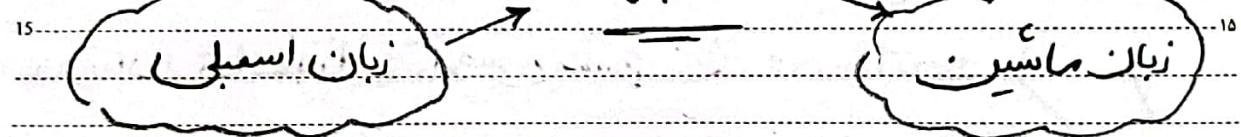
روز هنرهای نمایشی

لهم برای آنکه به فرمیم زبان مائین احتجوی کاری لند، ابتدا در ۱۰
لینم لئه پردازنه ها چلوت، کاری لند. ← اساساً یک
بریازنده از میلیون ها سویس بسیار لوچ بـ لـ مـ تـ وـ تـ خـ اـ مـ وـ مـ
و روئـنـ اـ سـونـدـ تـسـلـیـلـ سـهـ آـنـهـ: حـالـ بـ خـامـوـشـ وـ روـئـنـ
لـرـنـاـ بـرـخـیـ اـزـ سـوـیـسـ هـاـ خـاصـ) بـ مـ تـ وـ تـ خـ اـ مـ وـ مـ
تاـ کـلـرـهـاـ خـاصـ رـاـ اـجـبـاـمـ رـهـدـ، بـیـانـ مـائـیـنـ اـیـنـ اـمـلـانـ رـاـ بـ
سـمـاعـیـ دـهـدـ تـاـ باـ ① وـ ② (اـیـنـ سـوـیـسـ هـاـ خـامـوـشـ وـ روـئـنـ لـهـنـدـ .

* زبان اسپلی ← هدف از این زبان برقراری ارتباط

لـحـاتـ اـتـرـ باـ بـرـیـازـنـدـ نـسـتـ تـاـ بـهـ زـبـانـ مـائـیـنـ

14..... 15..... 16.....



+ تفاوت های زبان های (تفسیر) و کامپایلری :

۸. چالوئی اجر اسکن
۹. خطای بات

۱۰. وابستگی به سیستم عامل

۱۱. سرعت و میزان استفاده از حافظه و پردازنه

* زبان های برنامه نویسی استاتیک تایپ : ①

۱۲. زبان های statically Typed در این نوع متغیرها در زمان کامپایل شدن (compile time) بزماء مخصوص ای لرد

۱۳. در این نوع زبان ها زبانی که دلخواه، باید بازگشتی باشد

۱۴. متغیر در نظر می گیرد، دلیل عادی ب تخصیص آن نوکره بود

۱۵. جمله در این نوع زبان ها تایپ به متغیر اختصاصی یابد

۱۶. نه مقدار در نظر گرفته شده برای آن متغیر در صورتی که

۱۷. متغیر مایبی صورت دارد، با السیئش مواجب خواهیم شد.

۱۸. int $x = 5$ / str $y = Hello$ /

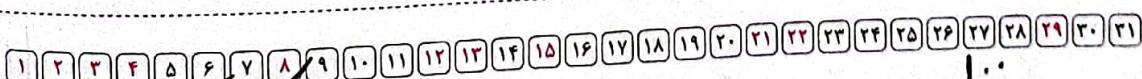
۱۹. $x = 5$ x ، $y = Hello$ x

* زبان های برنامه نویسی دینامیک تایپ : ②

۲۰. زبان های Dynamically Typed در این نوع متغیرها

۲۱. چنین اجرای برنامه (Runtime) (مشخص ای شود و دلخواه)

۲۲. در چنین کوئی نیازی به مشخص درون یستایی متغیر نوکره راست



۲۳. نتند : منظور از Type نوع داده ای است که ما ان سروکار داریم

* زبان های برنامه نویسی تایپ موقی : ③

زبان های strong Type، روی کاری نهایی خواهد بود.

روی متغیرها و داره انجام بین احساس و هر کار روی

اجازه نمی دهد انجام بین اتفاقات متغیر نمی نمایند.

لی نیست ارجح نماید :

$\text{int } x = 5$

$\text{str } g = \text{Hello}$

$z = x + g$ X

* زبان های برنامه نویسی تایپ فحیف : ④

زبان های weak Type بر علس تایپ موقی

حساسیت از ایجاد نشانه نمی دارد اثرا اجازه می دهد هر

کار روی متغیرها انجام بین انجام

$\text{int } x = 5$

$\text{str } g = \text{Hello}$

$z = x + g = 5\text{Hello}$

strong

Ruby ↑

python

Dynamic ← → static

Perl PHP C

Java Script C ++

weak

* پارامیم های برنامه نویسی

هر زبان برنامه نویسی به این اجرای جمیع ادوار از روشها استفاده می کند که با آن لغت شناسی زبان برنامه نویسی توانید پارامیم های برنامه نویسی پایانده به این لغتش سنجی اسست در تعیین دلیل پارامیم های برنامه نویسی یک نویلر خاص برای حل مسئله با استفاده از بحث از زبان های برنامه نویسی / همچنین می توان نقش روئی بله حل یک مسئله با استفاده از ابتدا و تبدیل های افراد اخیر و در دسترس مانند داریز.

۰ ۱ پارامیم های برنامه نویسی دستوری

(imperative programming paradigm)

/ لبه برنامه نویسی پرسه ای

(procedural programming)

/ لبه برنامه نویسی شی لرا

(object oriented programming / oop)

/ لبه برنامه نویسی رایانشی (موارد)

(parallel processing)

۷ پارامتریم‌های برنامه‌نویسی اعلانی (اخباری)

(declarative programming paradigm)

۸ کل ← برنامه‌نویسی منطقی

(logical programming)

۹ کل ← برنامه‌نویسی تابعی

(functional programming)

۱۰ کل ← برنامه‌نویسی پایگاه داده

(Database programming)

نصب و پیاده سازی

روز ملی فناوری هسته‌ای - روز هنر انقلاب اسلامی (سالروز شهادت سید مرتضی اوبنی - ۱۳۷۲ هش)

8 April 2020
۱۴۹۱ شعبان ۱۳

چهارشنبه

۲۰

فروزگان

۱۲۹۹

۸

آشنایی با محیط‌های کدنی پایه‌ولت

۱

۹

۱ نصب IDE خود پاپ‌تول

10

www.python.org نصب نصب از سایت note pad /

۲

لهم برات این کار ابتدا یک نایل تخف txt ایجاد کردم

سپس پسوندات را به .py تغییر داده

نتهی از راه نیز می‌توانیم به پسوند یک نایل رسماً پیدا کرد :

13 (This pc) → لذت بر منیو view در قسمت بالا

14 File name extensions ←

۳

خط فرمات و ترمیمال

15 ابتدا لهم برات دسترسی به خط فرمات (command prompt)

OK ← type: cmd ← [Enter] + [R] ۱

[D] search cmd ۲

16 سپس برات حالت تعاملی ← تایپ python

17 و برات اجرای نایل name ← آدرس نایل python

18 نتنهی می‌توان به جای cmd از powershell

19 نز استفاده لدم لهم برات دسترسی : open

Power shell ← Shift + ۱۰ ← راست کلید

20 windows ← ۱۱ ← دسترسی

21 در ویندوز open in windows Terminal ← دسترسی

۸ IDE (Integrated Development Environment) ۴

PyCharm

jetbrains.com کے قابل نصب از سایت

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

33.

34.

35.

36.

37.

38.

39.

40.

41.

42.

43.

44.

45.

46.

47.

48.

49.

50.

51.

52.

53.

54.

55.

56.

57.

58.

59.

60.

61.

62.

63.

64.

65.

66.

67.

68.

69.

70.

71.

72.

73.

74.

75.

76.

77.

78.

79.

80.

81.

82.

83.

84.

85.

86.

87.

88.

89.

90.

91.

92.

93.

94.

95.

96.

97.

98.

99.

100.

101.

102.

103.

104.

105.

106.

107.

108.

109.

110.

111.

112.

113.

114.

115.

116.

117.

118.

119.

120.

121.

122.

123.

124.

125.

126.

127.

128.

129.

130.

131.

132.

133.

134.

135.

136.

137.

138.

139.

140.

141.

142.

143.

144.

145.

146.

147.

148.

149.

150.

151.

152.

153.

154.

155.

156.

157.

158.

159.

160.

161.

162.

163.

164.

165.

166.

167.

168.

169.

170.

171.

172.

173.

174.

175.

176.

177.

178.

179.

180.

181.

182.

183.

184.

185.

186.

187.

188.

189.

190.

191.

192.

193.

194.

195.

196.

197.

198.

199.

200.

201.

202.

203.

204.

205.

206.

207.

208.

209.

210.

211.

212.

213.

214.

215.

216.

217.

218.

219.

220.

221.

222.

223.

224.

225.

226.

227.

228.

229.

230.

231.

232.

233.

234.

235.

236.

237.

238.

239.

240.

241.

242.

243.

244.

245.

246.

247.

248.

249.

250.

251.

252.

253.

254.

255.

256.

257.

258.

259.

260.

261.

262.

263.

264.

265.

266.

267.

268.

269.

270.

271.

272.

273.

274.

275.

276.

277.

278.

279.

280.

281.

282.

283.

284.

285.

286.

287.

288.

289.

290.

291.

292.

293.

294.

295.

296.

297.

298.

299.

300.

301.

302.

303.

304.

305.

306.

307.

308.

309.

310.

311.

312.

313.

314.

315.

316.

317.

318.

319.

320.

321.

322.

323.

324.

325.

326.

327.

328.

329.

٨ (Keymap) pycharm لیست های میانبر معمولی *

٩ zoom in / out نمود نمکار درون ۱ /

General ← Editor ← Settings

١٠ ✓ change font size with ctrl + mouse wheel

١١ Find in File ← Ctrl + F ۲ /

Replace in File ← Ctrl + R ۳ /

١٢ ⚡ search Everywhere ← Double Shift ۴ /

١٣ ⚡ Ctrl + Shift + Up ↑ + Down ↓ ۵ /

move statement up / down

١٤ Reformat code ← Ctrl + Alt + L ۶ /

١٥ Extend Selection ← Ctrl + W ۷ /

Shrink Selection ← Ctrl + Shift + W ۸ /

١٦ Duplicate current line ← Ctrl + D ۹ /

١٧ # comment with line ← Ctrl + / ۱۰ /

Basic code completion ← Ctrl + Space ۱۱ /

١٨ ⚡ show context Actions ← Alt + Enter ۱۲ /

١٩ ⚡ Optimize Imports ← Ctrl + Alt + O ۱۳ /

Quick Definition ← Ctrl + Shift + I ۱۴ /

٢٠ Quick Switch Scheme ← Ctrl + ~ ۱۵ /

١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩ ١٠ ١١ ١٢ ١٣ ١٤ ١٥ ١٦ ١٧ ١٨ ١٩ ٢٠ ٢١ ٢٢ ٢٣ ٢٤ ٢٥ ٢٦ ٢٧ ٢٨ ٢٩ ٣٠ ٣١

Find Action ... ← Ctrl + Shift + A ۱۶ /

* مجموعه Syntax ← مجموعه ای که قواعد است

که چنونی بزم اسنتکسی به یک زبان مخصوص را تعریف کند

برای نمونه این که یک متن چهار نوشته شود که توسط مفسر پایتون به عنوان توضیح در نظر گرفته شود یا نیز

شروع شده باشد، به عایت سینتیک تعریف شده در پایتون پرسنلی را و چنانچه مفسر برای این متن را با هم مقایسه و تعیین شده اک مطابقت دهد یا است که از این خواهد شد.

* سطوح (lines) :

① سطوحی فیزیکی physical lines

رواقعی همان سطوحی که توسط دیرایشدهای متن شماره زاری می شوند و توسط کاربر قابل تحریک

② سطوحی منطقی logical lines

سطوحی که برداشت مفسر از اجزای برمایند است

هر سطر منطقی بیانیل یک رسمت (statement) پاییز

نکته: یک سطر فیزیکی می توان شامل چند سطر منطقی باشد که در این حالت باید سطر منطقی (یا همان رسمت) توسط

این حالت باید سطر منطقی (یا همان رسمت) از یکیلیز جدا شود.

نحوه لکه (colon) (:) semicolon (؛) از یکیلیز جدا شود.

نحوه لکه (colon) (:) semicolon (؛) از یکیلیز جدا شود.

نحوه لکه (colon) (:) semicolon (؛) از یکیلیز جدا شود.

نحوه لکه (colon) (:) semicolon (؛) از یکیلیز جدا شود.

۱ سطر فیزیکی

نکته ② : گاهی نیز بحث است بارگیری خواهی بیشتر بیاید سطح منطقی
برقراری (چند سطح فیلی) پس از میان فاصله ایم : بارگیری شناسن
یک دارستور در چند سطح فیلی از کاراکتر ① (Back Slash) استفاده می کرد .

نمونه :

```

1 msg = " python programming \
2 language" # This is a message
3 print (msg)

```



نکته ③ : از \ (نیچه تول) برای شناسن سطح توضیح (comment)
استفاده می کرد و همینی نیچه تول پس از آن صحیح لون
توضیحی درج کرد .

نکته ④ : سترهای شامل {}, [], () هستند را
با توان بروز استفاده از \ , شست و در قالب
چند سطح فیلی نوشت) و برخلاف استفاده از \ میتوان
یعنی از شناسن سطحها comment ، اضافه کرد .

نمونه :

```

1 month_names = ['Januari',
2     'April', 'Mei', # Dutch names
3     'Juli', 'Augustus', # for the months
4     'oktober', 'November'] # of the year

```



Blank lines

③ سطوحای خالی

سطوحای که تنها شامل فضای خالی (Space / Tab)

باشد و توسط مفسن ناییده لفسته می شود و باید که

تجمیع نماید از این سطوحای اتوان برای

خوانایی) استفاده نرود.

- 1
- 2
- 3
- 4

نمونه :

return a ** b :

print power (2,3)



* مستندسازی (Documentation) *



استفاده از امکانات همچون ارائه توضیح در دادهای تولید

در کد و موالی خوابایی هرچه بسیار دارای برآمدگرد مراجعت

اکینده برآمدگویی و افاده ریلی که می خواهد برای توسعه این

فعال باشند یا از آن استفاده نمایند بسیار مفید خواهد بود.

④ توضیح (Comment) (جه طبع کاری کند)

یعنی comment در زبان پایتون توسط کاتر (#)

آنرا و با پایان سطر متنی که دعم یا این که پنیره توضیح ها

نیز مانند سطوحای خالی توسط مفسن ناییده لفسته می شود و ب

هدف از نوشت توضیح میانبین لدها، سخن منطق کد است.
راهنمایی کد مورد نظر چهل نوشته شده و چه کاری انجام گیرد.
نکت ①: گاهی نیز اندیشه توضیح (comment) برای غیرفعال
کرد که هایر کرفته می شود:

نکت ② comment: در پایتون سیا به سلسله تک سطحی
تعریف می شود و باید بین توضیح هایی با بسته از
یک سطر فریلی باید توجه داشت که هر سطر به صورت
 جداگانه می باشد با # آغاز شود.
(→ space)

نمونه کد:

1	# This is a comment, it helps you to understand
2	# - the code later on
3	#
4	# - Anything that comes after the ...
5	
6	print (" I could have code like this ") -- # ...

نکته: بسته‌ها (packages) را سه باشد Docstring

برای این منظور Docstring باید درون ماثل py--init-- نوشته شود.

۲۸

پنجشنبه
فروردین

16 April 2020
۱۴۴۱ ۲۳ شعبان

۱۳۹۹

(جی کاری لند؟)

Docstring

املاک دیگر را پسوند برای این توضیح بیشتر درباره درهای

برنامه است، این متن توسط سه علامت نقل قول

۱۰ " " " & " ") شروع و پایان می‌باید.

۱۱ (مجموعاً از آن به عنوان نمی‌شوند، رسود مائل، کلاس،

۱۲ تابع و متده استفاده می‌شود.)

۱۳ Docstring توسط منسنا دیده گرفت (نحوه رسود) و در زمان

۱۴ اجرانیز با استفاده از صفت doc قابل رسایابی

۱۵ نکته ①: مخاطب اصلی Docstring ها کاربران است از آن دو

۱۶ مبوط به آن استفاده می‌لند. بنابراین باید به سچ چهارمی

۱۷ استفاده از درها بپردازند.

۱۸ print (help (Name code)) → Docstring نهاده نمایش

۲۹

جمعه
فروردین

17 April 2020
۱۴۴۱ ۲۴ شعبان

۱۳۹۹

روز ارتش جمهوری اسلامی و نیروی زمینی

نمونه کرد:

۱ def complex (real = ۰ . ۰ , imag = ۰ . ۰) :

۲ (۱۱) Form a complex number

۳

۴ Keyword arguments :

۵

۶ real -- the real part (default ۰ . ۰)

۷

(۱۱)

۸ print (complex . -- doc --) → اجرا رسایاب در زمان

۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

* توزیعی (Indentation) *

بلک بندی اور زبان پایتون) توسط فونٹلی سطرها مسٹنچ

جی لرد؛ این عمل اور زبان ھائی مانند C و Java

توسط آکو ولاد ({ }) انجام می ہے۔

توزیعی در واقع عبارت است از میان فضائی خالی (Space & Tab)

هر دستور انہیں سطر فریلیپ خور

تمام دستور ھائی موجود در میں بلک بندی باستعمال میان

خاصیت نہیں ہے سرکاریں خور توزیعی رائے دے باشند۔

لئے : C : Python

if x > y : if (x > y) :

x = 1 ; x = 1 ;

y = 2 ; y = 2 ;

}

نمونہ مثال :

Block 1

(Header)

items = [5 , 8 , 1 , 4]

def bubble - sort (seq) :

Block 2

l = len (seq)

for i in range (l) :

Block 3

seq [n - 1] = seq [n]

return seq

print bubble - sort (items)

Block 2

Block 1

1 ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

* متغیر (variable) *

یک متغیر درست را بگوییم که این نام خوبی است (۱) بگوییم می باشد در حافظه، می باشد که مقادیر در آن ذخیره شده است.

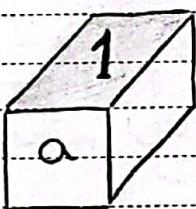
نحوه کد C :

a = 2

int b = a

رسانید int a = 1 بیان می کند که می باشد از حافظه بنا نام a برای انگاری اعداد صحیح (integers) است لفظ شود و مقدار ۱ در آن ذخیره شود. از این پس

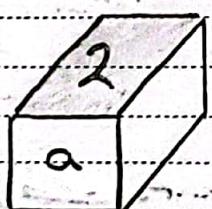
متغیر a معنی این نقطه از حافظه است (مانند جمعه).



در اینجا - حاوی مقدار ۱ است

در ادامه دستور a = 2 باعث شود مقدار

پیشین متغیر a حذف (از جعبه خارج) و مقدار جدید



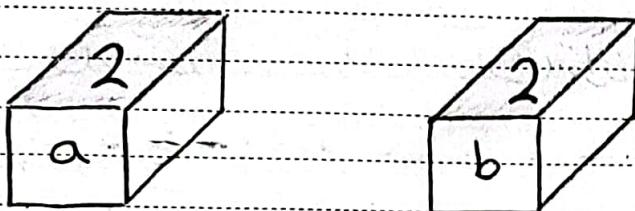
یعنی ۲ در آن صار را درست.

رسانید int b = a در ابتدا باعث

ایجاد شد که حل جدید در حافظه بنا نام b می شود و از نوع

همان اعداد صحیح پس مقدار درون متغیر a را درون آن

لپتوبی کی ای دن نوں یو محلہ برائے نکوداری (نوع) را دے int
در حافظہ موجود ہی باسٹہ لئے ہر دو حاوی مقدار 2 ہی باشد۔



نکتہ: در این استاذ زبان‌ها، نوع (type) توسط متغیر تعین
ہی لرد و تلاس (برائے قدر داری نوع دارہ ریکارڈ) بے غیر ان int
در متغیر a (مانند 3, 7 یا "string") باعث کا بروز
خطا زمان کامیابی ہی لرد۔

وچکاری پیشوند: یہ متغیر جنہی نہست خرید نام نہ بھیلے شخص
در حافظہ ارجاع (ریا اسدارہ) دارو۔ تعریف متغیر در پیشوند
سیار سادہ تر اسٹراؤنگ پا ایساپ (Assign) شی
بھیلے نام الجاد ہی لرد۔

نہاد (=)، عملہ (operator)، انساب (assignment) اسٹا۔

نکتہ: در تعریف متغیر پیشوند بخلاف اپنے در C مسائده لریم
ریانج بہ تعین نوع برائے آن نہیں تھا (زوراً int a) X
حال type از نوع شی تعین ہی لرد و یہ متغیر در
طول زمان احری می تواند یہ شی ہائی از انواع متفاوت ارجاع

۱

$$a = 1$$

۲

$$a = 2$$

۳

$$b = a$$

منس بک رسیدن به دستور $a = 1$ سلم پاسینا را اینجا :

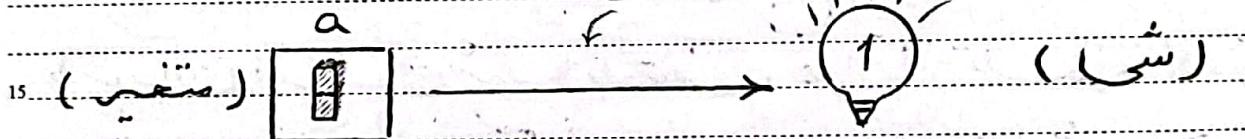
۱ یک شی از نوع اعداد صحیح و مقادیر ۱ را در جای از حافظه

ایجاد می‌نماییم (نوع توسط شی تغییراتی لرد و ۱ عدد صحیح)

۲ نی متغیر (ریاضی) را در جای ایجاد می‌نماییم که از حافظه اینداد می‌نماییم (ابتدا صورت از قبل ایجاد نشده باشد)

۳ یک پیوند از متغیر a به شی ۱ برقرار می‌نماییم که به این

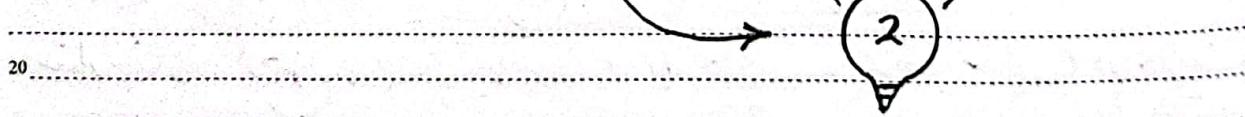
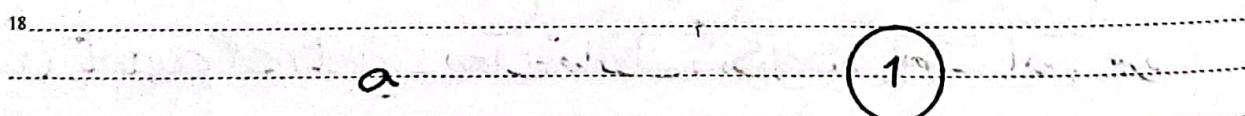
پیوند، ارجاع (Reference) نفت می‌شود.



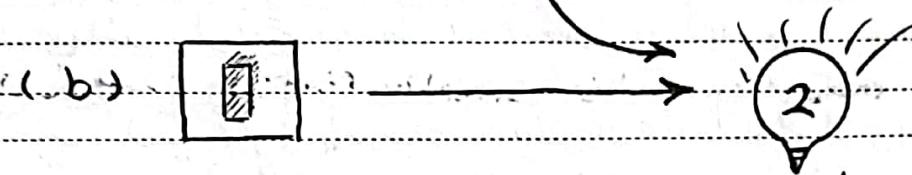
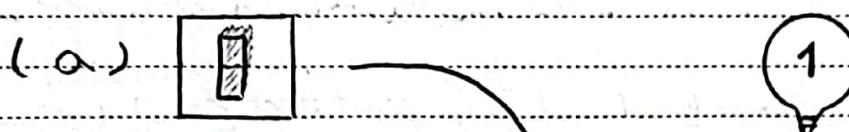
نکته: انساسب شی دیگر (از هر نوع دیگر) به یک متغیر موجود

باشد حذف ارجاع قبلی آن و ارجاع به شی جدید می‌شود

(دستور دوم)

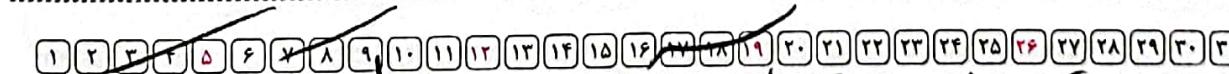


رسورسوم $a = b$ می تولد : یعنی متغیر جدید باتا b ایجاد نردد و به همان شی ایس ارجاع راسکت باشند لہ متغیر a ارجاع دار ہے۔



نکتہ ۱: هر سی شامل یہ سمارنده ارجاع (Reference Counter) ہے۔ یعنی این صورت میں درہر لمحہ، تعداد ارجاع ہا کے ان شی را نسبان می دھد دو۔ وباہر ارجاع جدید بہتی ہے، یعنی واحدہ ان اضافی بھی سووڑ و باحذف ارجاع یعنی واحدہ کا گھس ری پاید، چنانچہ مقدار ان بھی صفر برسد، شی ان تو سطح تلنیں Garbage collection ہے۔ مقدار و مقدار حافظہ ای کے تو سطح سی مصنف سوہ بود آزادی نردد۔ برائی مسماہدہ تعداد ارجاع ہا بیسے ہی تباہ از نارجی میں موقوف ہے getrefcount() دوست مائل کریں۔

نکتہ ۲: البتہ مفسر پاسیون اعداد صحیح و سنتہ ہائی لو جل را ہس انہیں مقدار سمارنده ارجاع بھی صفر برسد، باہذف صرف جویں درہرین ہے۔



ایجاد این اسیا برائی استفادہ در اینہے، ان حافظہ یال بھی لئے۔

- انساب چند گانه :

نکته ۱ : امثال این مثال چند متغیر با انساب چند گانه در پاسخ دارد و جواب دو گانه چند متغیر همی بیلسته با ارجاع دارند را نهاده باشد رسیده اینجا لرد :

$a = b = c = "python"$ نمونه کده :

نکته ۲ : با انساب اسایه صفاتیت چیزی است که بدل (comma) و ترکیب عامل انساب (=) استفاده نماییم

(تصدیق ناصل نوچه که عامل بین مساوی باشد)

$a, b, c = 2, 4, 5, "python"$ نمونه کده :

نکته ۳ : یکی از کاربردهای انساب چند گانه، جایگان (اسایه دو متغیر را به سادگی) باشد سطح لرد باشد یک عوض لرد :

$a = 1$ نمونه کده :

شکست حمله تمامی آمریکا به ایران در طیین (۳۵۹ هش

$b = 2$

$a, b = b, a$

- ثابت (constant)

به متغیری لفظی انتووند که مقادیر انت همواره بایست و بس ان تعیین

دیگر امثال تخصیص مقدار آن وجود ندارد نمونه کده `sara` :

`final int HOURS = 24;`

هر لونه تلاس برخی تعیین با خطاطا مواجهی مسود بس ان آن

۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

در پاییول اسلامی برخی تعیین بایست بخواسته

البت، الک تھیں ناپریں بولن مختصر ھا بریس ان اھمیت لے
عی تھاں پر ملائیں جاوی لد زیر ایجاد نہیں ہے:

```

1 # File: constant.py
2 # path: /home/user/Documents/MyModule
3
4 class const:
5
6     class constError(TypeError): pass
7
8     def __setattr__(self, name, value):
9         if name in self.__dict__:
10            raise self.constError("can't rebind const \\" + name + "\")
11            self.__dict__[name] = value
12
13 import sys
14
15 sys.modules["__name__"] = const()

```

(درپ بے لاس کونسٹ نیاز بے مطالعہ دیس مریوط بے
شی رائی و استئناھا را لارو۔)

یا پیوند یک زبان حساس به حرف ← case sensitive



* شناسه (Identifier) :

نامه نهادین که به دلخواه کاربر تطبیق و ازان کن برای تشخیصی
متغیرها، توابع، کلاس‌ها، مانعدها یا دلیل اسناد
باستفاده از یکی‌دیگر استفاده می‌شود.

انواع شناسه در یا پیوند نهادی دارد:

۱) تبا بایکی از حروف بندهک یا لوچک لفبا (انگلیسی)

— (Underscore) یا کالات (a..z & A..Z)

letter : lowercase | uppercase سرعی می‌شود.

۲) در ادامه می‌آید چنین لفبا (انگلیسی)، کالات — و

۳) عدد (0..9) با هر ترتیب آورده شود.

۴) هم معموری در طول شناسه وجود ندارد.

۵) identifier = (letter | "-") (letter | digit | "-")

لهم در تعاریف regex

۶) ۱: استفاده از کالات های خاص صافند، !، @، #، %، &، ^، <، >، .، _، - استفاده از مجاز نیست.

۷) استفاده از فاصله (space) مجاز نیست.

۸) استفاده از خط تیره (Hyphen) برای جباسازی نموده ممکن

در پایتون از تابع `name mangling` استفاده می شود.
توسط این تابع و نامهای انتساب ساخته شده، نفس خاصی به آن دارد. می شود.

۱. سازه حشوی مائل

الرئاسه یک کالر با اگان سود (رونه بیان نیز) مثل ← Name

۲. لئے سازه حشوی لاس

الرئاسه یک کالر با دو اگان سود (رونه بیان نیز) مثل ← Name

نکته: جدا از این صور در پایتون صفت‌ها (Attributes) و متدهای خاصی وجود دارد. از پس تعریف لئے و براحتی معرف مفهوم خاصی دارد. سازه این صفت‌ها و متدها با doc مثل ← Name بیان می شود.

المات لیک (keywords) :

نکته پایانی در مورد شناسه ها این است که نباید تواند یک شناسه با برابری باشد های لیک پاسخون انتها را بفرماید. های لیک لیک را واقع شناسه هایی هستند که از پیش بلای مفسر پاسخون تعریف شده و معنای منحصر به فرد آن دارند.

فرض است این های لیک از پاسخون توسط لرزید قابل

help ("keywords") نمایش است :

نکته ① : تابع iskeyword() بررسی کند که آیا این دокумент در راستایی از های لیک لیک باشد یا نه که در صورت تائید مقدار True را بازیگیراند.

۱ import keyword نمونه کد :

۲ print (keyword.iskeyword("en"))

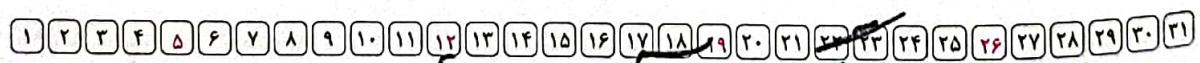
نکته ② بلای نمایش تعداد المات لیک باشد لرزید :

۱ import keyword

۲ print (len(keyword.kwlist))

تابع len ، تعداد اعضا یک شی را بازیگیراند.

نکته ③ نهاد المات لیک None / False / True



با حروف بزرگ امتازی سود .

* * عملکردها (operator) *

به نهادی لفته‌ی سود به عمل متناسب با بررسی اشیاء به انجام می‌رساند. همین به اشیایی که عملکر بررسی آن‌ها عمل را به انجام می‌رسانند، عملوند (operand) لفته‌ی انجام می‌رساند.

۱) عملکردهای حسابی (Arithmetic operators)

جمع (Addition) $+$

تففیق (Subtraction) $-$

ضرب (Multiplication) $*$

تقسیم (Division) $/$

توان (Exponent) $**$

تقسیم در شوه پایین (Floor Division) $//$

لے حذف مقادیر اعشاری از حاصل جواب

باقي‌مانده (Modulus) $\%$

۲) عملکردهای منطقی (logical operators)

و (and) بے زیانی که هر دو عملوند آن اینش True داشتند

یا (or) بے زیانی که هر دو عملوند آن اینش False داشتند

انش عملوند خود را بر علسن افی لند (not)

(درستورات شفی کاربرد دارند.)

جواب هر کدام از عملهای زیر با True / False

۱۱

پنجمین
اردیبهشت

30 April 2020

۱۴۴۱ رجب

۱۳۹۹

۳ عملهای مقایسه Comparison operators

(Equal) برابر $= =$ $= =$ (not Equal) نابرابر $! =$ $! =$ (Greater Than) بزرگتر از $>$ $>$ (Less Than) کوچکتر از $<$ $<$ (Greater Than or Equal) بزرگتر مساوی $> =$ $> =$ (Less Than or Equal) کوچکتر مساوی $< =$ $< =$

۴ عملهای انتساب Assignment operators

 $\% =$ $/ =$ $* =$ $- =$ $+ =$ $=$ $x += y \neq x = x + y$ شرط $// =$ $** =$

این عملهای ابتدا عمل مربوط به عمل حساب را برآورده

مقدار عاملهای اول و دوم خود انجام و سپس حاصل را بر عاملونه

۱۲

پنجمین
اردیبهشت

1 May 2020

۱۴۴۱ رجب

۱۳۹۹

سنت حسانست می‌باشد

۵ عملهای عضویت Membership operators

از آن باید بین دو عبارت مقدار در میان اعضای دنباله

sequence : لیست، رشته، ...) استفاده می‌شود.

True

 وجود دارد \rightarrow اگر مقدار عاملهای سنت حسان است باید in وجود ندارد \rightarrow اگر مقدار عاملهای سنت حسان نباشد باید $not in$ print ("p" in "python") \rightarrow True

۶ عملدهای همیتی Identity operators

از آن بلافاصله بررسی یکی بولن (رویسی) استفاده می‌شود.

الهر دو عملوند به یک شی ارجاع رائسته باشند، **True** is

الهر دو عملوند به یک شی ارجاع نرائسته باشند، **True** is not

نمونه کد: ✓

1 ~~1~~ $a = [1, 2]$

2 ~~2~~ $b = [1, 2]$

3 ~~3~~ $c = a$

4 $\text{print}(a \text{ is } b) \rightarrow \gg\gg \text{False}$

5 $\text{print}(a \text{ is } c) \rightarrow \gg\gg \text{True}$

۷ عملدهای بستی Bitwise operators

این رسته از عملدها، عمل مخصوصی را بر علاوه تابه

بیت‌های عملوند ها انجام می‌دهند. در پایه‌یون بلوچ استفاده

از این عملدها نوعی به تبدیل اعداد به پایه دو (دو دویی)

True False Binary (نیست)

نحوه عملهای اعداد دو دویی: $0 \leftarrow \text{حتمش} / 1 \leftarrow \text{بعضی}$

0 & 1	0 & 1	0 & 1	0 & 1	0 & 1	0 & 1	0 & 1	0 & 1	0 & 1
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
128	64	32	16	8	4	2	1	

a = 0b 0011

b = 0b 0010

مربوط به

مدار منطقی

>>> 3

>>> 2

۱۴

پیشنهاد
ارزیابی

3 May 2020
۱ رمضان ۱۴۴۱

۱۳۹۹

نکته ۲ : در زیر پایتون اعداد پایه دو جمجمه هی باشد تا باید

a = 0b 0011 شروع شوند .

نکته ۳ : از تابع bin() هی توان برخورده است آردن مقادیر

دو دویی یک عدد ده ریاضی استفاده کرد : البته این تابع

مقادیر دو دویی را در مقابل بگشتن (نوع string) برخی برداشت .

همینی بر حاست اعمالی یا بیرون باواره لرن اعداد دو دویی ،

خروجی ده ریاضی بگستایی آید .

print (bin(10)) >>> 0b1010

$\begin{array}{r} + \\ 2^3 \quad 2^2 \\ \hline \end{array}$

نکته ۴ : بیتی ← ترتیبی هایی از خروجی آنها

&

خواهد بود که هر دویی متناظر از عملوند های آن

print (a & b) ۱ باشند

>>> 2 a = 0000 0011

b = 0000 0010

And &

2

←

0000 0010

بیتی ← ترتیبی هایی از خروجی آنها

1

خواهد بود که هر دویی متناظر از عملوند های آن ۰ باشند

print (a | b) a = 0000 0011

>>> 3

b = 0000 0010

OR ۱

1 ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

3

← 0000 0011

8 XOR بینی کے تنہائیت ہای اور جو جو ان $\underline{1}$

9 خواهد بود لہ هر دو بینی متناظر ان عکلوند ہای ان مخالف

10 $\boxed{11}$ Print (a^b) \rightarrow پرینٹ بینی

11 $a = 0000\ 0011$

$b = 0000\ 0010$

12 $\begin{array}{r} \textcircled{1} \\ \xleftarrow{\quad} \\ 0000\ 0001 \end{array} \quad \underline{\text{XOR}}$

13 \ll شیفت چپ (left shift) بینی

14 عملوند سمت چپ را به مقادیر عکلوند سمت راست خود بھی سمت

15 جسیں جابجا کی لئے \ll (عملان ہای روسو) با \ll مقادیر دھی کی سور

16 $\boxed{11}$ print ($a \ll 3$) $a = 0000\ 0011 \quad \text{LSH}$

17 $\gg 24 \quad \begin{array}{r} \textcircled{24} \\ \xleftarrow{\quad} \\ 00011\ 000 \end{array} \quad \ll$

18 \ll (Right shift) راست \gg شیفت راست

19 بینی های عکلوند سمت چپ را به مقادیر عکلوند سمت راست خود بھی سمت

20 جسیں جابجا کی لئے \gg (عملان ہای روسو)

21 $\boxed{11}$ print ($a \gg 3$) $a = 0101\ 1000 = 88 \quad \text{RSR}$

22 $\gg 11 \quad \begin{array}{r} \textcircled{11} \\ \xleftarrow{\quad} \\ 00001\ 011 \end{array} \quad \gg$

نکت ۱: بیتی \rightarrow (اتاک عملوند) هر یک راز

بیت‌های عملوند خود را از ۰ به ۱ و برعکس تبدیل می‌کند.

نکت ۲: سی سیو بیان می‌کند اعداد علامت‌دار بود وی (منفی) :

$$a = 0000\ 1001 \quad 9$$

نمایش

$$\text{اولین عدد از سمت چپ را به عنوان} \quad 9 = 1000\ 1001 \quad \text{مقدار}$$

علامت \leftarrow علامت در نظر بگیرید ($0 \leftarrow 1$ و $+ \leftarrow 0$)

نتیجه می‌گیریم $-9 = 1111\ 0110$

آورده (همه ۰ تبدیل به ۱ و برعکس)

نکت ۳: همانند صنعت می‌باشد ابتدا عمل کرد و

ولی از راست تا اولین یک را رسماً نامذکور و بقیه را عوض

نکت ۴: مفهوم باستون بصورت بیس امیس اعداد علامت‌دار را به سیو

تابع ململ دو Two's complement اینگاهی می‌کند

~~print (~a)~~ $a = 0000\ 0011 = 3$

$$-4 = 1111\ 1100 = -4$$

$$\text{Two's complement } \uparrow \quad 0000\ 0100 = 4$$

نکت ۵: می‌توان مقدار a را با برابر حاصل $-a - 1$ در نظر گرفت.

۸. walrus operator: ⑧ عملکر شیء ماهی

۹. از نسخہ ۳.۰.۸ پاپیون پس محتوا جدید بنام

۱۰. بے سینٹس پاسوں Assignment Expressions

۱۱. اضافہ سرو است. نہایک این عملکر بے سلسل =

۱۲. کی باشد (تسبیہ دو جسم و عاج یک شیء ماهی)

۱۳. وہ ماین املاں را میں دھکہ نہ عملیات انساب و بازی روان

۱۴. مقدار تابع صورت ہمنماں ہے انجام برسائیم :

۱۵. `print("walrus = False")` >>> Error X

۱۶. `print(walrus := False)` >>> False ✓

(ھمانند C++)

۱۷. متغیر walrus با انساب مقدار False ایجاد و

۱۸. سیسٹم نے تابع print اسیاری لورد (کابربر بیشتر در لیست و حلقات)

۱۹. operator precedence

۲۰. جنگیں، عبارت شامل ہیں عملکر باشد، اینکے اعتبار عمل

۲۱. کرامیں بھی اسود، در حاصل تیجہ تائیں لذار خواهد بود.

۲۲. هر دام از عملکرها ان دید منس پاسوں را لکھ حق نقدم معاون

۲۳. استا (مائندہ پلز ریاضیاتی برنامہ روپی)

۱۸

- 1 پرانتز * Grouping
- 2 f() * تابع Function call
- 3 [index]: index] Slicing
- 4 [] Array Subscription
- 5 ** * توان Exponential
- 6 ~ Bitwise NOT
- 7 + - unary plus / minus
- 8 * / // %
- 9 + - Addition / Subtraction
- 10 << >>
- 11 & ^ |=
- 12 == != < >

۱۹

- 13 < <= > >=
- 14 = == /= // = % = *= + =
- 15 is is not
- 16 in not in
- 17 not and or
- 18 Lambda Lambda Expression

۸ دستور (statement) *

۹ واحدی از کد است که شامل کم های (لیست) بوده ، اجرا

۱۰ می کند و کاری را به ابعاد می رساند . در پایتون ۲ نوع دستور وجود

۱۱ ۱ simple statements

۱۲ دستورهایی که تنها در یک سطر منطقی بسازده ساخته می شود

۱۳ مانند دستور import / دستور انتساب / ارجاعی تابع و ...

۱۴ ۲ compound statements

۱۵ لوهی از دستورها به می توانند یک بخشی (مانند دستور

۱۶ if - تعیین شابع) یا چند بخشی (مانند دستور شرط if def

۱۷ باشند . هر بخش شامل یک سرآئیند (header) و یک

۱۸ بینه (suite) است . هر سرآئیند با یک لرد

۱۹ لیست آگهان و با یک : (colon) پایان می یابد .

۲۰) بینه بس از سرآئیند و بار عایت توزعیلی بینه نسبت به آن)

۲۱

۲۲

۲۳

۲۴

۲۵

۲۶

۲۷

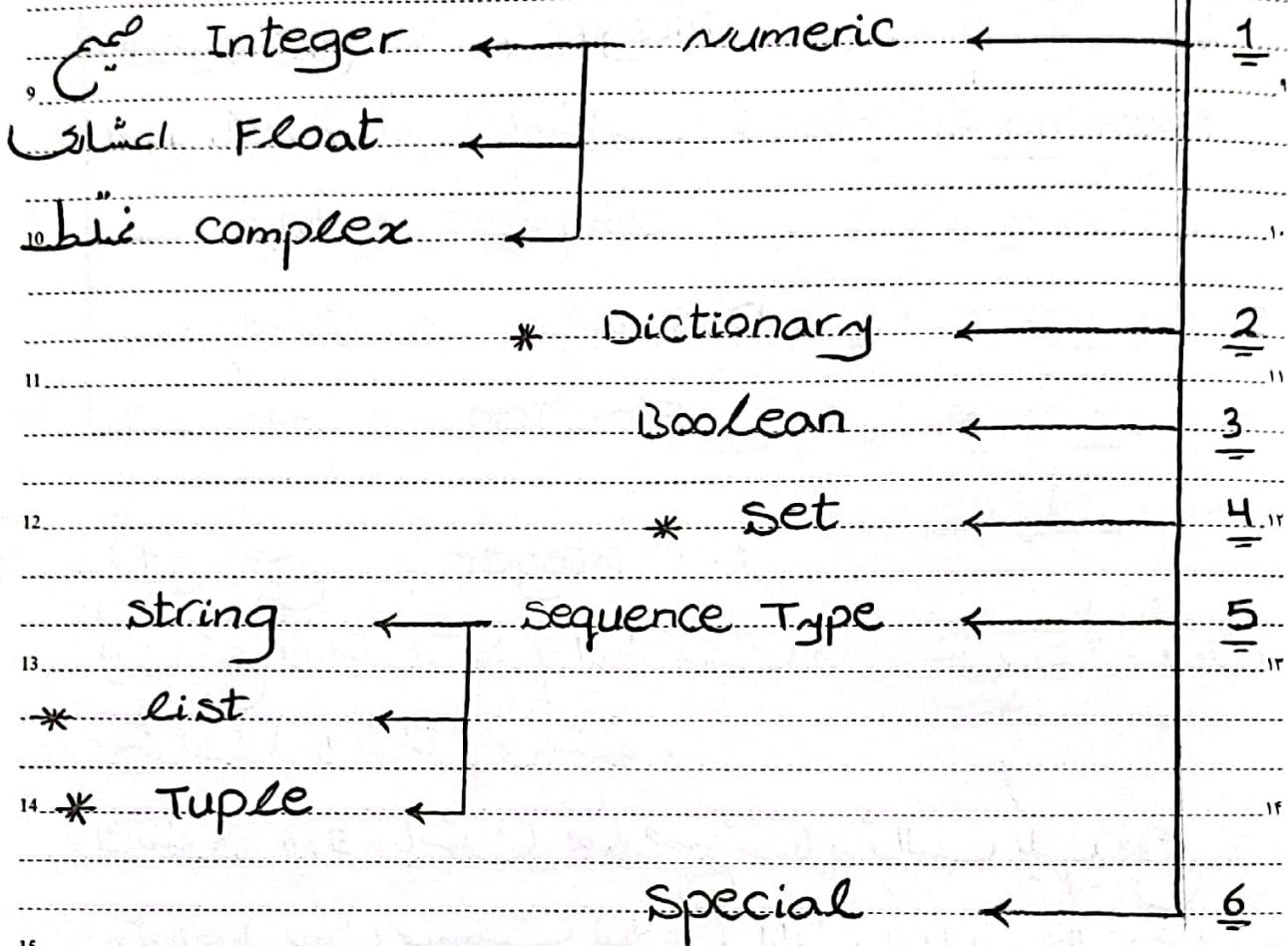
۲۸

۲۹

۳۰

۳۱

python Data Types



پایتون هر نوع را توسط یک کلاس (class) ایجاد می‌کند.

برای این کلاس یک نمونه یا یک شی از کلاسی

شخص است.

هر چند بزرگ نویس نمی‌تواند با تعریف کلاس، نوع دفعه ای

خود را راست می‌سازد و تنها درباره آن یعنی از انواع را دره

یا انواع شی ای که به شکل آماده (Built-in) در اختیار

مفسر پایتون قرار را دره می‌شود صحبت ننمی‌کنم.

* انواع عددی (numeric Types) *

8. Integer صیغه

9. Floating-point (اعشاری) میز رسانید

10. complex مختلط

11. Decimal دسیمال

12. Fraction کسر

13. * صیغه (integer) ①

این نوع از اشیاء تمام اعداد مثبت \oplus و منفی \ominus ، بعنه

میز اعشار را شامل نمی‌شود.

آن بسطه ۰، ۰۳ پایتون اعداد صیغه ترتیب بر قرار یافته‌اند نوع

14. int (بعنه صدوبیت اندازه) ارائه می‌گردد و اسنون

در پایین همچوینی برای طول اعداد صیغه وجود ندارد، اما در

15. صورت رقمی احوال با استفاده از sys.maxsize

16. مقدار عریض برای بیان مقدار این رست از اعداد بوده است اور در

17. توجه: متنظر از اندازه نامحدود اعداد این است لئے اندازه این اشیاء

18. تنها محدود به میزان حافظه memory کار نموده و تا هر

19. اندازه ای حافظه در رسپس بکسری توانند ز محاسبات

20. رشید راسته باشند.

اعداد صحیح را می‌توان علاوه بر پایه ده (Base 10) از اینها نیز در مجموعه اعداد دهی داشت که مقدار ممکن است برابر باشد با عدد دهی تواند بیشتر باشد اما مقدار صحیح آن $0 \leq a < 9$ داشته باشد (در پایه دو Binary، در پایه هشت Octal، در پایه شانزده Hexadecimal) و پایه شانزده را $0 \leq a \leq F$ نیز می‌دانند و همچنین دو عدد دیگر نیز در نظر نمی‌گیرند.

* اعداد پایه دویی باستاد باینری آغاز می‌شوند و همچنان که در پایه دهی داشتیم اینها را $0B$ / ab خواهیم نامend.

$ab_{11} \rightarrow 3$ عدد ۳ می‌باشد.

* اعداد پایه هشتادی باستاد باینری آغاز می‌شوند و همچنان که در پایه دهی داشتیم اینها را 00 / 00 آغاز می‌نمودند.

$00\ 14_{12} \rightarrow 12$ عدد ۱۲ می‌باشد.

* اعداد پایه شانزدهی باستاد باینری آغاز می‌شوند و همچنان که در پایه دهی داشتیم اینها را $0x$ / $0x$ آغاز می‌نمودند.

$0xA5 \rightarrow 165$ عدد ۱۶۵ می‌باشد.

نوع سی صیغه در پایه های دویالوئی آنماهی انداره (int) و در پایه های سیستم معمولی برای مخصوص روزن آنها استفاده شده است. همین‌ها همان‌وارد روزن

این اسعار حالت تعاملی یا استفاده از تابع print، ریاضی دهنده نهائی داره‌ای سود

۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

نکته ۱: برای تبدیل یک عدد صحیح از پایه ده به هر یک از پایه‌ها می‌توان از تابع `int()` استفاده کرد. تبدیل به پایه دو \leftarrow `bin()`
 تبدیل به پایه هشت \leftarrow `oct()`
 تبدیل به پایه شانزده \leftarrow `hex()`
 + همان توجه که خوب باشد از این تابع‌ها به صورت یک رشته / String برگردانده می‌شود.

نکته ۲: برای برگرداندن یا اعداد صحیح به پایه ده می‌توان از لیس `int()` استفاده کرد. آرگومان‌های نمونه این لیس به صورت `int(str, base)` می‌باشد.
 آرگومان `str` \leftarrow یک رشته عربی یعنی یک عدد صحیح (دھریا) داخل نوارهای ^①quotation (نقل قول).
 آرگومان `base` \rightarrow یا که آن را مستعار می‌کند.

1	<code>a = 165</code>	نونهاده
2	<code>type(a)</code>	$\gg> <\text{class}'\text{int}'>$
3	<code>b = hex(a)</code>	$\gg> b = '\text{0xa5}'$
4	<code>type(b)</code>	$\gg> <\text{class}'\text{str}'>$
5	<code>int(b, 16)</code>	$\gg> 165$
6	<code>type(int(b, 16))</code>	$\gg> <\text{class}'\text{int}'>$

نکته ۳: می توان اعداد را بدون حرف مخصوص (النده یا یا) بلا ۵۶،

۰۰ و ۰۲) به این ناس ارسال ننماییم.

همین‌ها تبدیل نوع داشت عددی (رئیسان) به دونهای

بتوان آن را یک عدد ازیابی نمود.) برایم داد به عدد صحیح

استفاده کرد.

۱. $a = "56"$

۲. `type(a)` \rightarrow `<class 'str'>`

۳. `int(a, 10)` \rightarrow ۵۶

۴. `int(a)` \rightarrow ۵۶

۵. `type(int(a))` \rightarrow `<class 'int'>`

نکته ۴: int بدون ازولان یک سی‌پی‌اس از نوع صحیح بزی دراند.

۱. `int()` ۰

نکته ۵: برای برسی اوردن اندازه یا صیان حافظه لغزیده توسط

یک سی‌پی‌او وحدت بایت (Byte) می‌توان از تابع

`getsizeof()` در مارک کو استفاده نمایم.

*- مین شناور (Float) *

تمام اعداد مشتبه و منفی که شامل بیکمین اعشار هستند

در پایتون به صورت اسیاکی با نوع Float (محاذل)

نوع double در زبان C) الائچی شود .

جزئیات این نوع با استفاده از Sys.float_info

تبلیل مبنایه است .

نکتہ : کاچی بلک نمایش اعداد از سیو نماد علیقی scientific notation

استفاده می شود ؛ در پایتون اهم می توان از حرف E یا e

که محاذل فربند ۱۰ به توان می باشد استفاده کرد .

$$4e5 \rightarrow 4 \times 10^5$$

پایتون این اعداد را در مقابل مین شناور float ارائه می دهد .

نکتہ ۲ : می توان از لاس () float بلک تبدیل اعداد یا رشته های

عددی به یک شی مین شناور استفاده کرد .

۱

$a = 920$

۲

type (a) \rightarrow class 'int'

۳

float (a) \rightarrow a = 920.0

۴

float ("920")

۵

float ("3e + 2") \rightarrow 300.0

نکته ۳: float بعنوان ارثومان یک شی صفر از نوع میز سیاره است.

~~۱۱~~ float () ۰,۰ بانگسته.

نکته ۴: چنانچه عددی از نوع میز سیاره در ارثومان نداش ()
مقدار پرداز، تنها نیس صحیح عدد برگردانده بود.

نمونه کد: $a = 2.31$

۲ type (a) \rightarrow class 'float'

۳ int (a) \rightarrow ۲

۴ type (int (a)) \rightarrow class 'int'

نکته ۵: با استفاده از نلاس float () بتوانیم اسیاپی باشد.

های مثبت و منفی بی نهایت \leftarrow infinity & inf

NAN \leftarrow not a number \circlearrowleft تعریف نشده

ایجاد نهایتیم \leftarrow (دوستی NAN با یکدیگر برابر نیستند .)

نمونه کد: $a = float ('inf')$ \rightarrow inf

$b = float ('+infinity')$ \rightarrow +inf

$c = float ('nan')$ \rightarrow nan

$a == b$ \rightarrow True

$d = float ('nan')$

۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

False

نکت ⑥: بلاس بررسی اینقدر مقدار بیشتری بحث نهایت یا تعریف داشته باشد.

نیست) می‌توان به ترتیب از نابع‌های (۱) تا (۵) برآورد کرد.

دروز مأول استخراج نایسیم (isnan) math

```
a1 = float('inf')
```

```
2 | b = float('nan')
```

3

4. import math

5 math.isinf(a) => True

6. `math.isnan(b)` >> `True`

: (complex) bis - *

اعداد مختلط ان یا نسبتی $\frac{a+bi}{c+di}$ دو عدد حقیقی a و b از مجموعه \mathbb{R} و دو عدد مختلط c و d از مجموعه \mathbb{C} باشند.

موهومی Imaginary تسلیل شده‌اند. در باستون این

اعمار المثلث بایر، δ , Real part + Imaginary part,

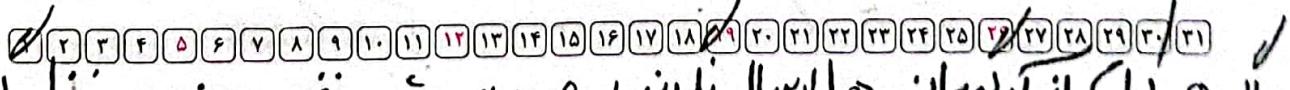
دارند له حرف گز نیسانلر واحد موهوچی است .

این امداد توسط اسپانی بانوں Complex ارائه می ہے۔

نستہ : انڈا لس (انڈا لس) complex میتوں اور ابجادیں سئی۔

استفاده از این مدرس الگوی ساخت $\text{Complex}(\text{real}, \text{imag})$

دالر ۲۰ نیش موشهوما بخش حقیق ۲۰



الرهن دار از آرایه ها ارسال نموده به صورت پیش فرض صف رئیس رفت.

۱

پنجشنبه
خرداد

۱۳۹۹

21 May 2020
۲۷ رمضان ۱۴۴۱

روز بزرگداشت ملاصدرا (صدر المتألهين)

نمونه کد:

$$1 \quad a = 3 + 2j$$

$$2 \quad b = 1 + 4j$$

۳ complex (a, b) $\rightarrow 3 + 2j + 4j$

۴ complex (3) $\rightarrow 3 + 0j$

نکته ۲: با استفاده از دو صفت `real` و `imag` میتوان بخش های

حقیقی و موهومی چنین یا `complex` را به دست آورد. بجز از

این که اعداد از جه نویسی را سهیل یاف نویسی مختلط شلت

که باشد، بخش های عدد مختلط به صورت مختلط شلت

آنلیل می‌گارد.

نمونه کد:

$$1 \quad a = 3 + 4j$$

2 $a.real \rightarrow 3.0$

3 $a.imag \rightarrow 4.0$

۲

جمعه
خرداد

۱۳۹۹

22 May 2020
۲۸ رمضان ۱۴۴۱

روز جهانی فقیس آخرین جمعه مبارک رمضان

نکته ۳: `complex()` میتواند عددی را به عدد مختلط تبدیل کن

به عدد مختلط را درست کاری توجه دست کر نماید داخل این

تئست هم مخفی خالی وجود داشت، باشد.

توجه: امثال موارد دست عددی (مختلط) یا خود یک عدد مختلط

که اندوکشن ندیس های `int()` و `float()` وجود ندارد و

۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

محب با بزرگ طایی سود.

- دسیمال (Decimal)



اسس صاریح این نوع بایک استفاده در موقعیتی است که خطاب نوع

مبنی شناور تابل نشست مانند توسعه برنامه حسابداری

در علوم کامپیوتر بایک این نوع مبنی شناور به کامپیویس از دلاری

Binary Floating-point اسکاراده می‌شود. این دلار

اعداد را به شکل ذیق ایجاد نمی‌دهد. به عنوان نمونه عدد ۰.۱

برابر با عدد زیر بیل است ۰.۱۰۰۰۰۰۰۰۰۰۰۰۰۰۰۱

شرطیت زاره می‌شود. هر چند که این عدد بسیار بزرگ باشد ۰.۱

نمی‌باشد اما به هر حال خوب آن نیست !!

نمونه کد :

$$1 \quad a = 0.1 + 0.1 + 0.1$$

$$2 \quad a = 0.3 \quad \Rightarrow \text{False}$$

$$3 \quad \text{print}(a) \quad \Rightarrow 0.3000000000000004$$

در پایتون نوع دسیمال با ایندادی از دلار

دلت ماثول decimal درست حواله بود.

خطه نداش له

from decimal import Decimal

2 a = Decimal('0.1')

3 print(a) $\Rightarrow 0.1$

4 type(a) $\Rightarrow <\text{class}'decimal.Decimal'>$

نحوہ کہ :]

1 import decimal

2 a = decimal.Decimal('0.1')

3 b = decimal.Decimal('0.3')

4 b = a + a + a ➤ True

بہ سوچیں ہے تو اسی لیٹریل میں ایجاد کریں :

a = Decimal(23)

b = Decimal('23.45')

c = Decimal("2345e-2")

Tuple e = Decimal((1, (2, 3, 4, 5), -2))

f = Decimal("infinity")

g = Decimal('NaN')

نوت ① : اذ اب تاں نوچ میں سارے وضیق نیست ، این اعداد را حتماً بہ

صورت رکھئے ہے Decimal اسال نہیں (b)

نوت ② : اعداد را جی توان بھروسہ یہ لیٹئی توپیل Tuple (ساختاری

مسئلہ : ((0, 0, 0, ...)) اسال کو (c)

توپیل صورت نظر باید ساختاری مسابہ الگوی زیر دلستہ باشد۔

(sign, digits, exponent)

لے میت (عدد صفر) و

بیانکر ہمات خود توپیل کو رقم ہائی سنی (عدد ۱) کا مشتمل توان دھیل را بیان کی لے



نسته ③: سیان رفت (Rounding) و عمل رفت (precision) اسیا، دیمال با استفاده از این context میتوانیم این رفت را در اختیار داشتیم؛ این این سیان یک سیان اطلاعات بخوبی را در اختیار داشیم، دیمال صریح هندی را در سیان به آن باید از تابع decimal مارفل getcontext() استفاده کرد.

تابع "getcontext()" اسیا، دیمال جایی برای ریاضی را فراهم میکند.

```

1 import decimal
2 a = decimal.Decimal('3.45623')
3 b = decimal.Decimal('0.12')
4 print(a+b)           => 3.57623
5 ctx = decimal.getcontext()
6 type(ctx)            => <class 'decimal.context'>
7 ctx.prec = 1          # رفت تایید نشود
8 print(a+b)           => 4
9 ctx.prec = 2          # رفت تایید نشود
10 print(a+b)           => 3.6
11 ctx.prec = 3          # رفت تایید نشود
12 print(a+b)           => 3.58
13 ctx.rounding = "ROUND-CEILING"
14 print(a+b)           => 3.59

```

در نمونه لد قبل می‌شاهد که می‌توان با استفاده از صفت context پیش از معرفت مفهومی را در متن ترتیب داد.

برای اینکه اعداد را در محدوده ۰ تا ۰.۵ بگردانیم باید طول خود را باز نمایند و این را با زیرنویس می‌کنند.

برای تنظیم این اعداد می‌توان از روش rounding استفاده کرد.

برای تنظیم این اعداد می‌توان از روش ROUND - HALF - EVEN استفاده کرد.

اطلاعات rounding می‌باشد:

۱۳. ROUND - CEILING

برای اعداد ۲.۱۹ از حدوده حذف می‌گردد و آنرا ۲.۲ می‌گردانند.

۱۴. ROUND - FLOOR

برای اعداد ۲.۱۹ از حدوده حذف می‌گردد و آنرا ۲.۱ می‌گردانند.

۱۵. ROUND - EVEN

برای اعداد ۲.۱۹ از حدوده حذف می‌گردد و آنرا ۲.۲ می‌گردانند.

۱۶. ROUND - HALF-EVEN

برای اعداد ۲.۱۹ از حدوده حذف می‌گردد و آنرا ۲.۱ می‌گردانند.

۱۷. ROUND - HALF-UP

برای اعداد ۲.۱۹ از حدوده حذف می‌گردد و آنرا ۲.۲ می‌گردانند.

۱۸. ROUND - HALF-DOWN

برای اعداد ۲.۱۹ از حدوده حذف می‌گردد و آنرا ۲.۱ می‌گردانند.

8 \rightarrow ROUND - Down 3
رور لورت به سمت صفر

9 برای اعداد + و - تنها ارقام خارج از محدوده حذف می‌گردد.

$$10 \quad 2.58 \rightarrow 2.5 / 2.19 \rightarrow 2.1 \quad 4$$

11 \rightarrow ROUND - UP .

12 برای اعداد + و - انتاک خارج از محدوده حذف و احیانی

13 رسم بائی مانده یک واحد از اینش می‌باشد

$$14 \quad -2.52 \rightarrow -2.6 / -2.19 \rightarrow -2.2 \quad 5$$

15 \leftarrow ROUND - HALF - DOWN .

16 نیم حذف شده نزدیک از 5 باشد به روئی

17 و در غیر این صورت ROUND - Down به روئی لرد

$$18 \quad 2.58 \rightarrow 2.6 / 2.55 \rightarrow 2.5 \quad 6$$

$$19 \quad -2.58 \rightarrow -2.6 / -2.55 \rightarrow -2.5 \quad 7$$

20 \leftarrow ROUND - HALF - UP .

21 نیم حذف شده نزدیک یا برابر 5 باشد به روئی

22 و در غیر این صورت ROUND - Down به روئی لرد

$$23 \quad 2.55 \rightarrow 2.6 / 2.51 \rightarrow 2.5 \quad 8$$

۷ ← همانند روئی ۵ است ROUND - HALF - EVEN .

وکی از صفاتی که رسم ایسلی یعنی حلق شده ۵ باید زیبار آن متفاوت است؛ در این حالت ال آخرین رقم باقیمانده نوج

باید به سیوه ROUND - DOWN و الی خر باید به

ROUND - UP که روی لرد .

۱۱ 2.68 → 2.7 / 2.65 → 2.6

۱۲ 2.75 → 2.8

۱۳ ← الی براساس روئی ROUND - 05 UP . ۸

۱۴ آخرین رقم باقیمانده ۰ یا ۵ باید به روئی ROUND - DOWN

و در غیر این صورت به همان سیوه ROUND - UP

۱۵ که روی لرد . ROUND - DOWN .

2.58 → 2.6 / 2.48 → 2.4

لسر (Fraction) *

این نوع بارگیری پیشیگار (Rational) اعداد بولیا در پایتون اساس شده و بارگارسی از نلاس Fraction درین مانع در سرس اصلی لید .

نمونه کد : import fractions

1. $a = 1$

2. $b = 2$

3. $f = \text{fractions.Fraction}(a, b)$

4. print(f) $\rightarrow 1/2$

5. type(f) $\rightarrow <\text{class 'fractions.Fraction'}$

روتیکی دیگر ایجاد نیست لسی :

از یک لسی مصیحتناور (بسته است این نوع بحالت رشت وارد شود)

print(Fraction('0.5')) $\rightarrow 1/2$

از یک لسی دسیمال

print(Fraction(Decimal('1.1'))) $\rightarrow 11/10$

از یک لسی رشت لسی (صورت و مخرج اسی بایست از نوع صحن باشند)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

print(Fraction('3/14')) $\rightarrow 3/14$

از یک سی ریلر

$$f_1 = \text{Fraction}(1, 2)$$

$$f_2 = \text{Fraction}(3, 5)$$

$$\text{Fraction}(f_1, f_2) \rightarrow \text{Fraction}(5, 6)$$

نکته ۱: با استفاده از صفت‌های numerator (نمایت)

و denominator (نماینده توان صورت)

و نتیجه‌سی نسرا دستیاب یافتا.

$$f = \text{Fraction}(1.5)$$

$$f \cdot \text{numerator} \rightarrow 3$$

$$f \cdot \text{denominator} \rightarrow 2$$

نکته ۲: از این نوع سی به مسائلی اچی توان در انواع محاسبات ریاضی

استفاده کرد

$$\text{Fraction}(5, 16) - \text{Fraction}(1, 4)$$

$$\rightarrow \text{Fraction}(1, 16)$$

$$\text{Fraction}(3, 16) / \text{Fraction}(1, 8)$$

$$\rightarrow \text{Fraction}(3, 2)$$

خالیه یک سی اصمم با یک سی نس افزوده شود \leftarrow حاصل سی نس
و اول یک سی این نشانور به یک سی نس افزوده شود \leftarrow حاصل سی نیز نشانور

* مثال ۱ : یک عدد از کارب رفته و ارقام آن را جدا کرده و سپس آن را چاپ کرده (با فضای اضافی بین اعداد ارقام)

نکته اول → باقی مانده هر عدد به 10 ، رقم آخر آن را بگیرید و در ده
نکته دوم ← بقیه هر شده به پایین هر عدد به 10 ، یک رقم از آن حذف کنید تا می‌شود .

1. $x = \text{int}(\text{input}("Enter x :"))$

2. $\text{temp} = x \% 10$

3. $\text{print}(\text{temp})$

4. $x = x // 10$

5. $\text{temp} = x \% 10$

6. $\text{print}(\text{temp})$

7. $x = x // 10$

8. $\text{temp} = x \% 10$

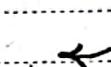
9. $\text{print}(\text{temp})$

18.

19.

20.

* نوع رشته (string) :

نوع رشته str در پایتون با قرار نفیضنا بینالا، اب از کارالتها
برخلاف یک جفت‌نوار نقل قول / quotation را تیزی (' ')
یا بعثای (" ") ایجاد می‌شود. به تفاوت با یکدیگر از نظر
نوع ندارند. 

در پایتون از خلاف برخ از زبان‌ها نوع کارالت را char

وجود ندارد؛ در این زبان یک کارالت چند جز یک رشت، با طول یک نمی‌باشد.

در پایتون می‌توان از نهادهای نقل قول در داخل یکدیگر نیز بهره برد.
در این سلسله ترتیبی پایست نهاد نقل قول داخلی با بیرون متفاوت باشد - چنان‌جواز خواهد بود از نهاد نقل قول بلسازی (سقف) ده نمایید باید از کارالت‌های Escape مانند \ باشد (در ادامه بررسی خواهد شد)

- نحوه کد:
- 1 a = " python "
 - 2 b = " python ' programming "
 - 3 c = " python " language
 - 4 d = " Mhdii # ۴۵۸۱ \$ Sbr "
 - 5

نکته ۱ از درس پیش با **Docstring** آشنایی شد. در کاربردی
 بیلی از سه نهاد فعل مغلق ("") یا ('') برای اینبار
 شئی رسته بین استفاده می‌شود. منیز این نوع رسته
 درین استفاده می‌توان آن را به سادگی ارجمند سطر و
 با هر سیل فردی مغلق نگوای نوشت.
 این موضع برخلاف که متصد استفاده از ددهای خاص بو
 مانند **HTML** را لارسی بسیار محدود خواهد بود.

* رسته به عنوان رنبار (sequence)

بخش از انواع شئی در پایتون به مانند رسته **string**
 تغیل **Tuple**، لیست **list** و ... با عنوان
 رنبار، ساخته می‌شوند.

رنبار در واقع یک رنبار از کاراکترهاست.

در پیشنهادی توان از این رنبار را با استفاده از
 اینیس (**Index**) موقعیت آن، رنبارها نموده اندیس
 اعضای از سرتاسری با عدد صفحه شروع او نیز هست اسید یک واحد
 یک واحد امر اینیس) می‌لردد.

P	y	t	h	o	n	.	s	t	r	i	n	g
0	1	2	3	4	5	6	7	8	9	10	11	12
-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

لذاری دستیابی به اعضای یک زبانه با $seq[i]$ از تلوی seq

نه نیز اندیس i عضو موجود نظر است استفاده می شود.

۱) تلوی $seq[-i]$ اعضای زبانه را نسبت راست به میانسایی از خد

۲) تلوی $[i : -i]$ اعضای از زبانه را به در بازه ای از

آندهست ن تأمین از اندیس i هستند راستیابی می کند.

۳) تلوی $[i : k]$ همانند قبلی با این تفاوت

نه ک اندیشه گام بیانس اعضا را تعیین می کند.

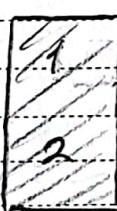
نکته ۱: با استفاده از تابع (len) می توان تعداد اعضای یک زبانه

را بدست آوردن (ترجیح از ۱)

نکته ۲: یک شی دست جزو انواع **immutable** یا بیانون است

ومقدار یا (اعضا زبانه) آن تغییر نمی تواند تغییر دارد.

برای این کار تنها می بایست یک شی جدید ایجاد کرد.



$a = \text{"python strings"}$



$a[6] = "-"$ \Rightarrow Error

عملّهای باقی ریشت:

با ریشت های نیز می توان از عملّهای + (باقی پسوند ریشت ها)

و * (باقی تکلیف ریشت ها) برو برو.

نویسه ریشت ها:

1. `a = "python" + " " + "strings"`

2. `print(a) ➡ python strings`

3. `print("-+-" * 3) ➡ -+-+--+-+`

1. `a, b, c = "python", " ", "strings"`

2. `print(a+b+c) ➡ python strings`

1. `name = "python"`

2. `version = 3.11`

3. `print(name + version)`

➡ Error :: can only concatenate
str (not "float") to str

نکت: باقی بیشی با بیشی بولن دوسته مانند دیلیس اسیا می توان از
عملّر = استفاده نمود.

نکت ②: از عملّهای عضویت همی توان ای ای بیشی وجود کمالت یا بیست و دی

1 ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

نکت: دست دای دیلیس استفاده کرد.

* کاراکتر های Escape : \

بر صورت پیش از متن تعدادی اکاراکتر خاص تعریف شده

لهمان آن دهارا درون رشته ها بدار بر . تمام این

کاراکتر های با \ آغاز می شود .

رواقعی این کاراکترها امکان برای درج بخ دیگر از کاراکترها
حسته که نفع نظری آنها را به ساری توسط صفحه لیید وارد نمود .

\n ← پایان سطر جایی و رفتار به سطر جدید

\t ← درج هشت ناصله (تغییر Space)

\r درج یک کاراکتر ،

\" درج یک کاراکتر "

\a درج یک کاراکتر A

\uxxxx درج یک کاراکتر یونیکد 16 بیتی با استفاده

از مقدار هذارسیمال (پایه 16) آن : \u067E

\uxxxxxx درج یک کاراکتر یونیکد 32 بیتی

با استفاده از مقدار هذارسیمال آن : \U0001D11E

\1000 درج یک کاراکتر با استفاده از مقدار انتال

(پایه هشت) آن : \123

\xhh درج یک کاراکتر با استفاده از مقدار هذارسیمال

(پایه هشت) آن : \x53

\b ← پاک کردن کاراکتر قبل آن Backspace



این دوئل را سه‌تارها گاهی موجوب مسئل می‌شوند؛ اصرفت^۸
لئید می‌خواهیم آدرس خالی از سیستم عامل ویندوز را جای را می‌

۱
۲

```
fpath = "C:\new\text\sample.txt"
print(fpath)
```

C :
ew ext\sample.txt

برای حل این مسئل صحبتون اصرح جای نیاز به ۱ است از^{۱۱}
استفاده در؛ ولی راهنمای جامع تر اینجا رشته‌های خام^{۱۲}
است در این نوع رشته‌ها، کاراکتر (Raw Strings).^{۱۳}

های Escape بی ائمه‌ستن. رشته‌خام با اخزوده شدن^{۱۴}
یک حرف کے یا R به ابتدئی رشته مجموعی اینجاد می‌گردد.^{۱۵}

* تبدیل کرد ب کاراکتر و برعکس رشته خام^{۱۶}
برای اینکه کامپیوتر تواند کاراکترها را درکرد نیاز به سیستم‌هایی^{۱۷}
است که آن‌ها را برای تبدیل آن درها می‌پایند. دو کار لذتی که^{۱۸}

به مانند سیستم اسلی ASCII (محدودیت) یا سیستم‌های جامع تری^{۱۹}
مانند UTF-8 است که استاندارد یونیکد Unicode در سیستم^{۲۰}

است. گاهی نیاز است که این درها سری داشته باشند و با^{۲۱}
کاراکترها بر اساسی این‌ها کار ننماییم.^{۲۲}

برای این منظور برای سوون می توان از دوتابع `ord()` و `chr()` استفاده کرد که کارالت به زیر مذکور است:

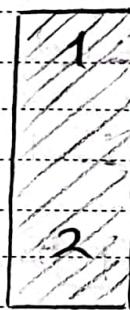
تبدیل عدد (درباره کارالت) و (درباره کارالت به عدد)

استفاده از `ord()` و `chr()` برای نمایش کارالت با ازفته

و بدلیل عدد (درباره ده) که بینگرد کارالت مورد نظر است

رایجی ترند: همینطور تابع `chr()` علی‌النعت عمل نموده.

(بامراجعه بر سایت `utf8-chartable.de` یا سایتهاها متشابه می‌توان درهای مربوط به کارالت را پیدا کرد.)



1. `print(ord("A"))`

222 65 ✓

2. `print(chr(65))`

222 A

* قالب بندی رشت (String Formatting)

قالب بندی اینان بجز جایلین (ردن یک یا چند مقدار (شی))

گاهی همراه با اعمال تغییر لخواه - (روزن یک رشت است

لبه چند دوستار برای ساخته سازی می‌گردد:

1. لوشن‌هایی و با استفاده از الدوی (values) `%s ... %s`

2. فراخوانی متد `format(values)` با الدوی

3. f-string

① مالیات‌بندی به روش اندیم

از دو بخش تشکیل شده است، بخش امتاچپ - کمیلر٪
رئستهای را مشخص می‌کند که شامل یک بازنده که جایلداری
سی محابا شد (ازدهای جایلداری آهنگ باشد) کارالت^{۱۰} بنز
برفع محاسبه شوند (۵٪) و در سمت راست آن همی
هزای خایلداری (جایلداری آهنگ) در رسته، داخل برانتز
محل دارد. این اشیا به ترتیب از هشت جهادون
رسته خایلداری نامی می‌شوند.

نمونه کد:

1 " %r is a %s language." % ("python"
2 و "programming")
3 'python' is a programming language.

برخ از ازدهای جایلداریها:

جایلیف) در قالب یک رسته به شلل خوبی داده (۹۰٪)
repr() از ماتبایت است به سکل احوج (تابع) (۵٪)
جایلیف) در قالب یک کارالت → یک عدد صحیح به نسائل
که کارالت می‌باشد را بکارالت یعنیک تبدیل نموده و در نهاد رسته، مارمی ازده
"A, B, C" (۹۰٪)

جا^یل^یخ^ن در مالب^ر یک عد^د ص^م در یا^ی ده

%

جا^یل^یخ^ن در مالب^ر یک عد^د ص^م در یا^ی هشت

%

جا^یل^یخ^ن در مالب^ر یک عد^د ص^م در یا^ی سانده با حرف لوح

%

جا^یل^یخ^ن در مالب^ر یک عد^د ص^م در یا^ی سانده با حرف بزرگ

X

جا^یل^یخ^ن در مالب^ر یک عد^د ص^م در یا^ی سانده با حرف بزرگ (فَسَاسْ مُضَفٌ: ۶)

%

همانند ۴٪ با این تفاوت نه `nan` و `inf` را به سل

%

درجی کرد `INF` و `NAN`

%

جا^یل^یخ^ن در مالب^ر یک عد^د ص^م سانور به سل نهاد علی با حرف لوح

%

جا^یل^یخ^ن در مالب^ر یک عد^د ص^م سانور به سل نهاد علی با حرف بزرگ

%

نت و همچنان در این الگوی این با استفاده از پس سی ای پی اسی ایجاده سانه

%

نمود که در آن اسیا با استفاده از لیز جایلر اسی اسوند و پیلس تریبا

%

آنها این می خواهند.

%

در اصل می دویان برای بخش سمتاچی این تاب ساختاری

%

ساخته بایست این نظر نرفت:

%

% [(key name)] [flags] [width] [:precision] typecode

%

که بیانگر همان حرف تعیین شده نموده نوع ده جایلر اسی ای باشد (احیاری)

%

نتر: در هر استفاده وجود هر یک از [] ها اختیاری یا بستگی به مورد

%

۸. \rightarrow در حالت مید را خل پرانتز (key name) نویسند.

۹. (در مواقع استفاده از سی دی اس تری) آورده می شود.

۱۰. \rightarrow می تواند بیان سند باشد flags.

۱۱. \rightarrow موجب ارجح علامت عددی مسود +

۱۲. \rightarrow موجب چیزی است که مقدار می برد (حالت

بیش از حد راست چین)

۱۳. \rightarrow تعیین فضای خالی اضافی با صفر یا می برد (حالت

بیش از حد space نداشت)

۱۴. \rightarrow اندازه رشته را تعیین می کند در مواردی

۱۵. لکه اندازه تعیین شده بیش از اندازه واقعی باشد فضای اضافی

۱۶. رایانه با صفر (به لکه حالت قبل) و یا space هر لرد و

در زمانی که لکه تعیین شده این ریشه نایاب می برد.

۱۷. \rightarrow در مورد اعداد صحیح سناور، دقتاً می

۱۸. تعداد ارگام بعد از صفر را تعیین می کند (دقتاً بیش از ۶):

۱۹. در مواردی که تعداد تعیین شده لکه از سه ا عدد واقعی ارگام بعد از صفر باشد

۲۰. عدد لکه می شود.

نکته: برحای width و precision می توانیم از * استفاده کرد

در لین صورت عدد مربوط به آن اعماقی در بخش اسمنت است

آورده می سود و سی جایلینی امی باست ارسال از آن

زیز لرید. (کاربرد در مواجهی که این اعداد در طول اجرای

برنامه زیست نزدیک است)

11 1

$n = " " "$

نمونه داده :

2 2

$\therefore \% 15s : \% - 10s$

3 3

$\therefore \% 15s : \% - 10s " " "$

4 4

$z = (" \text{First name} ", " \text{Richard} "$

5 5

"Last name", "Stallman")

6 6

print (n % z)

>>> First name : Richard

Last name : stallman

7 7

"%.f, %.2f, %.*f" % (1/3.0, 1/3.0, 4, 1/3.0,

8 8

>>> 1.0, 33.3333, 0.33, 0.3333

9 9

10 10

"... { } ..." . format (values)

format... تالیب بینی با متن ②

در این مالیات اشیاء که کارخانه هایی دارند ممکن است هستند و با استفاده از این سیاست یا نام آنها داخل {} در رشت
جایپلایس می شوند.

```
'{0} {1} {2}'.format("python", "programming", "language")
```

12. >>> Python programming language

نسته؛ همان طور نمود درس تابع خواهم اموخت، بدون انتیاز به رعایت
رسیب می‌توان آردویان‌ها را با انساب مقدار مورد نظر به
آن‌ها ارسال نمود.

```
'{2},{0},{1}'.format('a','b','c')  
+ + +  
=> c,a,b,
```

خانه بناهیم اسیا، به تسبیح لر در آردوغان استاد حصار دارند
جایلدار سویند، نیازی به ذکر ادیسا یا نام آردوغان نمی باشد.

نلتہ ② : سا اور دن یک * پیئسٹ اردو مانگ کے لیے یہ سئی اپنالاں ایسیت
(مائندگی رئستہ) می توان اعضا ان را رستیاب نمود،

۱۹) البته جانیم بخواهیم از آردومن حاکمیتی این استناده
لئیم دارم است اآن ها در استان مدد هزار داشتند، باشند ^{۲۰} در

8. ' { 2 } , { 1 } , { 3 } . format(' z * abc ')

9. z, a, b, >

10. پس نوئن رست، این مالب یعنی ساختار مساب زیر دارد :

11. { [Fieldname] [!conversionflag]
12. [:format spec] }

13. همان اندس یا نام اطومان \leftarrow Fieldname .

14. \leftarrow می تواند یکی از حروف r, L, F

15. باید له پر ترتیب str(), repr() را بروی سی

16. مراخواں می نہیں (این حرف با ! (! سمع می شود)

17. format spec : \leftarrow چونکی درج سی در رست را

18. تعین ای دن - (با : (: سمع می شود)

19. و خود ساختاری به صافنده ہائین دارند :

20. [[fill] align] [sign] [#] [0] [width]

21. [,] [. precision] [typecode]

22. fill \leftarrow می تواند هر کاراکتر قابل جایی باشد .

23. (از این لئین برای هر ہر کو نفای خالی که توسط width ایجاد می شود)

24. 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
25. align (استفادہ می شود .) وابسته به

align باید بتواند یکی از کاراکترهای $\{ \text{ل}, \text{ر}, \text{ن}, \text{ز} \}$ باشد.

باید در هر ترتیب راست چین، وسط چین و چین باشد.

و باید $width$ (اس از آن آورده بگشود. صفات اندازه نشست را بخوبی نمای لست)

\leftarrow sign \rightarrow بلکه اینها علامت عدد کاربر دارد و می توانند بیان اینها را فرمودند. فضای خالی space باید.

علامت تمام اعداد بیست و چهار تنها موجب لست علامت اعداد منفی درج باشند و منفی را درج نمایند. درج علامت اعداد منفی به جای علامت احتسابی space

\rightarrow برخلاف تابع قبل، می توان تبدیل یا نمایم. در اینجا دو راه را داریم. چنانچه باید # به بین اندک از اینها افزوده شود پسونه یا یک نیز درج می شود. (همان 0x ۰۰ ۰۰)

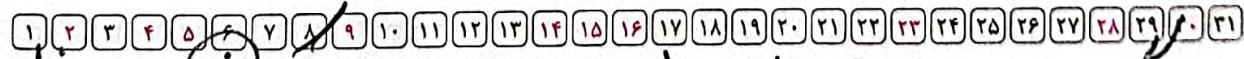
۰ \rightarrow تبعین فضای خالی با صفر یا می توانیم

با استفاده از یک کاما (comma) می توانیم عدد را در یک سریم از سمت راست جدا نمود (۳, 456, 789)

precision / width \rightarrow بر قابل تبل تعریف نشست.

typecode \rightarrow بر علاوه تابع قبل درج نوع لرجایی از اخیری ایجاد می شود.

یا بستگی به مورد استفاده دارد (در این تابع تنها در () وجود ندارد)



نکته برای بیان این صید می‌توان از $\%$ به ترتیب استفاده کرد.

نکته ۲: در مطلب قبل از * می‌توانستیم ترتیب‌های خود را در طرف یکی مقدار رهی نماییم: در مطلب اجیده برای این سطع می‌توان مانند کاری از برای جایلداری اشیاء آنرا کردم از { } استفاده و مقدار ترتیب‌های را در جایله از اینهاست است تعریف نماییم.

نوشته شده:

```

1 text = 'Right'
2 align = '>'
3 print ('{:0:{fill}{align}16}'.format(
4     text, fill = align, align = align))
>>> Right

```

۸. نالب‌بندی به نوشتار f-string

۹. ان نسخه پایتون ۳.۶ بـ امکان جیده در بین قاب‌بندی

۱۰. رسته‌های سد با عنوان f-string

۱۱. ساختار همان بساده شده حالت (۱) می‌باشد.

۱۲. یعنی اگر در لبری ای متن، حرف f یا F ماردهم

۱۳. آنکه می‌توانیم متغیر یا عبارات خود را مستقیماً در داخل آن

۱۴. با استفاده از {} ماردهم.

۱۵. ۱ name = "Mahdi"

۱۶. ۲ f"Hello, {name}."

۱۷. ۳ Hello, Mahdi.

۱۸. بسیار استفاده متخیرها (یا سیجـ حامل عبارات) یا استفاده

۱۹. صور نظر در سیوه f-string در نهایت بلکه قرار گرفتن

۲۰. درون متن یا استـ، صور نظر می‌باشد این نوع رسته تبدیل

۲۱. سوونـ. در این سیوه به صورت پیش‌فرض مصـ مـ است (۱) str-

۲۲. بلکه تبدیل به نوع رستـ، فراخوانـ میـ سـودـ. و کامـ تـونـ با قـلـ دـارـ

۲۳. نـسـانـلـ ۲۳! در انتـای سـنـی مـورـنـضـ، بـیـسـنـ لـرـ کـهـ مـتـ

۲۴. (۱) rperـ فـراخـوانـ (سـوـدـ).

۲۵. print (f"My name is {name!r} ")

۲۶. ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

۲۷. >>> My name is 'Mahdi'

نکته ۱: در این سیو میتوان از نهاد [] در خارج از اصول مطلب بیندی استفاده کرد و که باید توجیه را داشت که هر دو نهاد [] را بتن عوامان یک [] در نظر نهاد بتواند.

- نمونه کد:
- 1 $x = 3.546$
 - 2 `print(f"\{x\} is {x}")`
⇒ $\{x\}$ is 3.546
 - 3 `print(f"\{\{x\}\} is \{\{x\}\}")`
⇒ $\{\{x\}\}$ is $\{3.546\}$

نکته ۲: بخش درون رئه این مطلب، همانند متد `format` ساختار مسابه به آن را دارد و همین با قرار دادن یک یقه به صورت جداگانه تالب بیندی نماییم:

* بُرخ) متدهای کاربری یک رشته: ۰.۱
 ۸ \rightarrow یک پی از رشته نه کاراکترهای
 ۹ فضای خالی (whitespace) لازم است از انتها اشتباد.
 ۱۰ حذف آشده باشند.
 ۱۱ همین توان با استفاده از این متدهای کاراکترهای دلیلی
 ۱۲ را مشخص و بنابر این خود از ابتدای انتها است، حذف از
 ۱۳ لبه متدهای `lstrip` و `rstrip` نیز ممکن است مشابه
 ۱۴ با این تفاوت نه ممکن است حذف از انتها درست باشد `left`
 ۱۵ یا راست `Right` به انجام می‌رساند.

۰.۲ \rightarrow یک پی از رشته نه بحیثیت
 ۱۶ حرف‌گان به صورت بزرگ `Capial` نوشته شده با
 ۱۷ برگی می‌رساند.
 ۱۸ center(width). ۰.۳
 ۱۹ اندازه رشته است لفظ، و رشته را به صورت وسط چینی
 ۲۰ شوه دیدن این بازه برگی می‌رساند: در صورت نه اندازه بحیثیت
 ۲۱ لوچستر از اندازه واقعی رشته (len(String) باشد، رشته بین
 ۲۲ تبعیس بر این داشته باشد. این متدهای کاربری از انتها هم دارد نه
 ۲۳ توسط آن می‌توان کاری را بکار برد از فضای خالی بحیثیت نمود

۰۱ ۰۲ ۰۳ ۰۴ ۰۵ ۰۶ ۰۷ ۰۸ ۰۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱
 ↗ دو متدهای مشابه: `ljust(width)` / `rjust(width)`
 ۲۴ است چن جب چن

۴. برد (sub) Count (sub) \rightarrow یک رشته را لفظ و تعداد و موقع آن در رشته اصلی را برمی برداند: این متده دو آرایه ای را دارد که می توان نمط شروع و پایان اعمال این متدرسته را مشخص نمود.

۵. endswitch (suffix) \rightarrow یک رشته را لفظ و چنایه رشته اصلی با آن پایان یافته باشد مقدار True در غیر این صورت False را برمی برداند: این متده دو آرایه ای را دارد که می توان نمط شروع و پایان این متدرسته را مشخص نمود.

۶. find (sub) \rightarrow یک رشته را لفظ و اینیس شروع آن را بگزینی و موقع دیدن رشته اصلی برمی برداند و در صورت آن آرایه دیگری را در رشته اصلی یافتبند شود مقدار ۱ را برمی برداند: این متد دو آرایه ای را دارد که نمط شروع و پایان اعمال این متدرسته را مشخص نمود. لئے متد مشابه دیگر با الگو (algo) find وجود دارد ولی با این تفاوت ده اینیس شروع آرایه را بلکه آخرین موقع دیدن رشته اصلی برمی برداند.

نکته: در صورت که نیاز به اینیس ندارید و تنها می خواهید له وجود یک رشته



درین رشته اصلی بحیثی سیماید از عمل in استفاده

۷. `index(sub)` همانند متد `find(sub)` است
با این تفاوت اگر رشته ای که آن را می بینیم در رشته اصلی
یافت نشود یک خطا `ValueError` ارائه می کند.
لهم متد مشابه دیگر `rindex(sub)` بالدو

۸. `join(literalable)` یک رشته با اعضای تماش
از نوع رشته را به صورت آرایه ای دریافت می کند و با
استفاده از رشته اصلی اعضای آن را به یکدیگر پیوند را
و برمی کرداشد.

۹. `split(sep)` یک رشته را به صورت آرایه و
رشته را براساس آن از هم جدا کرده و به صورت آرایه
شی لیست `List` برمی کرداشد. این متد یک آرایه
اخیاری تر دارد که می توان
۱۰. `lower()` تمام حروف الفبا انگلیسی موجود در رشته را به حرف لوچ
که رشته را از سمت راست به میانسایی کند.

۱۱. `upper()` تمام حروف الفبا انگلیسی موجود در رشته را به حرف بزرگ
تبديل می کند برمی کرداشد.

۱۱. `new` (`old.replace(new)`) ← الرئسته صورت آن
اگر لوگون در راسته `old` در تمام رئسته اصلی بخش اضافی به
برابر مقدار لوگون `old` هستند را با لوگون `new` بجای مقدار لوگون `old` هستند، جدید را برمی کردند.
این استدیکت لوگون اختیاری است اما اگر نه میتوان تعداد عمل
جاگایزی را احسن نمود.

۱۲. `islower()` ← الرئسته حداقل شامل یک از حروف الفبا
انگلیسی بوده و تمام حروف الفبا آن به صورت لوچلر
باشد مقدار `True` در غیر این صورت `False` را برمی کردند
لکن `(upper())` بر عکس است و تمام حروف الفبا آن به
صورت بزرگ باشد `True` در غیر این صورت `False` را برمی کردند.

۱۳. `isalpha()` ← الرئسته حداقل شامل یک کاراکتر
بوده و تمام کاراکتر های آن تنها یک از حروف الفبا انگلیسی
(لوچل را بزرگ) باشد مقدار `True` در غیر این صورت
False را برمی کردند.

۱۴. `isalnum()` ← الرئسته حداقل شامل یک کاراکتر
بوده و تمام کاراکتر های آن تنها یکی از عدد های ۰ تا ۹ یا
حروف الفبا انگلیسی باشد مقدار `True` در غیر این صورت
False را برمی کردند.

١٥ . () isdigit() ، حاصل شامل ين ← الرئست

با سند مقدار True در غیر این صورت False را برمی‌لرایز.

10) casefold هماین متود عمل لرده و lower() تفاوت نه
تمام حروف را به حروف های لوچ تبدیل لرده با این تفاوت نه
قدرت عمل و سایپورت بیشتری در بقیه زبان‌ها دارد.

لابه space تبیل می‌کند و در صورت ارسال لیست به این متده (عدم صحیح) آن تعداد را با space جایلوبرنامی کند به جای اد

15. **isIdentifier()** \leftarrow صورتی که بست صور نظر شایط
16. اخاب شناسن (زدیش) را است باشد True و در غیر آن False برای تعبیه.

19. `[[z, [x, y], maketrans(x, y, str.translate)]]` این روش استاندارد برای جدول ترجمه مابین استفاده برآمده است. (۱) `str.translate()` برمی‌گرداند
که نقطه پیک آن را کاملاً ارسال شود، باید پیک دیلشتری باشد.
آن دوباره عمل ارسال شود، باید رسته های با طول مساوی باشند که در
این صورت هر کاراکتر در مجموعه بقیه کاراکترها را به کار نماید.
رسته باشد که کاراکترهای آن به `None` نداشته باشند.

ساختار های راه در پایتون :
در درس قبل با انواع راه در پایتون آشنایی کردیم و در این درس به
بررسی راه دیگر راه زبان های برنامه نویسی عموماً با عنوان
ساخته راه بیان می شوند .

بر اساسی ای لفته می شود له برای Data structures

نکره ای از یک مجموعه راه طراحی شده اند و هر یک راه را
ویژگی های مخصوص به خود دارد .

چهار ساخته راه مترادف در پایتون هست

tuple / dict / set / tuple

* لیست (List) *

لیست یک نوع بسیار انعطاف پذیر در پایتون می باشد .

این نوع همانند رشته مکالمه sequence بوده ولی

برخلاف آن یک نوع تغییرپذیر Mutable است .

لیست با استفاده از کوچک [] ایجاد می شود و درینها
اعضا را حفظ و می تواند عضوهایی از هر نوع را داشته باشد .

لیست کما و از یکدیگر جدا می شوند .

لیست اند و امتحن می باید نکره ای ایشیا ، لونالون او تیکیز نیزی است .

l = [15 , 3.14 , 'string' , [1 , 2]]



← حتی شامل خود لیست

۱۰

سده شنیده

لند

۱۳۹۹

30 June 2020

۱۴۴۱ ذی القعده ۸

روز صنعت و معدن

نکته ۱: سی لیست جزو انواع دنباله (همان ویژگی‌هایی را که برای عنوان دنباله بیان نموده) است.

نکته ۲: چنانچه یک دنباله جزو عضوهای سی لیست باشد، با استفاده از اللو `for i in seq` می‌توان عضوهای آن را در دسترس بگیری. این ترتیبی نمود که این‌پس از `for i in` عضوی لیست اشاره دارد و این‌پس از `i` اشاره به این‌پس از خلی عضوی دارد.

(این اللو به همین این‌سویه می‌تواند ادامه یابد)

نکته ۳: نمونه داده:

```
1 l = ['python', 2.56, [128, 'a', 'z']]  
2 print(l[2][1][0]) # درج a  
3 print(l[0][:2]) # درج py
```

نکته ۴: سی لیست جزو انواع **Mutable** پایتون است و می‌توان عضوهای آن را تغییر دار؛ این تغییر به شکل حذف، درج عضو جدید یا جایگزین کردن یک یا چند عضو باشد.

حذف لرن → کافی است یک سی لیست خالی [] را بیلی یا چند عضوانشی لیست موردنظر انتسبار همیم:

```
l = [0, 1, 2, 3, 4, 5, 6]
```

```
l[2:5] = [] # درج [0, 1, 5, 6]
```

دیگر لرن عضو جدید سے [] نہیں بلکہ موصحی

ان شئیز کو کہ اس کا رارڈ می خواہیم دیج دیکھنے اپنے
سیوڈ فریز لرن می خواہیم دیج لیکن ایسا نہیں
لینے می باسنتا یہ دنبالتہ باسندورہ)

$$l = [0, 1, 5, 6]$$

$$l[2:2] = [2, 3, 4]$$

$$l[0:0] = 'abc'$$

$$\rightarrow [a, b, c, 0, 1, 2, 3, 4, 5, 6]$$

$$l[3:3] = [d, [-2, -1]]$$

$$\rightarrow [a, b, c, d, -2, -1, 0, 1, 2, 3, 4, 5, 6]$$

• جایزیں لرن

$$l = [1, 2, 3]$$

$$l[1] = 'py' \rightarrow [1, 'py', 3]$$

$$l[:2] = [0, 0] \rightarrow [0, 'py', 0]$$

* دستور `del` ← با استفاده از این دستور می توان یک

عضو یا یک تک لرزی لیست را حذف کرد :

$$l = [a, b, d, e, f, g]$$

$$del l[1:4] \rightarrow [a, f, g]$$

۱۲

پنجشنبه

نیم

۱۳۹۹

2 July 2020

۱۴۴۱ ذی القعده ۱۰

همین‌جا توکیم اذاین راستون (۱۴۷) باین حرف کامل
یک متخصی استفاده نهایم . با حفظ یک متخصی در ارجام آن
به سی نزدیک حرف می‌شود و چنان‌جایز ریلیت (به آن ایسی وجود
نشسته باشد، سی ای) که متغیر به آن ارجام راست نزدیک آن
حافظه حرف می‌کرد .

* اسلوب چندگانه :

ایوان یک سی دیست (یادلی یک سی دنبال) را به تعدادی
از اسلوب راد و متغیرهای جداگانه‌ای ایجاد نمود؛ این عمل
خوانده می‌شود . در این شرایط مفسر پاسیون از عکس
دنباله را با حفظ ترتیب به می‌لی از نام‌ها اسلوب می‌دهد
که در حالت عادی ای باست از عبار نام‌ها با عضوهای دنباله

برابر باشند :

۱۳

جمعه

نیم

۱۳۹۹

3 July 2020

۱۴۴۱ ذی القعده ۱۱

ولادت حضرت امام رضا علیہ السلام (۱۴۸) م

$$l = [1.1, 2.2, 3, 4]$$

$$a, b, c, d = l$$

$$a = 1.1 / b = 2.2$$

$$c = 3 / d = 4$$

نکته: ولی می توان یکی از نام ها را توسط نهاد * نسخانه لذاری ۱۳۷۹
کرد: در این سلایط مفسر با استفاده از آنچه از این نام عضوهای
دبیالله و نام های ابیاد می شود: این حالت تمام عضوهای
اضافی (در توالی با یک بقیه است) به نام نسخانه لذاری شود.
استساد: داده می شود: (البته ترتیب عضوهای ذباک
در هر سلایط: رعایت خواهد شد) .
۱۱

1 a, * b, c = [1.1, 2.2, 3.3, 4.4]
=> a = 1.1
b = [2.2, 3.3]
c = 4.4

* حملہ حابہ لست :

کتاب از عبارتی + (پیوند لیست) و * (تلار عضوی لیست) بروید.

17 هچنین برای برخی از برابر بودن مقادیر دوستی ایس-امانند ریلز
18 ائمای از عمل = استفاده از

و از عملرهای عضویت هم می‌توان برای برسی وجود سی ای دعوای استفاده نمود.

* بُلْدَلَتْ : اجْتَمَعَتْ

همانند دیگر اسیا می‌توان با انتساب یک متغیر بوجول از شی لیست به یک نام جدید، متغیر دلخواه این نوع شی ایجاد کرد. البته در این حالت شی بُلْدَلَتْ نمی‌زند و تنها یک ارجاع جنبدی از این نام جدید به شی آن متغیر را دارد. این موضوع با استفاده از متابع (def) مقابل آن مقول است. خروجی این تابع برای نشان شی در حافظه می‌باشد و بیرون از آن مقدار آن پیسان بگذارد و متن این بودن شی آن هاست:

$$1 \quad l1 = [1, 2, 3]$$

$$2 \quad l2 = l1$$

$$3 \quad \text{print}(l2) \quad \boxed{[1, 2, 3]}$$

عضوهای یک شی لیست تغییر نمی‌هستند و باید توجه داشت اگر این نمونه را هر دو متغیر بگذارد آن اشاره دارنده الگوی متوسط یعنی از متغیرها، عضوهای شی صور نظر تغییر دارند سو نه، مقدار صور انتظار ما از شی تغییر دوست نیز تغییر خواهد کرد.

$$1 \quad l1 = [1, 2, 3]$$

$$2 \quad l2 = l1$$

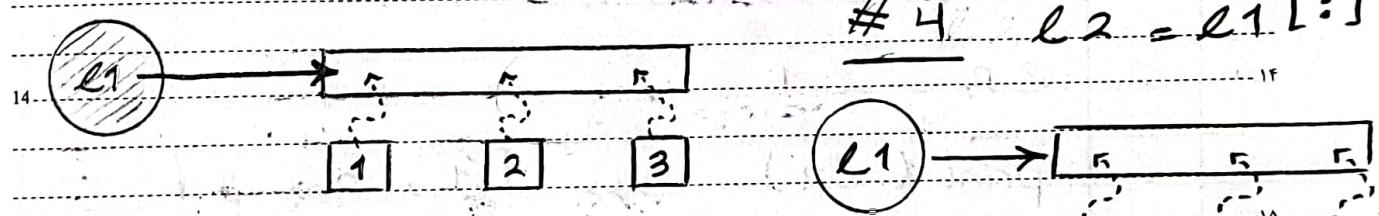
$$3 \quad l1[0] = 7$$

$$4 \quad \text{print}(l2) \quad \boxed{[7, 2, 3]}$$

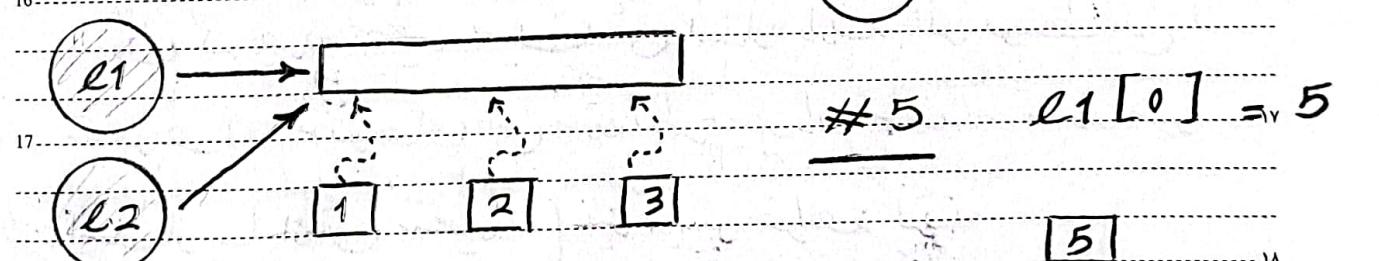
الراین موضوع را پیش می‌سازد برای وضع آن می‌توان از یک متغیر پیش‌بینی کرد که این پیش‌بینی ممکن است جدید نسبت به پیش‌بینی اول باشد و این پیش‌بینی ممکن است جدید نسبت به پیش‌بینی دوستی باشد. در این سایر هر دو متغیر به اساتیج ارجاع کاری در محل های تفاوت از جهانگردی استواره خواهد بود.

در حالت اعداد میان این افراد پیش‌بینی انسان به کار جدید نسبت می‌شود اما این افراد از این افراد (معنی تمام عضوها) استفاده نمی‌کنند.

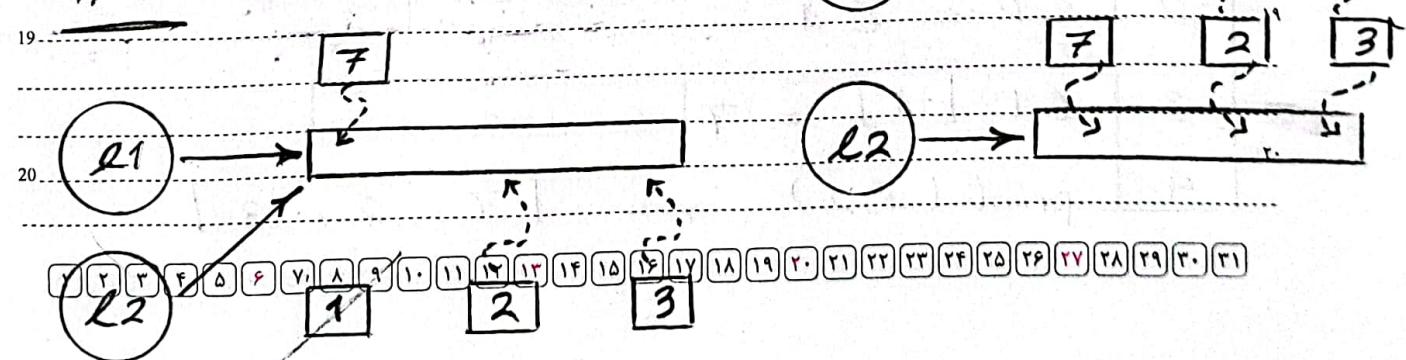
#1 $l_1 = [1, 2, 3]$ شردم ($l_2 = l_1.\text{copy}$)



#2 $l_2 = l_1$

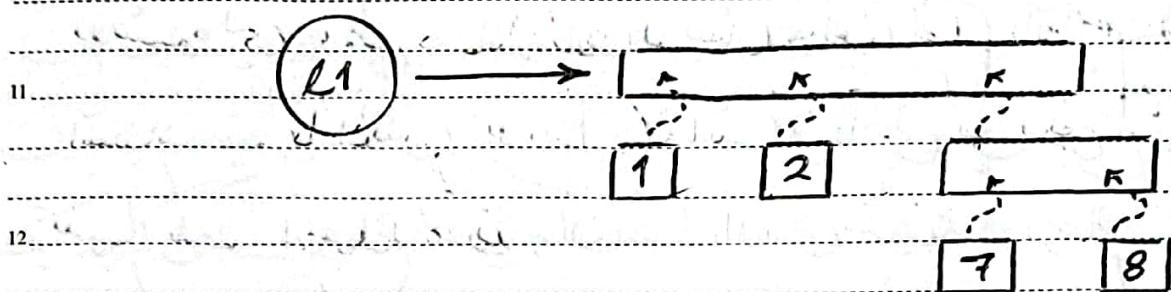


#3 $l_1[0] = 7$

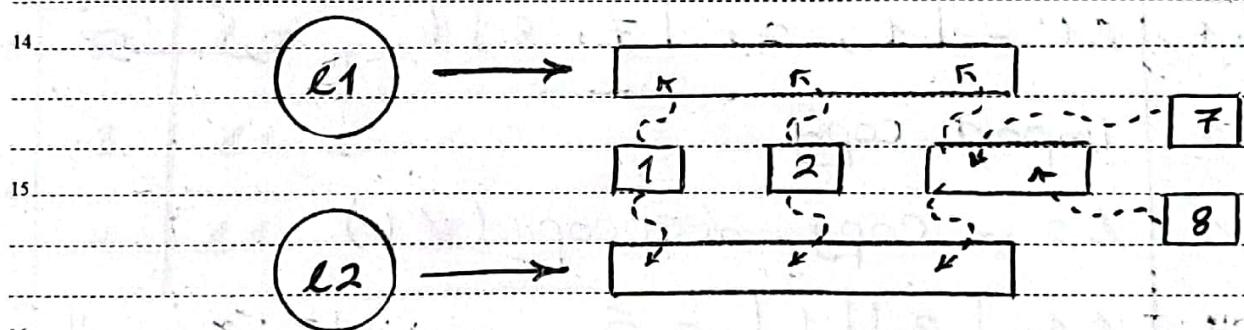


ولی الرئیسی لیست مودرن نظر عضو های از نوع لیست ها
 (یا هر نوع تیغیز پنیر بیل) رئیس نایب رئیس مسئل موق همچنان
 برای عضو های ابتدی است.

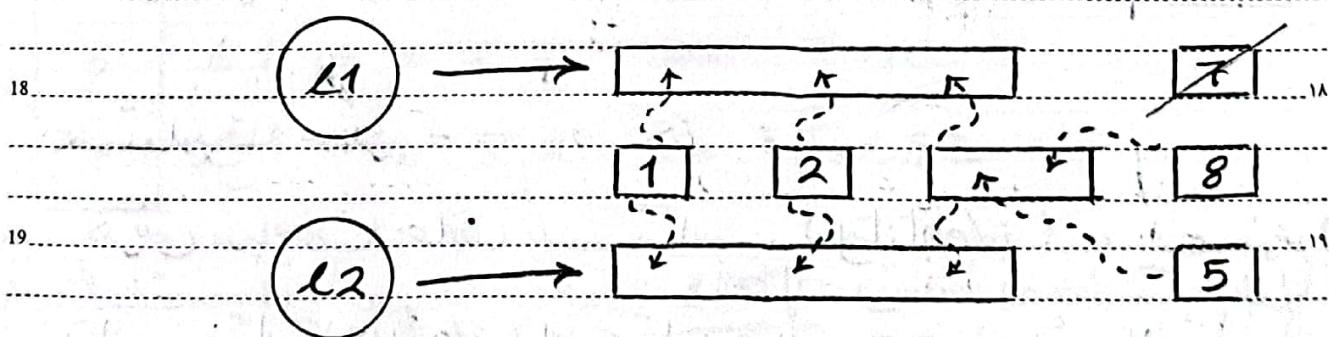
$$\#1 \quad l_1 = [1, 2, [7, 8]]$$



$$\#2 \quad l_2 = l_1 [:]$$



$$\#3 \quad l_1[2][0] = 5$$



در پایتون لی (shallow copy) به دو شیوه سطحی (copy) و عمیق (Deep copy) ایناک می‌شود که بر ترتیب توسط تابع همان (copy) یا عوض (deepcopy) از درون مسئول پرستی می‌شوند.

بر شیوه سطحی همانند کاری که بین این اینها دارد این (بعنی انسادی ب با اینس []) اسیاء داخلی لی (عنی سیوند و نهایی) ارجاع جدید به آن‌ها دارد می‌شوند؛ در حالی که توسط شیوه لی (عمیق) از تهمای اسیاء (تغییر یافته) داخلی نیز لی (ایجاد می‌گردید).

نمودن کرد:

```

1   l1 = [1, 2, [7, 8]]
2   import copy
3   l2 = copy.deepcopy(l1)
4   l1[2][1] = 5
5   print(l1)      [1, 2, [7, 5]]
6   print(l2)      [1, 2, [7, 8]]

```

* تفاوت اعمال == و کن ==

هر شی در پایتون حاوی یک سیاس (identity)، یک نوع (Type) و یک مقدار (Value) است. عملر == دو شی را از نظر یکسان بودن امتیاز مور برخی (قلد) می‌گردید. در حالی که عملر کن دو شی

را ان تظر بیسانا بودن تخاصم (خوجات ایام (زمان))
 یا همان نشان آن ها را حافظه مولای سی می دهد .
 مفس پایتون بلکن صفحه جویی بر زمان را حافظه از ساخت
 عدد اسیاء نوع صیغه و شرط : لوجیک محقق اجتناب
 ی لند و آنرا یک ارجام جدید به آنها می دهد . اما این
 موضوع در پور اسیاء دیگر درست نباید و مفس پایتون
 بلکن چرمه عی که بلکن این نوع اسیاء تعریف کامی لزدید که
 شئی جدید ایجاد می کند و به آنها ارجام می دهد .

13	$l_1 = l_1 = [1, 2, 3]$	نوونکر :
14	$2 \quad l_2 = l_1$	
15	$3 \quad l_1 == l_2 \rightarrow \text{True}$	
16	$4 \quad l_1 is l_2 \rightarrow \text{True}$	

5	$l_2 = l_1 [::]$	
6	$l_1 == l_2 \rightarrow \text{True}$	
7	$l_1 is l_2 \rightarrow \text{False}$	

نکته : با استفاده از دیس list می توان یک شئی بیسانا ایجاد کرد یا اسیایی که از نوع زنگاله هستند را به یک شئی بسایریں کرد .

نیز سرچهای کاربری ملکیتی نیستند.

سی لیست، تغییر یعنی اسباب و مصدّهای آن برخلاف اشیا
لسته یک سی جمله تغییر یافته را پر نمی کردند بلکه تغییرات
داری همان اشیا اینداد می کنند.

سی x را به اسکی لیست موردنظر اضافه کنید.

اضافه می‌نماید (حکایت این صد همانند عمل $[x] + 1$ است)

۲. extend(s) ← عضوهای سئی دنبال کر را به انسانی
لیست مورد نظر اضافه می‌کند.

۱۴ insert(i, x) . ۳
این عضو جدید مانند x را در موقعیت i از لیست باز نمایی کرده و آن را در آن جای داده.

۱۴. $\text{remove}(x)$ را از سمت اچ x در لیست A مورد نظر باز بگیرید.

۱۵. به نیمال رئی x می‌گردد و تخصیص موردنیافست‌سده را از لیست A برداشته و آن را بازگرداند.

۱۶. حذف می‌گردد. حذف یعنی هم عضوی ابزار رئی x با یافتن شود و از لیست خطا مارک شود.

نلت : در مواردی که می خواهید انسان خاصی را از بیست

حُفَّ نَاهِيَهُ از رَسْتَوْن deh استفاده نماید.

۵. pop() ← عضو متناظر با اینس نرا از لیست حذف و به عنوان خروجی برمیگرداند. چنانچه اینس به همراه حذف شده، نسخه بروزرسانی شده اینس (آخرین عضو از لیست) مورد نظر را حذف و برمیگرداند.

۶. index(x) ← در لیست مورد نظر از سه تا هجده به دنبال شئی x میگردیم و اندیس نخستین مورد یافته شده را برمیگرداند. این متده آرایمان اخراجی n را دارد که به نکات آن میتوان تبعیض نمود اینس چندین مورد یافته شده برمیگرداند. همچنان عضوی برابری باشی x یافت شود یک خطاب رئیس میگردد.

۷. count(x) ← تعداد وقوع شئی x در لیست مورد نظر میگذرد. برمیگرداند.

۸. clear() ← تمام عضوهای لیست مورد نظر را حذف میکند (برای موارد این متدها دلیل رسیدور [] یا باشد)

۹. reverse() ← عضوهای لیست مورد نظر را وارون میگردند. (دروی خود لیست اعمال میکند.)

۱۰. reversed() ← تابع هم عملکرد مشابه این متدها را اما با این تفاوت دارد که لیست جدید تسلیل میگردد و در خود لیست اعمال نمیکند.

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰	۳۱
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

۱۰ مر (۱) sort را به عضوهای یک لیست را مرتب می کند.

این است در حالت سئی هفتم) به صورت صحیح اینجاست:

مرتب سازی می کند ولی میتوان یا فرستادن مقادیر True

به آر دومان اختیاری reverse، سیوه آنرا به نمودی بخسیر

در اینجا sort (reverse = True)

ست (۱) sort آر دومان اختیاری دلیلی نبین باشیم key دار که

میتوان با ارسال یک تابع تبلیغ آر دومانی به آن عمل دلخواهی

را بر روی تک تک عضوهای لیست مهربان نظر داشتیم) از معایس

و همیشه از اینجا بر اینجا رساند، انتبه، باشد توجه داشت که

تنهایی باشیم نام تابع به آر دومان است مرتب شوند

این دلیلی کامل آن.

$$l = ['a', 'D', 'B', 'c', 'e', 'f', 'G', 'h']$$

۱۶ ۲. sort (

$$l = ['B', 'D', 'G', 'a', 'c', 'e', 'f', 'h']$$

همان طور می شناسیده می شود حروف بندگی های این لیست مرتب

شده هر کار لرفته اند در واقع حروف بزرگ موجود در لیست به مقادیر

لوچری ارزیابی شده اند لکه ای بده اسلی این حروف توجه نمایید

توجه علیست این ارزیابی خواهد بود:

برای نفع این مشکل می توان بیش از آنکه عمل معایس را

برای این ساختار ابتدا نیز با خواص تابع پرورش
 عضوهای لیست، تمام حروف را به نسبت یا توجه کنیم
 تبیل نام اسم تا حرف در سطح میانی برای مقایسه
 قدر بگیرد:

$$l = ['a', 'D', 'c', 'B', 'e', 'f', 'G', 'h']$$

۲ $l.sort(key = \text{str.lower})$

$$['a', 'B', 'c', 'D', 'e', 'f', 'G', 'h']$$

نتد ۱: با استفاده از تابع آماره `dir` میتوانیم لیست از همان صفت‌ها و متدهای درست را بگیری و بگذارد.

نتد ۲: به جای استفاده از `sort()` میتوان از تابع آماره `sorted()` تغییر باهیف را تضمین استفاده کرد.

۱۷

۱۸

۱۹

۲۰

* تپل (Tuple) *

نوع تپل همانند نوع لیست است آنرا یا این تفاوت آن را تغییر نیز نیست . (بنا بر این پر نیست) مقدار حافظه کمتر (مصرف بیش از) و عضوهای آن ابرون پر نیز () مول را در می شود .

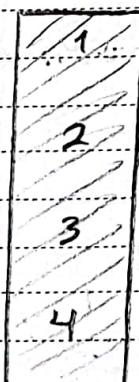
برای ایجاد سی تپل حی می توان از داردن یک لیست صرف نظر کرد و آنرا اسیا (با عبارت اها) را با کامار و از یک دلیل جذب خود نکته در انها سی تپل (تک عضوی) می باشد که نهاد کما مکار را در مانند (۱) از اینکه پر از در عبارت اها بین استفاده می شود (۱ + ۱) ، یا این که مفسر یا سیوون یک سی تپل را از عبارت ساخته است (۱)

$$t = ([1, 2, 3], 5)$$

۲ not int True

۳ t[2][1] ۲

۴ t[0] = 'x' Error



تجربه : نوع تپل یه لیل تغییر نایدی بولن ، سیسته ب نوع لیست در مصرف حافظه بسیار کمی باید ؛ بنا بر این پر نیست اسیا در مواقعی که نیاز به تغییر خاصی در راه ها نیست که نوع تپل استفاده می شود . همین در مواقعی که نباید راه ها تغییر دنند ، استفاده از سی تپل به جای لیست می تواند ازان ها در برقراری تغییر حافظه را نزد .

نکته ③: به دلیل ساختار ارجحیتی بسته اشیاء ذریعاتیوناً، اینها

- تغییرپذیری را درون اسی تولیل (صنایع لیست) وئرای اساهای

- خود را راسئه و همچنان تغییرپذیر خواهند بود.

نکته ③: همانندی بسته اسی تولیل نزد بر دو متده است (index)

و (count) را ترسی ادارد.

نکته ④: هر زیارت لذینان ابر اعمال تغییر در رئیس تولیل باشد، می توان

اسی صورت نظر را به صورت موقعیت اجرا کنیم لیست را تغییر کرد

لیز با استفاده از دلاس (list)، درین حالت می بتوان از

وئرای و متد های اسی لیست اینها را در آن اعمال

کرد و در نتیجه اینها با تغییر نوع دلیل دوباره به اسی تولیل بازنشسته

(توسط کلاس tuple).

نکته ⑤: دلاس (tuple) بعنوان ارتومنات یک اسی تولیل خالی را ایجاد می کند.

نکته ⑥: در همان استدلال مذکور تغییر تولیل به موضوع بجز انتشار اشیاء

تغییرپذیری توجه داشته و در صورت نیاز از ماروال copy

استفاده نمایید.

* دیکشنری (Dictionary) *

یک دیکشنری از انواع ساختهای راهه در پایتون دیکشنری باید باشد
 له بانام دو تا شده dict الائچه می‌شود. اشیاء نوع دیکشنری
 با استفاده از زماد آنلود { } معرفی می‌شوند و هر راهه در آن
 به شکل یک جفت مقادیر : نمایید (key : value)
 ذخیره می‌گردید. از این نوع سی با عنوان آئی mapping (نکاشت)
 پایتون ناین یار می‌شود که در این نوع هر سی نمایید به یک سی
 مقادیر map یا نکاشت راهه می‌شود.
 سی دیکشنری دنباله نسبت و هر عضوان به جای اندیس
 با استفاده از لیکر دستیابی می‌شود.
 دیکشنری پایتون یک چیزی نیز نیست و سی مقادیر می‌شوند از خص
 نویجی باشد حتی یک سی دیکشنری دیگر ولی سی نمایید نیز
 می‌باشد از انواع یک‌ناینیزی انتساب شود و باید توجه راهه
 له تا م تلیدهای یک دیکشنری ای پایسنا می‌باشد unique باشد
 از نسخه ۳ پایتون، دیکشنری قابلیت حفظ ترتیب
 عناصر خود را نیز دارد یا به اصطلاح ordered است.

$$d = [1: 'one', 2: 'Two', 'c': 'Three']$$

2 d ['c'] \rightarrow Three

3 d [2][1:] \rightarrow wo

نکته ۱: براي دستيابي هر مقدار سئي ديلستري آن الگو dic [key] استفاده مي کنيم که key را کن. الگيه متصل به مقدار موردنظر مي باشد.

نکته ۲: براي ابعاد سئي ديلستري مي توان از دلخواه dict() نيز استفاده نمود.

```
1 d = dict (one = 1, Two = 2)
2 print(d)
3 { 'one': 1, 'Two': 2 }
```

توجه راسته باسیمه که عضوهای سئي ديلستري از طبق آوردهاند اها وزیر شغل Key = value به دلخواه مرساده مي سوند و درین حالت براي انتساب لیلدها باید قوانین انتساب نام در پایتون را رعایت کنیم.

نکته ۳: از متابع () len نیز می توان براي ارسان آوردن تعداد عضوهای سئي ديلستري (جفت مقدار: الگيه) استفاده کرد.

نکته ۴: با انتساب يك مقدار جديده به يك ليد موجود از سئي ديلستري آن الگيد را تعديل و با انتساب يك مقدار به يك ليد جديده در سئي ديلستري وجود خارد يك عضو جديده به آن اسي

نکت ۵ : با استفاده از دستور `del` از دیکشنری حذف کرد.

نکت ۶ : اگر این کار را با عضویت دیکشنری خواهیم داشت.

نکت ۷ : اگر این کار را با عضویت دیکشنری خواهیم داشت.

جید) اضافه نمایید :

1 $d = \{'name': 'John', 'job': 'unemployed'\}$

2 $d['occupation'] = d['job']$

3 $del d['job']$

4 $print(d) \rightarrow \{'name': 'John', 'occupation': 'unemployed'\}$

1 $d = \{"b": 5, "a": 10, "c": 20\}$

2 $x = d['c']$

3 $del d['c']$

4 $d["x"] = x$

5 $print(d) \rightarrow \{'b': 5, 'a': 10, 'x': 20\}$

+ عملکردهای دیکشنری :

عملکردهای + و * بلك ائمای دیکشنری (عرفت نشده اند).

ان عملکردهای عضویت ای تول بلك بررسی وجود یکی باشد.

درستی دیکشنری استفاده کر.

+ بیک لرن :
 همان طور لره نفترمیزی (نیشتری) از انواع تیغه زدنی باشون است.
 و همان توضیحات ره در مورد سئی لیست ابیان است، اینجا
 حتم درست است.

+ تبدیل به سئی اریشتری :
 برای تبدیل اسیاء زبانه به مانند لیست را و تولی بردیشتری
 می باشد از ساختار تولید تو استفاده کرد، به تواند ای ره
 هر عضو این ساخته ها خود شامل دو عضو باشد.
 اما روئا ساده تر استفاده از تابع zip() است. می توان
 این اثون تصویر را این شرح تعدادی سئی زبانه را می بینیم و عضوهای
 نظری به نظر این اهالی در این هم می دهد، این این زبانه ها باید
 تعداد عضو ملزوم باشد؛ چنان ره عضوهای اضافی هر
 زبانه ناریده رفتگی می سود.

خوب (1) zip یک سی جدید ان نوع zip است و می توان آنرا
 را به عنوان آردومن از لاس dict ارسال نمیم.

$$K = [1, 2, 3, 4, 5.]$$

$$Z = [x, y, z] = \{x: 'x', y: 'y', z: 'z'\}$$

$$Z = zip(K, L)$$

$$print(dict(Z))$$

$$\Rightarrow \{1: 'x', 2: 'y', 3: 'z'\}$$

* بخش متد های کازبری سی ایلیشنری :

۱. items ← تمام عضو های شی زایبری لر دارد (نوبت رو عضوی)

۲. values ← تمام مقدار های موجود در شی زایبری لر دارد

۳. keys ← تمام کلید های موجود در شی زایبری لر دارد

۴. clear () . ۲ ← تمام عضو های یک سی ایلیشنری را حذف می کند

۵. fromkeys (seq) ← یک دیکشنری تبدیل های دریافت

ویلشی شی جدید ریلیشنری با استفاده از آن اینها بهادری نبود

البته تبدیل های این شی مقادیر مقدار هستند (None) که باید

در زماف به کن اهم قرار دارند. (dict . fromkeys ())

(این متد توسط خود للاش dict مرا خوانن می شود)

این متد یک آر دومان اخشاری این دارند که توسط آن می توان

یک شی را به عنوان مقدار پس از مقدار کلید های می توان

۶. pop (key) ← عضو دارنده کلید key را حذف کرد

مقدار آن زایبری لر دارد. خاصیت عضوی با این تبدیل یافتن سود

شی ایس امیس ایسند (آر دومان رفع کر اخشاری است)

زایبری لر دارد و لر این آر دومان ارسال نموده باشد یک خط

نمایشی می کند.

از این متد می توان یک تبعیس را حست که تبدیل ها استفاده نمود

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰	۳۱
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

d ['occupation'] = d . pop ('job')

۵. / `popitem()` ← بیوں آر لومان اسٹکا۔ ایں متے

در هر فراخواختایک عضو اسئی صورت را به صورت لیفواہ

حذف و په شل تولی (value و key) برمی لردازه و

چنانچہ دلسته خالی باشد یک خطای ارائه می کند.

۶. ✓ `get(key)` ← مقدار مربوط به لید key را

برمی لردازد. چنانچه درون اسی صورت نظر همی باشیں لید

وجود نداشت، باشد شی بیشتر مرضی بخوبی اسیده (آر لومان اخیری

روم) را برمی لردازد و اسینا آر لومان ارسال نسده باشد همچ

خطای ارائه نمی رهد.

۷. ✓ `update()` ← یک شی دلسته دیگر با به عنوان

آر لومان اسی لید و عضوهای شی صورت نظر را براساس آن

تغییب می رهد.

۸. ✓ `setdefault(key)` ← مقدار مربوط به لید را برمی لردازد

چنانچه عضوی باشیں لید روم، اسی صورت نظر وجود نداشت، باشد

لید را به همراه مقدار بیشتر مرضی بخوبی اسیده (آر لومان اخیری

روم) درون اسی اضافه می کند و خود این مقدار را برمی لردازد،

اگر آر لومان روم ارسال نسده باشد به صورت بیشتر مرضی مقدار name

در نظر لفتے می سورد و درون اسی اضافه می کند.

۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱



۸- مجموعه (set)

مجموعه از انواع سیکل ترتیب و ترتیب نداشته

Mutable پایتون است که برای مفهوم مجموعه در ریاضیات

می باشد . هر عضو مجموعه می باشد باید یکتاویل از انواع

تغییر نماین باید . نوع مجموعه را می توان بای استفاده از

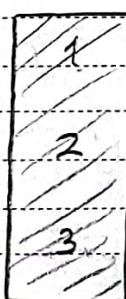
لیس / set () پاک نماین با استفاده از نهاد آنوار {} ایجاد کرد .

نکته : هم سینتیس خاص بای ایجاد یا باین یک سی خالی از نوع

مجموعه وجود ندارد و تنها می باشد از لیس set ()

لیس آن را مانند استفاده کرد . توجه داشته باشد که

{} باید یک سی ریلشنی خالی است و نه یک مجموعه خالی



$$5 = \{ 1, 2, 3, 4, 5 \}$$



۱۵- ۱. $a = \text{set}()$

۱۶- ۲. $\text{print}(a)$

۱۵

۱۶

+ عملرها برای مجموعه :

تحمای از عملرها جستجو له برای اسیاء مجموعه تعریف

خاصیتی ای لذت در حالی که در صور اسیاء دیگر چنین انتشاری

نمایند

۱۷

۱۸

۱۹

۱۹

۲۰

۱ . ۱ اجتماع (union) $A \cup B$ مانند $\{1, 2, 3\}$ خاصل

آن جموعه ای می باشد که تمام عضوهای جموعه A و

جموعه B را داشته و هیچ عضواضافه دیگر ندارد.

لئے استفاده از متده است (union)

۲ . ۲ اشتراك (intersection) $A \cap B$ مانند $\{1, 2, 3\}$

خاصل جموعه ای می باشد که تنرا شامل عضوهای است که

هم در جموعه A و هم در جموعه B

لئے استفاده از متده است (intersection)

۳ . ۳ تفاصل (Difference) $A - B$ مانند $\{1, 2, 3\}$

خاصل آن جموعه ای می باشد که تنرا شامل عضوهای از جموعه A

است که در جموعه B نیستند.

لئے استفاده از متده است (difference)

۴ . ۴ تفاصل تقابلی (Symmetric difference) $A \Delta B$ مانند $\{1, 2, 3\}$

خاصل آن جموعه ای می باشد که برابر اجتماع

تفاصل A از B و تفاصل B از A می باشد.

$$A \Delta B = (A - B) \cup (B - A) = (A \cup B) - (A \cap B)$$

لئے استفاده از متده است (symmetric-difference)

رتک آرکومانی (Arcomani)

۵. $A \subset B$ \leftarrow مانند (Subset)

که اگر مجموعه A از مجموعه B باشد مقدار True را پرسی کراین.

لئے استفاده آن متد (is subset).

۶. $A \supset B$ \leftarrow مانند (Superset)

اگر مجموعه A بگویی مجموعه B باشد مقادیر

True را پرسی کراین.

۷. لئے استفاده آن متد (is superset).

۸. برخی متدهای کاربری یک سی جمیعه:

۹. $\text{clear}()$ \leftarrow تمام عضوهای یک سی جمیعه را حذف می‌کند.

۱۰. $\text{add}(x)$ \leftarrow سی جمیع نایدی خ را در ضمونه از قبل

مدون سی جمیعه صور نظر وجود نداشت به آن اضافه

می‌کند.

۱۲. $\text{remove}(x)$ \leftarrow عضو خ را از سی جمیع صور نظر حذف

می‌کند. در صورت که خ سی جمیع وجود نداشت، باشد یک

خطای از اش می‌کند.

۱۴. متد مسابه دیگر باللو (discard) وجود دارد که خانه خ و وجود

راسته باشد آن را حذف می‌کند و در صورت تپورن خ خطای

۱۵. ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

تلریس نصی لند.

۸) pop ← لین ستدار الومانی ندارد و بره صورت دلخواه ۹) ۱۰) ۱۱) ۱۲) ۱۳) ۱۴) ۱۵) ۱۶) ۱۷) ۱۸) ۱۹) ۲۰) ۲۱) ۲۲) ۲۳) ۲۴) ۲۵) ۲۶) ۲۷) ۲۸) ۲۹) ۳۰) ۳۱)

۸) pop ← لین ستدار الومانی ندارد و بره صورت دلخواه

۹) یک عضو را حذف و بره عنوان (زوج) برش لردازد. در

۱۰) ساریگ له بجهوع خالی مایل خطای زریس امی لردازه

frozen set *

۱۱) یک نوع جدید بجهوع است اد همان لون، له می توانیم

۱۲) یک سی تویل را محاذل یک سی لست را تغییر ناینین تصور

۱۳) لئيم: frozen set راندز می تواند یک نئی بجهوع

۱۴) تغییر نایدیز تصور لر.

۱۵) نوع frozen set همان نوع set است با تفاوت ویژه

۱۶) های ان به عربان تیس زیری له با استضاده ان للاست

۱۷) () frozen set ایبار می کرد.

۱۸) نکته: متدهای طرایی عبارت update، آن عمل منظر را برعی

۱۹) خود آن بجهوع، اعمال می کند.

* بولی (Boolean) :

لایسی یا نر پاسون ازان برای اینجادی بولی استفاده می شود.

در واقع یک لایس فرزند از لایس اعداد صحیح int است.

این نوعی تئامی تواند یکی از دو مقدار True (درست)

یا False (نادرست) را راست بگذارد True برابر

عدد صحیح یکی و False برابر با عدد صحیح ۰ از نتایج می تردد.

نکته ۱ : لایس () یا متد () یا bool مقدار بولی یکی را بر جی براند.

نکته ۲ : در پاسون اشیاء زیر به مقدار بولی False از نتایج می تردد :

None ←

False ←

لے کے شی صفر (در انواع لونالون) : ۰ / ۰.۰ / ۰

لے کے تمام اشیاء دناله خالی : " " ، () ، []

لے کے شی دلستی خالی { }

لے کے شی مجموع خالی set()

نون (یک مقدار هیچ) را با (float('nan'))

۹

پنجشنبه
مرداد

۱۳۹۹

30 July 2020
۹ ذی الحجه ۱۴۴۱ (روز عبا) - روز اهدای خون

: None Type *

یک شی آصله و ب فعل مقدار در پایتون است.

این شی را معرف مقدار هیچ یا null در پایتون

بین در نظر گرفت. یک کاربرد صریح از این شی در زمانی است

که خواصم یک سخن بدهن مقدار اولیه ایجاد و مقدار دهن را

بر اساس سلیط دلیل در ادامه بزاید تعریف نمایم.

* نوع داره باینیت (bytes) :

نوع داره دیگری که معرف یک رساله از بایت های باشد.

از این نوع داره برای تعریف آنها داره های باینیت در پایتون

استفاده می شود. داره های سائل یک مایل تصویر یا داره های

لهم برلوگ لایس سبلیس (Network). تبادل می شوند، همان

در پایتون با این نوع داره مایل برداشت هستند.

۱۰

جمعه
مرداد

۱۳۹۹

31 July 2020
۱۰ ذی الحجه ۱۴۴۱ (تعطیل)

عبد سعید قربان (تعطیل)

در پایتون با این داره نوع bytes می دوام از حرف ظایر برای اینها
رسانی، یا از نداس (bytes) استفاده می کنند. از استفاده از نداس
نام است که سیستم در دارای آن را منصف نمایم و بهتر است
داره های عددی را نیز به شلی یک شی لیست ارسال نماییم.

به این حالت کاربرد جزء اسلی نباید.

خطا لایست بی اندی:

3 b2 = bytes([9F]) a

نکته ۱: برای تبدیل نوع bytes به str نیاز به کد شانسی `decode()` داریم؛ این کار را بتوانی با استفاده از متود `str()` با مشخص کردن سیستم کوڈ طاری به انجام بگیر.

نکته ۲: همانند لیس `()` و bytes این باینری ابیاد نوع `bytearray()` از لیس `bytearray()` استفاده می‌کرد. (این نوع در واقع برخلاف تغییر پذیری از نوع bytes)

* جمع بندی:

انواع دنباله (Sequence Types) -

`bytes` ← ② `str` ← ①

`tuple` ← ④ `bytearray` ← ③

`list` ← ⑤

انواع تغییر پذیر (Mutable Types) -

`list` ← ② `bytearray` ← ①

`set` ← ④ `dict` ← ③

برخی دیگر

`None Type` ← ② `Zip` ← ⑦

نکته: دسته‌های موجود در پایتون ورثت ۲ شامل:

لے ۱. اسلی واره‌های باینری class 'str' ←

لے ۲. دسته‌های یونیکد class 'unicode' ←

اما دسته‌های موجود در پایتون ورثنا ۳ شامل:

لے ۱. دسته‌های معمولی (اسلی و یونیکد) class 'str' ←

لے ۲. داره‌های باینری class 'bytes' ←

لے ۳. داره‌های باینری (تغیرپذیر) class 'bytearray' ←

۱۳

۱۲

۱۴

۱۳

۱۵

۱۴

۱۶

۱۵

۱۷

۱۶

۱۸

۱۷

۱۹

۱۸

۲۰

۱۹

۸

* دستورهای نسل در پایین

در حالت عادی جریان اجرای یک بنامه، روند تابعی را ر به این نسل ل کردها سطح به سطح، از بالا به پایین خوانده و اجرای سئوند دستورهای نسلی امدادی برای نسل یا تغییر این جریان تابعی است. با استفاده از این دستورهای توان اجرای یک است. بلکه سطح تغییر از در راه رزمان اجرای این سطح بمقابل نباشد از اجرای بلکه صفت نظر خواهد شد یا می‌توان شرایطی را به وجود آورد که اجرای یک بلکه را زیان چنین بلکه شخصی انتخاب کرد و همین‌ای توان اجرای یک بلکه را چنین بار تلخان کرد.

۱۵

* دستور انتخاب (تصییم) :

لے دستور محظوظ

۱۶

با استفاده از دستور انتخاب می‌توان از حسب شرایط بنامه در زمان اجرای تغییر از این یک بلکه دستور اجرای سئوند یا خیر و همین‌ای ازین را یا چند بلکه دستور ندام یک انتخاب و اجرای کرد.

۱۷

برای تین دستور انتخاب در پایین گذاشته شده است که می‌تواند به سه شیوه از انتخابی، دو انتخابی و چند انتخابی سازه کرد.

۲۰

۱ ساختار سازه (رتک ایناپ) :

این ساختار یک استودیو است که در سرینه آن یک شرط (condition) بررسی می‌کند و نهاد صورت که این شرط برقرار باشد بدنم اجرا خواهد کرد، در غیر این صورت منسخه اجرای این ساختار (ها) بدنم صرف نظر نموده و به سرعت نیستن استودیو بعد از این ساختار می‌رود. این ساختار با استفاده از دستور if که الدویی مشابه یا نیز برابر سازی می‌گردد:

if condition :

statement Block

منصور از شرط عبارت است که می‌توان آن را به یکی از مقادیر های بولی (False / True) ارزیاب نموده و در اینجا اگر شرط برابر True ارزیاب نردد پس از بدنم ریستور گز اجرا می‌گردد.

نمونه کد:

1 $a = 5$

2 $b = 3$

3 if $a > b$ and $a > 0$:

4 print ("a is positive and greater than b")

>>> a is positive and greater than b

نکت: می‌توان از عملکردهای منطقی (not / or / and

بلکه بررسی برقرار بودن یا نبودن همینها جزئی شرط بروز گفت.

نکته ②: عدد یک و سام اعداد مختلف صفر در یا سو نا برابر مقادیر بولی True و عدد صفر و اسیا و خالی نا برابر مقادیر False انتسابی ای آردن

نکته ③: میتوان از ساختار If به شکل تعددی (nested) نیز برو

برای درین حالت بدن دستور If: حاوی بلک یا چند دستور

If: دیگر میتواند که اینها بین میتوانند حاوی دستور های

If: دیگر درینه خود را داشتند.

نمونه کلا:

1. a_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

2. if (n := len(a_list)) > 5:

print(f"List is too long ({n} elements")

>>> list is too long (10 elements)

۳ ساختار حمله با شرط `else` (روابط):
 با استفاده از لغه Python `else` می توان بالا را با اجراء حالتی که شرط برقرار نیست (نمانده شرط) برای مقدار بولی `False` ارزیابی نمود (تیکن الد).
`else` یک بخش جدا است که سایید و درنه منصوص به خود ندارد.
 این سایید می بایست فاقد هرگونه شرط باشد.

نمونه کد:

```

1 a = 7
2 if a in [1, 2, 3]:
3     print("a is in list")
4 else:
5     print("a is not in list")
6
7 a is not in list
  
```

۳ ساختار همراه با `elif` (چند انتخاب):
 دستور `if` را می توان لست شرط را و بخش های ایشان را با شرط های لوانه ایجاد نمود؛ به این صورت که ابتدا شرط بخش `if` بررسی می شود و خانه باید `True` ارزیابی نزدیک شرط مربوط به نمیشاند `elif` بررسی می شود لذا باز هم بررسی `True` نمی شود.
 شرط بخش `elif` بعده بررسی جواهد شد و به همین صورت ادامه می یابد؛ در اینجا از همین دام از شرط ها (`if / elif`) باید

٨- نسخه اولیه بینه مربوط به بخش else (در صورت وجود) اجرا نمی شود . الموارد این ساختار مانند زیر است :

٩- if condition - 1 :

١٠ statements

١١ elif condition - 2 :

١٢ statements

١٣ elif condition - 3 :

١٤ statements

١٥ else :

١٦ statements

۱۷- تعداد بخش های elif اختیاری است و محدود بیان رسان

و وجود ندارد .

۱۸- در این ساختار نزد وجود بخش else اختیاری است .

۱۹- در این ساختار بخش های بینه ای ارسالا به یا این بررسی می شوند و با True (ارزیابی اشکن) شرط مربوط به آن اجرا و از بررسی دیگر بخش های مورد انتظار بررسی شوند .

۲۰- خانه های متصدی را می توان شرط های مورد انتظار بررسی شوند .

۲۱- مسئولیت از جزء مستور (من برشل مسئول استفاده ننمایم)

مسئلہ ۸: برنامہ اک بنویسید جسے ۳ عدد وارد کرنے کا سطح کاربر
کے آنکھاں می باشد۔ \min

```

1 x = int(input("x : "))
2 y = int(input("y : "))
3 z = int(input("z : "))
4 Min = x
5 if y < Min :
6     Min = y
7 if z < Min :
8     Min = z
9 print(Min)

```

* دستور match / case *

در صورت لہ سابقہ برنامہ نویسی بانیان ہائی دلیل چھپوں کے

و فارق را لستہ بائیسید ہے اباد دستور Switch یعنی اسنا

ھسپتہ (تاپیٹ) اذ انسخہ ۱۰. ۳ پاسون ان دستور در زبان

پاسون پیازہ سازی نئوہ بود و تھرا می روائیم اذ شاخار if

استفادہ نہیں۔ ولی لئون پاسون اذ ساختار مسابی می نام

پیسبانی می لئد لہ ساختاری match / case

دارد:

۸. match value :

case matching - rule - 1 : statement - 1

case matching - rule - 2 : statement - 2

case matching - rule - 3 : statement - 3

⋮
⋮
⋮

۱۱. این رستور یک مقدار را دریافت کی لنگر با الگوهای ریج شده.

۱۲. توسط case طبقت کامی رهبر (از بالا به پایین) و با

۱۳. خوب عمل انطباق صونق، رستورهای مرتبط با آن را

۱۴. اجل نرده و سیس بیان کمی یابد.

۱۵. نت: هر بخش case کی تواند بیش از یک انطباق را بررسی کند.

۱۶. بلکه این منظور می تواند از کارالت ① بلکه جداسازی الگوها استفاده نمود.

۱۷. همون کد: یعنی نصیحت زوج بودن یک عدد

۱۸. ۱. list = [۴, ۵, ۶, ۲, ۰, ۱, ۳]

۱۹. ۲. first_num = list [۰] ➡ ۴

۲۰. ۳. match first_num :

۲۱. ۴. case ۰ | ۲ | ۴ | ۶ | ۸ :

۲۲. ۵. print ("The number is even")

۲۳. ۶. case ۱ | ۳ | ۵ | ۷ | ۹ :

۲۴. ۷. print ("The number is odd")



نکته ۲: همچنان می توان این case بسیار مرض نیز برای این

ساختار سنظر رفت، بلکه برداشت مقادیر زیاد که با

هم یک انبوحه ای موجود تطابق داشتند. بلکه بعد

بسیار مرض انبوحه Case استفاده می شود.

این انبوحه ساختار با هر مقادیر تطابق دارد می شود و می باشد

برای جلوگیری از خطاهای متصفح، حتی این عنوان آخرین

عملیات داره شود Case

نکته ۳: در بعضی Case حتی می توان از دستور if نیز استفاده نمود.

برایین ندادرین یا سینیس از دستور if در جامعه پایسون،

عملیات guard می شود. در این ساختار متغیر

له درست است if مولد می شود، همان مقادیر ریاضی است.

این متغیر می تواند هر نامی راست باشد ولی حتماً می باشد

با متغیر دوست از دستور if هماناً باشد.

در این سریع جایگزین از یا ب دستور if برابر مقادیر True

باشد، دستورهای Case اکن اجرایی شود و در غیر این صورت

انبوحه Case بعد از مورد برداشت قرار خواهد رفست.

ترفند: لفظ چند و عددی در یک خط: (ربات مثال ۳ ورودی)

$a, b, c = [float(x) for x in input('') .split(',')]$



* دستور تلار *

گاهی نیاز پیدا می کنیم که بدلای را چندین بار پشت سرهم اجرا نماییم . به ساخته تلار ، حلقه Loop دستور می شود .

لے دستور while

این دستور می تواند حلقه تلار است را یک سط را در سریعه خود بررسی کند و خانه سط را با برابر مقدار True ارزیابی کند ، دوباره به سریعه برگشت و سط را بررسی کند اتمام اجرای بدن دوباره به سریعه برگشت و سط را بررسی کند لکن اگر سط حضوز حجم بصرار باشد از تو دستور های بدن اجرایی برند در حالت اعادی یعنده تلار اجرایی بدن تازه تر می شود سریعه برابر ارزیابی کرد ادامه خواهد یافست .

النچ این دستور به صانته بایسنا است :

while condition :

statements

سط هموار می بایسیست از دوست بدن سلسله سود به دوست اول لکن در مرحله خاصی برابر مقدار False ارزیابی کرد ، بر غیر این صورت یک حلقه بی نهایت ایجادی شود از سریعه جدا نمی تواند از اجرای آن

1 2 3 4 5 6 7 8 9 ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

press ctrl + c to stop !!

خارج شود .

نوونه کد ۱ : ماسب فاکتوریل عدد ۱۰

```

1  a = 10
2  n = 1
3  while a >= 1:
4      n = n * a
5      a = 1 # a = a - 1
6  print(n)  → 3628800

```

نوونه کد ۲ : ماسب دنباله فیبوناچی تا مقدار n

```

1  n = int(input("Enter a number : "))
2  i = 1
3  a, b = 0, 1
4  while i <= n:
5      print(a)
6      a, b = b, a+b
7      i += 1

```

+ دستور : continue

این دستور در هر نقطه از بخش برنامه که اورده شود، دستور هایی بعدی از آن ناریه رفت، می شوند و جیان اجرا بر ابتدای حلقه یعنی بخش اولین بخش ای انجام می شود.

+ دستور : break

این دستور در هر نقطه از بخش برنامه که اورده شود، دستور هایی بعدی از آن ناریه رفت، می شوند و جیان اجرا از حلقه خارج می شود.

+ دستور : close

در while نیز می شود از بخش else استفاده نماییم؛ به این صورت که اگر حلقه به صورت طبیعی پایان یابد، (رنم توسط دستور break) آنها برنه else اجرا می شود.

+ دستور : pass

امانی بلک دولوپرهای این زبان مراهم می کند تا در صورت نرم بخشی از آن دستورات را خلی یک Function، ساخته حلقه های تکرار و غیره را در آینده پیاده سازی نمایند به طوری که سورس دیده نظر از نظر سینتکس مسلسل نداشته و به راحتی اجرا شود اما بارسیدن این دستور pass تاکنچه اینها نسود.

نونه ده، آیا n يك عدد اول است یا خير؟

```

1 n = int ( input ( "n : " ) )
2 i = 2
3 if n > 1 :
4     while i < (n // 2) + 1 :
5         if n % i == 0 :
6             print ( n , "is not a prime number! " )
7             break
8         i += 1
9     else :
10        print ( n , "is a prime number! " )
11 else :
12     print ( n , "is not a prime number! " )

```

17

18

19

20

for مجموعا برای تعداد تکلار مشخص

۲۸

سند شنیده
میرداد

۱۳۹۹

18 August 2020
۱۴۴۱ ذی الحجه ۴۸

گودزاری امروز کوارای بارگردان شاهد ۱۴۲۲ (۴۷) هـ

لهم رستور for

این رستور مطلب یک حلقة تکلار است له بر اساس تعداد عضو های یک سی نسباله یا در حالت تک تک یک سی تکلار لئنه (iterator). اجرای رستورهای بدن را تکلاری لئد.

اللئوں این رستور به تسلیل زیرینی باشد:

for target in object :

statements

هر حلقة for دیقا به تعداد عضوهای سی object تکلار می لرید.

هر بار یک عضواز نسباله (یا تکلار لئنه) object با حفظ

مرتب اعضا به متغیر target انساب کاره می شود و یک

مرتبه بدن اجرایی لرید؛ این لعند تا پایان ہمایش عضوهای

object الامه می یابد، از متغیر target می توان در داخل

بسنر استفاده لرید و در مرتبه لئست اجرای حلقة به عضویلم

و با احلاهای بعدی به عضوهای بعیی از object اشاره

خواهد داشت.

18



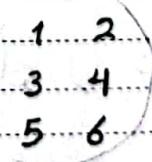
$$l = [(1, 2), (3, 4), (5, 6)]$$

19

for a, b in l :

20

print(a, b)



نکته ۱: در حالت اعماقی باید تئوری پیشنهادی، بینهای آن پیاس سی سووند:

۹ `1 d = { 'name': 'John', 'job': 'designer' }`

۱۰ `2 for key in d:`

۱۱ `3 print(key)`

name

job

۱ `2 for key, value in d.items():`

۱ `3 print(key, value)`

name John

job designer

نکته ۲: مانند حلقة while، راهنمایی تولن از دستورهای break

استفاده کرد. همین ب حلقة for می تولن

سامان بخش else باشد.

* تابع range(stop):

این تابع یک شی از نوع range تابعی است؛ این شی ب

ذی بالر تغییر نماینده است تا سه مولو از آن باید پیاس در

حلقه for استفاده می شود.

با تبدیل شی range به نوع لیست خواهیم دید که این شی ب

ذی بالر مربا از اعداد صفر تا آرلومان stop رو نه خود آن)

است: (آرلومان stop می باشد عددیم منتهی باشد)

نکته ۱: این نایاب را بتوان به صورت رو آر دومن (range(start, stop)) نیز خواهش نمود که آردومن یکم عدد آغازین زیبالترا تعیین می‌کند و کی تواند یک عدد منفی باشد.

در این حالت بتوان آردومن اسهم نیز برای تعیین کام یا ناصله سی اعداد بجهه ترتیب:

(هر سه آردومن می‌باشند اما نوع صحیح باشند)

نکته ۲: برای تعیین آردومن stop منفی می‌باشند آردومن کام را نیز به سلسله منفی تعیین نمود.

نمونه کد: ۱. list(range(2, -10, -1))
 ۲. [2, 1, 0, -1, -2, -3, -4, -5, -6, -7, -8, -9, -10]

نکته ۳: می‌توان سیم حلقه for را مستعین به یک سیاستا مدار داد:

۱. $l = []$

۲. for x in range(5):

۳. $l.append(x ** 2)$

$\Rightarrow l = [0, 1, 4, 9, 16]$

مانند خلیج ساده بخوبی بلنویسی کرد

$[x ** 2 for x in range(5)] \rightarrow \text{list comprehensions}$

* تابع enumerate (iterable)

علاوه بر تابع range، حلقه های for می توانیم از تابع enumerate نیز استفاده کنیم. این تابع یک سی دنباله یا تکلیر کننده را به عنوان آردومن دریافت می کند و یک سی از نوع enumerate برای لردن، با تبدیل این سی ابزار می کند که می تواند این سی عضو های آردومن و روی خود را به شکل جستجویی بر همراه آدرس معرفت کن. همان خوب است.

(index , value)

استفاده از این تابع در مواقعی که بحیثیت می دنباله غیر عددی یا بجزی اندیس دنباله حلقه را در نظر نداشت، باشد مفید است.

نوشته کرد: 
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
 $s = "python"$

for index , value in enumerate (s) :

print ('%s' % s % (index , value * 7))

>>> 0) pppppppp

1) ffffffrfrf

2) tttttttt

3) hhhhhh

4) 00000000

5) nnnnnnnn

نکتہ : این تابع ہمچنین یہ اردو مان اختیار با جنم start دارد کہ با مقادیر دارن بہ ان می توان عدد ابتدائی سیاریں اندیسہ اسها را تحسین نہ کرو؛ مقادیر پیش از مرض این اردو مان عدد حفظ کرنا است۔

10.

* تابع انتہائی : random

یکی از تابع انتہائی های Built-in (پیش از مرض) ہے ایسا کہ نیازی به نصب ندارد۔

12.

مائیل random در واقع براں ایجاد انتہائی های تصادفی کے یا یہی استفادہ می سوو۔ این انتہائی تصادفی می تواند عدد، رشتہ، لستا و ... باسے۔

14.

برای سرعی میں ھم تابع انتہائی حاصل برداری کرنے کا لائنا رہیم۔

15.

① import random

16.

② from random import ...

متود (۱) random () ← ایجاد عدد تصادفی با تعداد اعشار بالا بین ۰ و ۱

متود (۲) uniform () ← خوبجا ہمانند random با این تعلوادتا

کہ این عدد اعشاری بریلے محدودہ عدد کا وجود داری کہ ماں بہ عنوان

کے اردو مان ہا مستحصلہ لریم۔ (ریلے عدد اعشاری تصادفی را بر محدودہ بھی لرلاند)

19.

متود (۳) randint () ← یہ عدد صحیح تصادفی را بر محدودہ بھی لرلاند۔ نعم

متود (۴) randrange () ← اس بنا وہیم بریلے محدودہ مستحصلہ کام مستحصلہ

ملتبر (۵) choice () ← ایک ایسا عورت تصادفی از ایک لیسٹ ہائے ایکل دستے بھی لرلاند

20.

متود (۶) shuffle () ← جاں عنامر لیست را تغیر دیے دست جریدہ با ہمسوں میں

* تابع (Function) *

تابع به بلای از دستورات لفت می‌سود که بلای به اجرای آوردن نیازمند خواهد بود و این خواصی که تواند پیشتر آنکه یک بار اجرا شود . تابع که تواند به هنگام خواص خواهد بود که این را دریافت کند و در صورت لزوم مقادیر اینکه عنوان شده برگرداند .

تابع جایزه‌یابی نخواهد تواند بزرگ‌نمایی است که با یک بار توسعه و چنین بار خواهی واجهات آن می‌تواند از پیچیدگی برنامه جلوگیری و تغییر آن را آسان نمود . استفاده از توابع باعترف بالابردن ممکنیت استفاده مجدد از آنها می‌سود و اضطرابی را نیز کاهش می‌آورد . تابع ابزاری بلای خود را منطبق بزرگ‌نمایی واحدهای اجزای توجهی بلای سهل ساخت بزرگ‌نمایی است .

مفهوم تابع از علم ریاضی وارد بزرگ‌نمایی شده و به معنی ابزاری است که می‌تواند سیمه (یا تابع) را براساس داده‌های ورودی محاسب نماید . جنایت که این نخواهی اساسی بزرگ‌نمایی باشیم (یعنی توسعه یابد) ، به این سیمه توسعه « بزرگ‌نمایی تابعی » (Functional programming) دویند .



← همچون زبان‌های Haskell ، Scala ، Lisp

ناآسوعی حسنی (تعطیل) - روز مبارزه با تروریسم (انفجار دفتر نخست وزیری به دست منافقان و شهادت مظلومانه شهدان رجایی و باهنر - ۱۲۶۰ هش)

تابع در زبان های دیلر همچوں pascal، C، C++، C#، جنیز و جبود راردن، البت این زبان ها درسته برنامه نویسی دلستوری مترادف دارند و در پیش از Imperative programming.

این ها سبب است بر تابع دارند کامل معاونت از جنیز است. در زبان های برنامه نویسی تابعی وجود دارد.

برای چنین برنامه نویسی تابعی با استفاده از پاسخنامه توأم است Functional programming in python از استاد ادوارڈ آرلی O'Reilly مراجعت نماید.

در دنار اصطلاح تابع در برنامه نویسی، اصطلاح متساب دیلر نیز به عنوان بعل (Procedure) وجود دارد. بعل و تابع در ساختار سسی بلبل هسته با این تفاوت لر بعل ها مقادیر برنجی ارائه دند. در پایتون سینتیسی برای تعریف بول محدود ندارد ولی به تابعی که مقادیر برنجی ارائه دند، بول نفته هی سود که اسیه است چرا که توابع در پایتون انت هر ساری یک مقادیر برعی ارائه دند حتی اگر این مقادیر None باشد.

* سینتکس تابع :

سینتکس تابع در پایتون همانند هر ستد مطلب دیگر شامل یک سری از دستورات است. بخش ساینده شامل کلمه def، یک نام برای شفاه کاربر و پارامتر باشد. لیکن باید عمل تملیفی بارامترهای تابع را نمایش داد. (هر تابع می تواند هم یک یا جزو بارامتر بیندازد.)

def function-name (param1 ...) :

statements

هنلای له اجران برنامه به کمک کلمه def می بدم. ابتدا یک سئی تابع ایجاد (class 'function') و سپس از نام تابع بآن ارجاع داره می شود.

بدنه تابعی له تابع خارجی ندارد، اجر نمی شود. بلکن خارجی تابع از نام تابع + function-name باشد، می باید در صورتی له در معرفت تابع بارامترهای اول را داده شده باشد، می بایست این خارجی آرگومان های مناسب باشند. با این باره ایز ارسال نردد.

function-name (arg1 ...)

ملاحظه: در بحث توابع، به متغیرهای لهم دوستی ایند تابع تعریف می‌شود.
 ۸ پارامتر parameter و به رده‌های نهنگام فراخوانی تابع
 ۹ ارسالی درون آرگومان Argument لفظی شود. به این
 ۱۰ هر پارامتر در نظر لغتش شده در تعریف تابعی باستاید.
 ۱۱ آرگومان نظریه به آن ارسال شود. در واقع آرگومان‌ها مقادیری
 ۱۲ هستند که خواهیم به پارامترهای یک تابع انتساب دادیم.
 ۱۳ چیزی را که هم‌نام بولان آرگومان‌ها و پارامترهای نظری
 ۱۴ وجود ندارد ولی وجود هم‌نامی باعث خواهد بیست و دویمی شود.
 ۱۵ بینه تابعی تولید حاصل لهم لیکی return نیز باشد.
 ۱۶ در واقع return دستور است که در هر جایی از بینه اورده شود، اجرای تابع در آن نقطه متوقف و مقداری (شی ای) را به عنوان نتیجه به محل نماخوانی تابع بازی می‌راند.
 ۱۷ return value زل نماید. None باز ندارنده می‌شود.

def function_name (param1, param2, ...):

 return value

نکته: نکته: در این تابع return دستور ذکر نشود، مفسر پایتون به صورت ضعی دستور return None در نظر می‌گیرد.

در ادامه به ایجاد انواع سیستم‌های مخوب مبتنی پایه‌یون / تطابق آرکومان‌ها با پارامترهای Argument Matching

تابع می‌پردازیم .

۱) لئے سیستم معمول لئے تالنون استفاده می‌کوئیں یعنی به صفت
برآورده پارامتر کے آرکومان نظری ارسال کرو . عمل تطابق
در این سیستم براساس صفت آرکومان‌ها انجام می‌شود
له دریجہ می‌باشد تا رسید آرکومان‌ها، مناظر با ترتیب
پارامترها در ساید تابع باشد .

```

1 def f(a, b, c):
2     print("a = ", a)           دو دو a = one
3     print("b = ", b)           ط دو b = 2
4     print("c = ", c)           دو دو c = [3, 33]
5
6 f ("one", 2, [ 3, 33 ])
    
```

۲) لئے سیستم name = value ← در این سیستم آرکومان‌ها
برآورده پارامترها انسار داره می‌شوند و اذ آنها له عمل تطابق
براساس نام پارامترها انجام می‌شود یعنی موجودیت یا ترتیب
آرکومان‌ها اهمیت ندارد .

نکته: می‌توان از سینتیس (۱) و (۳) به صورت ترتیبی نیز استفاده کرد.
 نقطه ناید توجه داشت آرگومان‌های که عمل تطابق آن‌ها
 وابسته به موضع استا با برایست تهیبا موارد جمل تران
 آن در ابتدا تعداد رفیم.

```

1 def f(a,b,c):
2     print("a = ", a)    a = 1
3     print("b = ", b)    b = 2
4     print("c = ", c)    c = 3
5
6 f( 1 , c = 3 , b = 2 )

```

لهم سینتیس * iterable در این سینتیس یک سی از (۳)
 نوع تکرار پذیر (iterable) مانند انواع رشته، تپل، لیست
 و ... به توسط یک کارالت ستاره * نشانه‌داری شده است.
 به تابع ارسال می‌گرد. در این صورت براساس ترتیب موضعیت
 اعضای درون سی تکرار پذیر به باره‌های تابع اختصاص می‌یابند.

```

1 def f(a,b,c):
2     print("a = ", a)    a = 1
3     :
4     My-list = [ 1 , 2 , 3 ]    b = 2    c = 3
5     f( * My-list )

```

لهم سینتیس \leftarrow dict \leftarrow دادن سینتیس با پارامتری سی (۴)
 ریلیشن \leftarrow توسط دو کاراکتر استار؛ * نشانه تعاریف شده
 است به تابع ارسال می‌شود. لیکن این سی ریلیشن‌ها بایست
 همنام با پارامترها تعریف شده در ساینده تابع باشند؛
 پس از مراحلی تابع، این سی ریلیشن باز می‌شود و به اساس
 نام لیست در چیزی که مقدار آن بخوبی آن پارامترها
 تابع مقدار دهی می‌شوند.

1	def f(a, b, c):
2	print(a, b, c) عدد ۱ ۲ ۳
3	
4	b = {'a': 1, 'c': 3, 'b': 2}
5	f(**b)

این چهار سینتیس برینجین (دویولن) ها در هنلار مراحلی تابع بست
 نی ازند و در نهایت آن‌ها می‌باشند تعداد آن‌ها می‌باشد و بدین
 با تعداد پارامترها تعریف شده در ساینده تابع برابر باشد و بدین
 است له در دو سینتیس پایانی نازم است تعداد اعضای سی
 هنلار نیز یا تعداد چیزی که مقدار آن بخوبی آن پارامتر ریلیشن
 با تعداد پارامترها می‌باشد.

1 ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱

پارامترها تابع برابر باشند.

در ادامه بیهوده سنتیس هایی در لین نمایند. می پوچیم یعنی هنام
یعنی بارامتر های تابع نقضی بازند.

لئے سیلس مجموعه تالون اسناده می کردیم یعنی به صراحت
لئے تک پارامترها را تعریف ننمیم.

لئے سیلس یعنی مقدار پیش افرض باید پارامترها
می تواند هنام یعنی هر پارامتر را تعریف کنند، مقدار را
نیز به آن انساب دارند. در لین سلطیح آن ایجادیان نظری با آن
پارامتر ارسال نلرزد، مقدار پیش افرض آن پارامتر در نظر نرفت.
خواهد شد. (به این توان پارامترها، اختیاری لفظی شود)

def chaap (text = None) :

if text :

print (text)

else :

print (" nothing ! ")

chaap (" python ") ➔ python :

chaap () ➔ nothing !

نکت : پارامتر با مقدار پیش افرض نمایی توان در نظر پارامترهای اجباری را بروز رساند.

مقدار پیش افرض () تعریف کنند. لین سلطیح می باشد.

پارامترهای دارای مقدار پیش افرض را در نشانه امداد دار.

لهم شنیدنی این * name از دوستان های ارسالی را بس
 تاب یک شئ (Tuple) را گفتند (3)

این قابلیت در موقعیت نه تعداد اردومن های ارسالی مستحب است

است لطفاً نظر کنید.

```

1 def f(a, b=2, *args):
2     print("a = ", a)           a = 1
3     print("b = ", b)           >>> "FF"
4     print("args = ", args)    (3, 4, 5)
5
6 f(1, "FF", 3, 4, 5)
7 f(1)                      b = 2    args = ()
8 f()                        Error

```

نتیجه داشته باشید نه توان اردومن را با استفاده از سیو name = value ارسال کرد.

نتیجه داشته باشید بدهد این اردومن نظریهای پارامتر هایی را پس از پارامتر سازه دار می دارد.
 name = value لفظ داشته باشید به صورت

ارسال نماید def f(a, *b, c)
 چنانچه جزئیاتی مقداری بدست موقعت نمی ماند هر تعداد اردومن را دریافت می کند

Key word - only Arguments

نام ۱۷ شهریور و کشتار جمعی از مردم به دست مأموران ستم‌شاهی پهلوی (۱۳۵۷ هش)

1390

نکته ۳: در بسط این بحث لام است اضافه لرد به می مان ارسال
کار نومن به بخش پارامتر های یک تابع را ملزم به بوش
نام = name، لرد = return شیوه ای کاریت = * به عنوان
یک پارامتر نسباند استفاده می لرد بی این صورت به تمامی
پارامتر های بعد از آن تنها پاسخ به صورت مقادیر = نم
مقدار دهن سوند. (* در اینجا پارامتر نبوده و تنها یعنی
یک نشانگر برای دارد)

	def f(a, *, b, c):
13 2	print(a, b, c)
1 3	
4	f(1, b=2, c=3) ➤ 1 2 3
5	f(1, 2, c=3) ➤ Error

16- لے سیٹنگز name کے تمام اڈیوٹان ہائی
17- key : value کے مقابلہ میں اسکی دیلیٹری
دریافت کی جائے۔

positional - only parameters

۱۸

سندیه
شهریور

۱۳۹۹

8 September 2020
۱۴۴۲ محرم ۱۹



نکته ۴: از نسخه ۳.۸ پایتون سینتکس جدیدی به پایتون اضافه شد.

لرید که این املا را به صاف دهد تا بتوانیم تعدادی پارامتر را

در تعریف یک تابع بجبور به ارسال آرگومان تناظر آنها

بر اساس موقعیت نماییم. و به تبادل این املا ارسال آرگومان

به روشن مقدار = نام را با آنها غیرفعال نماییم:

این سینتکس در موقعیت نام پارامترها در آینده خواهد بود. این است

دست خوب بخوبی سوزن مفید خواهد بود. زیرا بخش اعلیٰ پیلر

که بنا بر این نیاز به تبعیض درین حالت شواهد داشت.

در این سینتکس کافی است در تحریف پارامترهای تابع،

پس از نام پارامترهای مورد نظر خود از کارتر یک به عنوان

یک پارامتر نسائلن انتقاده نماییم. در این صورت ارسال

آرگومان به روشن مقدار = نام برای تمام پارامترهای نسائلن از

آن منع نمود. (نام در اینجا پارامتر نبود و نسائلن یک

نشائلن رک دارد.)

نائلن رک دارد.)



def f(a, b, c, d, e, f)

ارسل آرگومان تنها ارسال آرگومان پس این ۲

به روشن مقدار = نام به هر یوئیس زیر پارامتر به روشن مقدار = نام

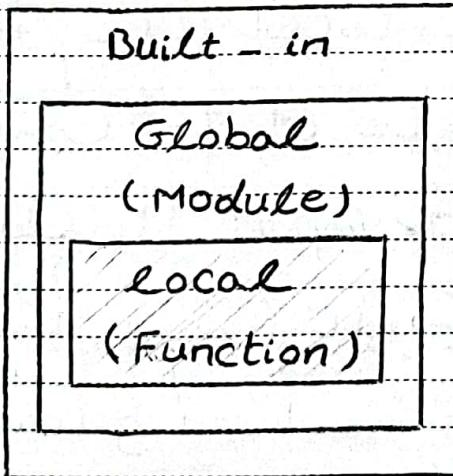
جگانی باشد موقعیت و نام = مقدار منع X

در زبان برنامه نویسی پایتون، تابع یک موجودیت first-class است. آن را می‌توان مانند دیگر اسیاء به صورت پویا ایجاد یا تابودار، به صورت آردوینو، به توابع دیگر ارسال نمود،
به عنوان سبب توسط return باند ازدند و در سببی تواند یک تابع را دهن برقراری دستورات شل (if، while و...) یا دفعات یافته تابعی دیگر یعنی فراخواند نمود.
بر میان دیگر زبان‌های برنامه نویسی که با توابع خود مثل سریوند در جزء اول برخورد کردند، در واقعیت یاد ریاضی مورد نظر انداشتند و تئکی ها سایر دستورات کنند.
 ۱. اختصار برای مقادیر تابع برای مسحیر
 ۲. می‌سازند تابع به عنوان آردوینو تابعی دیگر
 ۳. برای راندن یک تابع به عنوان مقادیر برای تابع دیگر
 ۴. ذخیره در تابع در رده ساختارهای مختلف
 ۵. بتواند متدها و تئکی های خاص خود را نام دارد.

* فضاهای نام و جوئه (Namespaces and Scope)

بر هر برنامه پایتون تعداد زیادی نام وجود دارد که بطور نسبتی می‌توان به: متغیرها، نام تابع، نام کلاس‌ها و ... اشاره کرد.
 بدینی است که براحتی می‌توان این اسامی را با نام است (Names) مخصوص بخواهیم باشند، ولایت چنین امری ممکن است برنامه حقیقتی کار نماید.
 سعی است، در زبان پایتون بلکه درسته بسیاری و جلوگیری از تداخل نام‌ها، ساختاری با عنوان فضاهای نام در نظر گرفته شده است (Namespaces). هر فضای نام بخشی از نام‌های دیگر برنامه را در بر می‌گیرد. به صورت لیک فضاهای نام پایتون در سه سطح: تود توک (محلي) (local)، سراسر (Global) و Built-in (Built-in).

می‌شوند (در بخش متابع سطح چارچوب از این سیاست است) Enclosing (لفتی شود)



هر مثال پایتون یک فضای نام سراسری دارد که خود تکمیل می‌گردد. نسبت به فضای نام دیگر مادول‌ها ایزوک است. فضای نام تمام مثال‌ها درون فضای نام بزرگی ایجاد می‌گردند که به عنوان

نضایانام in - ~~گلایات~~ ساخته می شود و نام تابع توابع آماده مانند `(1) open` براین فضای مادر لفته است . ساختار توابع توی سطوح فضاینام باعثی شود که بدون نیاز به `Built-in` مانع خاصی در هرجای برنامه به توابع آماده `in - import`

دسترسی داشته باشیم .

هر مانع می تواند شامل تعدادی تابع و لاس باشد . یا مراحتی هر تابع یک فضاینام علی یعنی آن تابع درون فضاینام مارول صبوط ایندادی لرد و با پایان اجتی تابع نیز از بین اجتی رود .
بر سور للاس ها هم اتفاق مشابهی رخ می رهد . بر همین اساس می توانیم درون تابع متین های متفاوت و لی هم نام با متغیر های خارج از تابع بر مارول اینداد نمایم چنانه آن های درون فضاینام متفاوت - قلد لارند و از طرفی ب دلیل داخل بودن فضاینام تابع درون فضاینام مارول خود ، می توان این های خارج از تابع نیز دسترسی داشت .

تفصیل فضاینام مانع ها سبب به پذیری اینوک هستند . بنابراین برای دسترسی به نام های بدون مانع های ریلی ، ایندادی بایست آن مانع `import` نمایم له براین صورت با استفاده از نام مارول (بر شلل پسوند) تابل دستیابی هستند . برای نمونه دستیابی به نام `getcwd` له برای تابع از فضاینام ۵۵ ارجاع دارد :

نمونه رستایاب به نام دوستانل :
import os

os.getcwd() → function - name

nameSpace.global : module

اما استفاده از نام های یک مارک صفت حوزه چلوبت است بجای آن که فضاهای نام دیگری همچون توابع نیز وجود داشت و لیکن پیشنهاد مانند نام مارک وجود ندارد به توان نام های دوستانل این

فضاهای متفاوت را از پذیری نمی دارد. برای اینکه بدانیم هر نام مارک

در هر قسم ای از همان مارک چلوبت روبروستایاب تمدنی تمرید با معرفی دیگری به نام حوزه scope آشنا می شویم.

به صورت لی حوزه به نوای ای لفته عی شود که بتوان یک نام را بیرون از همین پیشنهاد و البته بیرون تداخل با نام های دیگر بکار برد.

(بحث حوزه صفات در داخل هر مارک مطرح است)

- موارد حوزه :

• بند مارک ← محدود نوای ای که خارج از بند توابع و ملاسها می باشد حوزه سراسری Global scope است

تجزیه رئیسی براساسی که واره سراسری در بند حوزه (یا فضای نام)

نیز با به سراسردهای داخل هر مارک لشاره دارد و نه سراسر برآمد.

(هرجا از زبان یا بیرون از واره سراسری Global رئیسی بیاد مارک

نکته ۸: بدنم دستورات `while`، `if`، `else`، `for`، `def`، `lambda` نامهایی
فضای نام `global` است و تعریف متغیر در این نویحه از برنامه
برون حوزه سراسری قرار گیرد.

بدنم هرتابع یک حوزه محلی `local scope` است و به صورت
بین مرضی نام متغیرهاست که درون توابع ایجاد می‌گردند درون
حوزه محلی قرار نرفتند بلکه بالهای `global`
یا nonlocal مشخص شده باشد.
مثالی بعنوان دو تابع انساب با همی از نام های موجود در
حوزه سراسری اینجا درست، می‌باشد از دستور `global` استفاده نیم.

1	<code>a = 0</code>	1	<code>a = 0</code>
2	<code>def My_func():</code>	2	<code>def My_func():</code>
3	<code>a = 3</code> <i># new a</i>	3	<code>global a</code>
4	<code>print(a)</code>	4	<code>a = 3</code> <i>* global a</i>
5	<code>a =>> 0</code>	5	<code>print(a)</code>
6	<code>My_func() =>> 3</code>	6	<code>a =>> 0</code>
7	<code>a =>> 0</code>	7	<code>My_func() =>> 3</code>
8		8	<code>a =>> 3</code>

در توابع تو در تو نیز مفعول ندارد، هر تابع له ملحوظی می‌شود ضمایمانی بجز برای آن ایجاد می‌شود و حوزه محلی خود را خواهد راستا.

دستور nonlocal در پاسوپت ۳ ایجاد شده و در توابع

تو در تو کاربرد دارد. هنلایی له بخواهیم داخل بدلن تابع انسابا

به نایی تعریف شده دریلی از توابع بروی آن اینام دهیم

می‌باشد آن این دستور برای مشخص کردن نام صور نظر استفاده

لیسم

12

و حقی از ضمایر استفاده می‌لیسم، مفسر پایتون ابتدائی باشد

حوزه و مضانام آن را تصحیص دهد تا بتواند سی ای له این متغیر

به آن ارجاع دارد رایسا لند. فرض می‌لیسم متغیری درون عبارت

در بدلن یک تابع به کار رفته باشد در این صورت مفسر

ابتداء حوزه محلی له مسجیو در آن وجود دارد برای بجز یافتن تعریف

متغیر جست و جوی لند و چنانچه نیابد به ساعت حوزه محلی تابع

بر روی آن (در صورت وجود) می‌رود و همین طور ایسوسی رخداد

در تهابیست حوزه سراسری مأول و پس از آن نیز in Built - in را

بررسی می‌لند؛ اگر هم یه سیب، ای بررسید یک استثناء

خواسته name Error رخ کار نکند.

Built - in

Global

local

* ارسال آرگومان

به صورت خودکار با ارسال آرگومان به تابع، معمولاً هایی علی، از انساب اسماً از توکان‌ها به اسمی پارامترهای مُجبر در سراسر این تابع به وجود می‌آیند. توجه داشته باشید که با انساب شوی اسماً جدید به پارامترهای تابع، عمل ارسال آرگومان بی‌نایند می‌گردد.

1 def f(a):

2 a = 2

3 print(a * a)

4 b = 3

5 f(b)

عمل ارسال آرگومان طبق تأیین

نکته مهم در ارسال آرگومان، توجه به چلوانی آن است.

در بین زبان‌های برنامه‌نویسی دو شیوه رایج برای ارسال آرگومان

است: `value by value` و `by reference`

که در شیوه `value by value` یک پی از مقدار آرگومان به تابع ارسال می‌گردد و در شیوه `by reference` با بعضی مقدار پارامتر متناظر در تابع، مقدار آرگومان

ارسال شده در خارج از تابع بدون تغییر باقی می‌ماند.

که در شیوه `reference by value` به جای ارسال یک پی از مقدار آرگومان، یک ارجاع از آرگومان به تابع ارسال می‌گردد. می‌توان این طور

متوجه نفت که پارامتر متناظر در تابع، همان آرگومان در خارج از تابع است.

در شیوه `by reference` با تغییر مقدار پارامتر در تابع، مقدار آرگومان در خارج از تابع نیز تغییر می‌کند.

این دو اندیشه های مرسوم در زبان های برنامه نویسی هستند ولی ارسال پایه است. به صورت خاص در زبان برنامه نویسی پایتون چنین است که در پایتون هر جزئی یک سئی است و تبیین ارسال ارلومان ها در هر سریط به صورت by reference انجام می پذیرد.

آن سوال سود لر علت تعاون رساناد چیست؟

پایه در اینم لر علت با ماهیت اسیاء ارلومان های ارسال مربوط است. ارسال اسیاء تغییرناپذیر Immutable به صافند انواع بولین، اعداد، رئست، و متویل به تابع، باعث بروز نقصان مسابه با اندیشه value by ref می سود ولی در صورت ارسال اسیاء تغییرپذیر Mutable به صافند لیست، دیلشنی، گروه، انلونگ نخواهد بود.

اسیاء تغییرپذیر در پایتون اسیائی هستند له بدون تغییر id آنها مقدر شان قابل تغییر است. جوچی تابع id باش هر شی بیانلر رساند مخصوص به فرد لکن سئی است له در راجح اسماف آن در حافظه نیز می باشد.

```

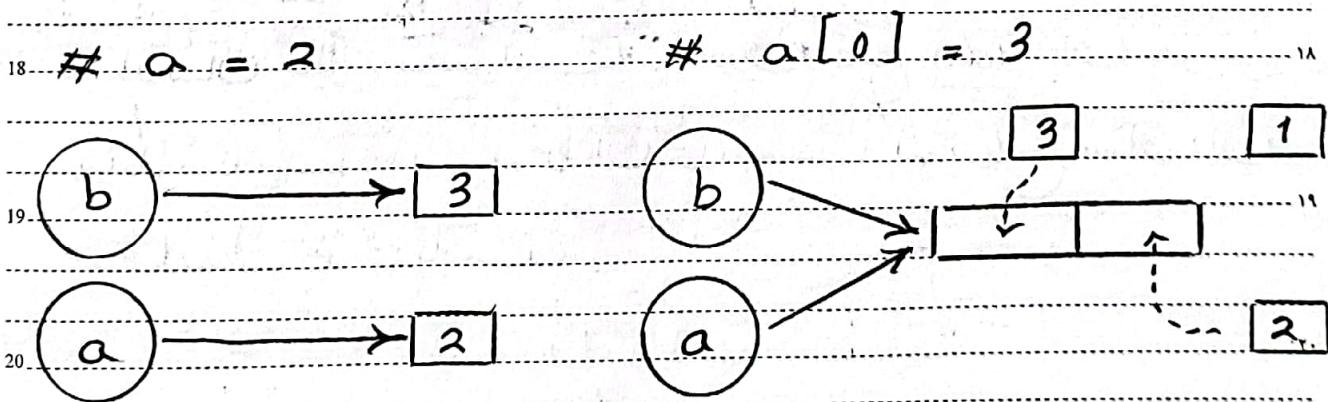
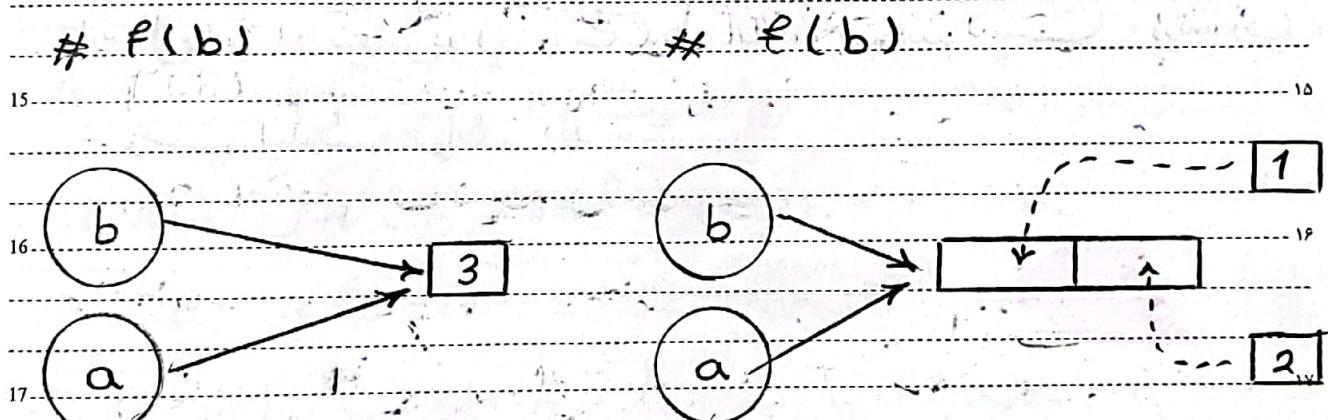
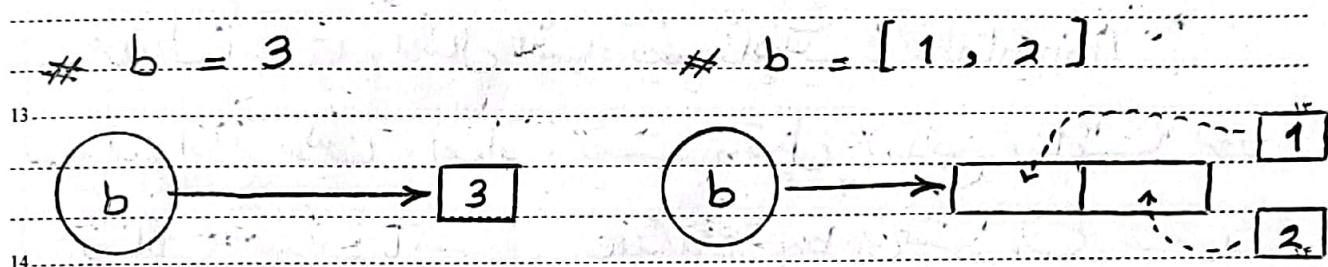
8 1 def f(a):
9 2     a = 2
10 3     print(a*a)
11 4 b = 3
12 5 f(b)    => 4
13 6 print(b) => 3

```

```

1 def f(a):
2     a[0] = 3
3     print(a)
4 b = [1, 2]
5 f(b)    => [3, 2]
6 print(b) => [3, 2]

```



بله جلویی از تغییر اسئاء تغییر پذیری رفوت تابع، می توان به تونه ای،
که لفتر شد (۱۵ تیر) یک بی این از این نوع انسانیت را ایجاد و
سپس به صورت آردومن ارسال کرد.

این گاهی و اصحاب نیاز است از مقدار تغییر یافته از محدودیتی که به تابع
ارسال می سود را نبیند بروز از تابع هم در اختیار باشند باشند

بله این منظور در بخش زبان های برنامه نویسی املان ارسال به

سیوه reference بوط بنابر خواسته برنامه نویس نماهم

سته است. بله مثال در زبان PHP این کار با مدارک دادن

یک & در پست یارا متر مورد نظر انجام می سود.

در یا یون حسن مالکی وجود ندارد، حداصل یک اسئاء تغییر نایدند!

و یعنی توالی با استفاده از املان بازیست جذبیتی توسط

رسنور return آن را یوئیش نمود. با استفاده از این سیوه می توان

هر تعداد از یارا متر های مورد نظر خود را به خارج از تابع انتمال داد.

توجه رائسته باشید که در این حالت رسنور return تمام این

اسءاء را در قالب یک سی تویل برخی ارائه دارد.

و این می تواند این را در اینجا بخواهیم

و این می تواند این را در اینجا بخواهیم

و این می تواند این را در اینجا بخواهیم

و این می تواند این را در اینجا بخواهیم

+ lambda : توابع ناشناس :

در زبان Python بیان ممکن نیست یا سیوں توابع ناشناس anonymous function

یا lambda functions توابعی هستند که می‌توانند هر

تعبار ارثومان را ساخت باشند و آن بدنم آن‌ها می‌باشد آنرا

سامان یک عبارت باشد... بلکه ساخت آن راستا تابع از

لهمه دلیلی lambda استفاده می‌سود... الیک ساختاری آن

نوع تابع به صورت زیر است:

lambda args : expression

در این الکو args معنی هر تعداد ارثومان است که با استفاده

از دامارویی از پلیملر جرایته اند و expression بیانگر آنرا

یک عبارت یا سیوی است که شامل دستوراتی همچون

while یا for می‌باشد.

کاربر اصلی lambda functions بآنست

این راستا تابع ستر بر مواجهی که می‌خواهیم یک تابع کوتاه‌مدید

به عنوان ارثومان به تابعی دیگر ارسال نمایم کاربر را زند:

برای نمونه کاربرد در sort، reduce، filter، map

لیست می‌شود که متدهای sort() ارثومان اختیاری باشند key کاربر می‌توان

با ارسال یک تابع که ارثومانی به آن عمل رحمه‌هی کاربر روی آنرا عضو

های لیست بورد نظر، پیش از مقایسه و مرتب سازی به آنجام

1 $l = [a, 'D', 'c', 'B', 'e', 'f', 'g']$

2 $l.sort(key=\lambda b: n.lower(b))$

3 $print(l)$

$\Rightarrow [a, 'b', 'c', 'd', 'e', 'f', 'g']$

+ شی تکرار لیننده (iterator)

تاں انواع نتیوالریبل iterable ہستے؛ در واقع بے اسیاب با این تکلیف کہ بتوان دھر لخطہ میں عضووں ان را دستیاب نہ کرو iterable لفڑی سود۔ المئ انواع آمادہ شی لہ رہیاں ہوں یہ اتناسیم یک iterable است؛ انواع شی رئست، لیست، توپیل، دیکشنری range، Zip یا یک بی شی فائل file دھر شی ان للاس کے خود تان بھر ہمارہ صورت ویرہ () iter یعنی دوسرے نہ اسے بل iterable ہستے۔

بر ایندہ یہیں ان مطالعہ دروس مربوط بھئی تکلیف عالیہ ساختاں للاس واستفادہ از میدھائی ویژہ رہیاں ہوں اخواہ ہید بعد دھر لان زمان می توانید خور بایارہ سائی مت ویرہ () next یک بھی iterator ایجاد نہایت۔ اس امر این مرحلہ ما خلیفہ ایجاد را بایک ماروں از تباہت اسٹاندارڈ پیسے خواہیم برد۔

۱. هی توانی باید شی iterator را نیا با استفاده از تابع آماده `(iter)` ایجاد کرد. این تابع یک شی iterable را به عنوان ارگومان در یافتن اسکنده بخواهد.

۲. یک شی iterator این مایل است که امداد کنی تواند عضوهای دفعات آن را بخواهد با استفاده از متود `(next)` بخواهد.

۳. ترد؛ این متود بار نخست خواهد عضویلم شی و در فتحمان بجای خواهی بود عضوهای بعدی را برای لردن.

۴. با خواهی بود تین متود را به انسان بخواهی؛ زمانی که این عضوی بخواهد برای اینکه وجود ندارد یک خطا (Exception) را ساخته است که بخواهد اینستا.

۵. با نام `stopIteration`.

۶. `l = [1, 2, 3, 4]`

۷. `itr = iter(l)`

۸. `itr.__next__()` \rightarrow ۱

۹. `itr.__next__()` \rightarrow ۲

۱۰. `itr.__next__()` \rightarrow ۳

۱۱. `itr.__next__()` \rightarrow ۴

۱۲. `itr.__next__()` \rightarrow Error : `stopIteration`

این ریتم همان کار است که در دستور `for` بجانب آن جای خواهد داشت.
 زمانی که از دلیل نسبالس باید پیمایش این دستور استفاده
 می کنید؛ `for` دوست `iteration` صفتی دارد که به بینت
 بینی و سیسما پیمایش یافته بین عضوهای آن را انجام می کند.
 در هر لحظه داده `iteration` دارای دو خصوصیت مسحیج است.
 پایان این پیمایش مسده و تکرار حلقه را بیان می کند.
 با استفاده از مثالی `for` tools می توانید `iterator` های
 بخانه ایت `infinite` یا بدون آنوقت ایجاد نمایید.
 برای نویسندگان `cycle` درون این مثالی، یک `iterator`
 می سازد که در انسانی پیمایش امتیاز می دهد و آن متو ب اینها
 سی بولسند و عضویتیم را برای اینها دارد.

دکوراتور (Decorator) +
 دکوراتور (Decorator) برای یعنی افزودنی سود به منظور پوشش
 (wrap) توابع یا لامس‌های دلخواه پیاده‌سازی شوند.
 دکوراتورها در پایتون ابزار بسیار کاربردی و مفید هستند که به
 برنامه‌نویس این امکان را می‌دهند تا با کاهش حجم کدنویسی و بعدها
 نگیرند. درینجا توابع و لامس‌های خود را فیلر و پرئال های آن
 را معرفی می‌کنیم. درین پنجه تبلیغ روی اعمال دکوراتورها به
 دلیل این است:

برای پوشش یک تابع بوسیط Decorator از سنتیسی مسابد
 @decorator_name @ بر بالای نیش سرایند استفاده

می‌شود:

```

1 def decorator_name(a_func):
2     pass
3
4 @decorator_name
5 def function_name():
6     pass
7
8     function_name()

```

نمودار این سینس @ + Decorator-name در بالا
نخست سرآیند یک تابع بای مفسر باشون اجباری نند تمام
سابه با سینس یا بین است:

wrapper = decorator-name(function-name)

wrapper()

هر چیز در بایشون یک سی است احتماً مقاهم بیمه ای به مانند

تابع؛ به خاطر طریق که تابع در بایشون یک سی جعلی است

تابع first-class است که یعنی می توان تابع را مانند ذیل

اسیاد به صورت آرگومان به توابع دیگر ارسال نمود. نمونه

که بالا نیز نیائی ارسال یک تابع function-name به تابعی دیگر

به تابعی دیگر است.

1 def decorator-name(func):

2 def wrapper():

3 print("Something is happening before the func")

4 func()

5 print("Something is happening after the func")

6 return wrapper

7 @decorator-name

8 def function-name():

9 print("Somthing !!")

function-name()



مناخطنا

همانطور که در لد قبل تابع مساحتده است. زلولوایرون چنایل
 روپیش (wrapper) بدلی توابع و پیلیس صاری می‌بیو جود
 می‌آورند. در هنلار مراخوان تابع function-name
 بنسر یا بیون مسوجه، آن شرطه است و
 به جای اجر، یک نمونه‌شی از آن را به decorator مستحق
 شده (decorator-name) ارسال می‌کنند و یک شی
 جدید که در اینجا با تابع wrapper مستحق شرطه است را
 بیافتد و اجر می‌کنند.

در مورد توابع دلایلی برای اسر نیز باید توجه داشت که در هنلار
 مراخوان تابع صوره طور و ارسال آرگومان به تابع، مفسر
 پایتون این آرگومان‌ها را به تابع decorator wrapper از آن
 ارسال می‌کنند.

می‌توان بیش از یک Decorator به دلایل‌ها و توابع خود
 اعمال کرد که در این صورت ترتیب اعمال این‌گونه این رکوردها
 باید مفسر یا بیون دلایل اهمیت است:

1 @decorator_3

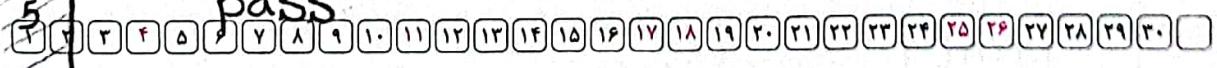
2 @decorator_2

3 @decorator_1

4 def function_name():

5 pass

6 function_name()



۸ wrapper = decorator - ۳ (decorator - ۲ (

۹ decorator - ۱ (function - name)))

۱۰ wrapper ()

۱۱ - همچنان می‌توان به دکوراتورها ارث‌گیران نیز ارسال کرد. در این

۱۲ حالت می‌توان پایه‌ی ارث‌گیران را بر تابع Decorator

۱۳ ارسال می‌کند و سپس حاصل را با ارث‌گیران دوباره تابع مورد نظر

۱۴ ضلیعه‌ی خواسته

۱۵ ۱ @decorator_name (parms)

۱۶ ۲ def function_name () :

۱۷ ۳ pass

۱۸ ۴ function_name ()

۱۹ temp_decorator = decorator_name (parms)

۲۰ wrapper = temp_decorator (function_name)

۲۱ wrapper ()

۲۲ Higher_order functions مطرح است به نام

۲۳ (تابع مرتبه بالاتر) و بر توابعی نفت می‌شود که اعمالی را روی

۲۴ توابع دیگر انجام می‌دهند یا یک تابع جدید را به عنوان خروجی برخی از آنند.

۲۵ ۲۰

↳ @functools.wraps

برهمن اساس یک مارپیچ بنام `functools` نیز در لیابانخانه است.
استاندارد پایتون تماری رفته است که یک سری توابع کمی و کاربردی
برای این دسته توابع ارائه می‌کند. یعنی از توابع داخلی این مارپیچ
که معمولاً می‌باشد.

```
1 def func(x):  
2     """ does some math """  
3     return x + x * x  
4 print(func.__name__) >>> func  
5 print(func.__doc__) >>> does some math
```

```
def logged(func):
    def with_logged(*args, **kwargs):
        pass
        return func(*args, **kwargs)
    return with_logged
```

5 @logged

6 def f(x):
 """ does some math """
 return x + x * x

۹ print (f. -- name --) داده with - logged .

۱۰ print (f. -- doc --) داده none .

در زبان استفاده Decorator و می خواستیم نام تابع را چاپ کردیم .

۱۱ تابع new با جدید with - logged چاپ شد و نه تابع اصلی (f) .

استفاده از دکوراتورها همیشه به معنی از دست رفتن اطلاعات

صرفه بوده تابع اصلی است لکه به منظور جلوگیری از این آتفاق و حفظ اطلاعات مربوط به تابع اصلی خود می توسلیم از تابع

۱۲ wraps درون مارکر functools استفاده ننماییم .

۱۳ این تابع در واقع خود یک Decorator است که وظیفه

۱۴ این یعنی اطلاعات از تابع را به عنوان ارجومند دریافت کند و تابعی که برای آن انتساب دارد را اجرا کند .

۸. زنرآور (Generator):

۹. زنرآور (مولد) به توابعی لفته‌ی سود نه به میظور ایجاد یکتا
۱۰. تابع با رساندن مثابه اشیا iterator (۱ هر) پیاده‌سازی
۱۱. می‌گردند. هنلام مرا خوان یک تابع معمولی، بدنه تابع اجرایی سود
۱۲. تابع دستور return پس و خاتمه نماید و کجا با مرا خوان یک تابع
۱۳. generator، بدنه تابع اجرایی سود بدل، یعنی Generator
۱۴. برگزاره خواهد شد که بتوان با استفاده از متده next
۱۵. آن، مقادیر مورد انتظار خود را یکی بیش از دیگری در خواست دارد.
۱۶. عملکرد Generator به صورت lazy (ذخراو) می‌باشد و دارای همان
۱۷. را یکی از خوب نبینی کند بلکه آنرا با تها در همان زمان که در خواست
۱۸. می‌سود، تولید Generate می‌کند. بنابراین در هنلام برخورد با
۱۹. مجموعه داده‌های بزرگ، زنرآورها مدیریت حافظه، کار احمد
۲۰. ترکی دارند و صحیح‌تر ما مجبور نیستیم بیش از استفاده از یک زنرآور
منظر بگذیم تمام مقادیر آن تولید شوند!

۲۱. برای ایجاد یک تابع Generator تنها کافی است تا در یک
۲۲. تابع معمولی از یک یا چند دستور yield استفاده نماییم
۲۳. (النون) منسق پایتون در هنلام مرا خوان چنین یادی یک سئی
۲۴. generator برای برگزاره توانایی تولید یک زنرآور
۲۵. از مقادیر (یا سئی) برای استفاده در کاربردهای تکرار و پیشرفت از این راه

سینتکس دستور `yield` سبب `return` است ولی با کرید متفاوت است. این دستور در هر نقطه ای از تابع که باشد، اجرای برنامه را در آن نقطه متوقف می‌کند و مانند توانیم با استفاده از متدهای `next()` یا `next(a)` مقدار `yield` (حاصل) شوند اما در پایان نماییم.

با این توجه داشت اگر پایان نماییم تولید تابع `Generator` توسط استخراج `StopIteration` می‌شود. البته در زمان استفاده از رسورهای به مانند `for` این استثنای سلسله و خلصه پایان می‌پذیرد.

```

14 def a_generator_function():
15     for i in range(3):
16         yield i * i
17     return
18
19 my_generator = a_generator_function()
20
21 my_generator.next() => 0
22
23 my_generator.next() => 1
24
25 for i in a_generator_function():
26     print(i)

```

تعریف یک `generator` در مغایر `my_generator`

سلسله و خلصه

ویژگی‌های تابع Generator:

- تابع Generator شامل یک یا چند دستور yield می‌باشد.
- در زبان فرآخوند تابع Generator، تابع اجرا نی شود بلکه در عوض یک شی از نوع generator براحتی تابع برگردانده می‌شود.
- با استفاده از دستور yield می‌توانیم در هر نقطه‌ای از تابع گذشت و آن را در آینده با استفاده از متود next دوباره اجرا کنیم.
- با نمایش مراحل متود next تابع اجرای شود، تازه مانی.
- به یک دستور yield برسد. در این زمان دستور yield یک میله تولید می‌کند و اجرای تابع متوقف می‌شود. با ضرایف مجدد متود next اجرای تابع از ایام قبل از دستور yield شروع می‌شود.
- معمولاً نیازی به استفاده مستقیم از متود next نیست.
- توابع Generator از طریق دستورهایی به مانند for یا توابعی به مانند sum و ... که توانایی دریافت یک منبع را دارند، صور استفاده می‌شوند.
- در پایان تولید توابع Generator یک استثنای stopIteration ایجاد می‌شود.
- در نقطه‌ای که خود تراویث می‌کند می‌باشد درون برنامه از این اتفاق بگذرد.

- ۸ مارٹش نتیجہ لے اس فاصلہ ان سیوار return درہن لہاں بعد
تایبخ بامٹا پایاں یافتئا اجری تایبخ دلک نقطع ہی سیورد
تایبخ Generator از لینا اور مستینا نہیں!
۹ با مرا خواں متد close ہی تو نویں یک نئی
را خاموش کنیں۔ توجہ راستہ بائیسید لہ بس ان مرا خواں
۱۰ این متد جنابیں، باز ہم درخواست ایجاد مقدار ارسال شود،
۱۱ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۱۲ stopIteration یک استینا (next) لرائیس
۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۲۶ ۲۷ ۲۸ ۲۹ ۳۰
۲۷ ۲۸ ۲۹ ۳۰
۲۸ ۲۹ ۳۰
۲۹ ۳۰
۳۰

```

1 def countdown(n):
2     print("counting down from %d" % n)
3     while n > 0:
4         yield n
5         n = n - 1
6     return
7
8 countdown_generator = countdown(10)
9 countdown_generator.__next__()
10 >>> counting down from 10
11
12 10
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

```

با درخواست ایجاد مقدار
پس از آن استینا لرائیس و شیڈ

+ کوروتین (coroutine) :

Generator یا یون 2.5 ولت های جدیدی به تابع از سنه پاییون

اُنزووْدَه شَدَّ . الْأَدْخَلِيَّ تَابِعٌ ، دَسْتُورِيَّ يَلْدِيزِيَّ رَايِّدِرِسْتَ

لست بـ عـملـيـن اـسـابـ = مـلـدـمـهـ اـنـلـاهـ تـابـعـ مـذـكـورـ

رضا رضایی از خود نشان می‌دهد که به آن در زبان

برنامه نویسی سازون (coroutine) یک سود

پس زدن در آنها، می‌توانیم مقادیر لذت‌گاه خود را به تادیم

Generator اسما، لئے:

Generators will be coroutines

لَا تَأْتِيَنَا مُؤْمِنٍ بِإِيمَانِهِ إِلَّا مَنْ جَاءَنَا

رسال معمار به دلخواه نایب احمدیار اسب

```
print("Ready to receive")
```

while True :

n = (yield)

```
print ("Got %s" % n)
```

receiver - generator = receiver()

receiver-generator -- next -- ()

>>> Ready to receive

receiver-generator.send('woow!')

A curious course on Coroutines and

Concurrency



8 October 2020

اردویو خسرو (Jaxx)

۱۷ پنجشنبه ۸ October 2020 صفحه ۱۴۴ ارمنی (نیافروردی) ۱۳۹۹

concurrency با ملاحظات تابع coroutine بدن اجرا شود بلطفاً یک شی از نوع generator بدانندگی شود. متدهای next، اجرای برنامه را به تحسین yield می‌رساند. در این نقطه تابع در وضعیت تعطیق (Suspend) ملکیت یield و آماده دریافت مقادیر است. متدهای send مقدار مسنتظر را بدین تابع لرسال می‌کند. این مقادیر توسط عبارت yield دریافت می‌شوند. پس از دریافت این مقادیر، اجرای coroutine تاریخی خود را بازیابی می‌کند. این تابع تابع coroutine بدانندگی شود. متدهای close این تابع آن را توسط برنامه با ملاحظات می‌کند. یا خود را با پایان خصوصی اجرای تابع، پایان بیندازد.

- دریافت coroutine ها برای رهایی



9 October 2020
۱۴۴۲ صفر ۲۱

٣ صفر

@coroutine

می توانی از Decorator ها استفاده کرد :

```
def coroutine(func):
```

```
2. def start(*args, **kwargs):
```

```
3 generator = func(*args, **kwargs)  
4 generator.__next__()
```

`return generator`

6 return start

نکته: جایی که بlyn از پایان coroutine میگذرد، send() مراحلی شود یک استثنای stopIteration میگیرد.

درینه: coroutine ها، یک coroutine چنانچه

همزمان با دریافت مقدار، حروف بین تکمیل و برقراراند:

```

1 def line_splitter(delimiter = None):
2     print("Ready to split")
3     result = None
4     while True:
5         line = yield result
6         result = line.split(delimiter)
7
8     Splitter = line_splitter(',')
9     Splitter.next() => Ready to split
10    Splitter.send("A,B,C") => ['A','B','C']
11    Splitter.send("100,200,300") => ['100','200','300']

```

در صور سطر ۵ (line = yield result) چیزیم نه برای

انجام عمل انساب، این دام است میتوان عبارت سمت راست

انساب و سپس به سمت اجرا انساب را میشود.

یعنی مفسن پایتون ابتدا `yield result` را اجرا نمایی لند له عامل
 آن باز نراید، نتیجه استخیر `result` (ربار بختی اجرا خلف `None`)
 بر خارج تابع خواهد بود و سپس عبارت
^{۱۰} `line = yield`
 له مقدار ارسالی از متده `send` را به متغیر `line` انتسب
^{۱۱} کرده .

+ تابع بازگشتنی (Recursize Function)

با استفاده از `while` ، `for` آشنائیم ، این
 دستورات تنها اینکه ما برای تعداد قسمتی از آن بودند . اینون با
 پیاده سازی شیوه ای جدید در تعداد آشنا می شویم .
 به بیان ساده ، تابع بازگشتنی به تابعی لفته می شود که خود را از
 داخل یک خود فراخوانی می نماید . پیاده سازی تابع به صورت
 بازگشتنی شیوه ای است که از آن راه حل بخواهیم این رفت
 می شود و باید باید از توابع بازگشتنی یک سینتکس خاصی یا
 دستور خاصی در زبان پایتون نیست بلکه یک شیوه حل مسئله
 می باشد له با استفاده از تابع در زبان برنامه نویسی پایتون
 (همچون بسیاری از زبان های پلیر) تبلیغ پیاده سازی است .
^{۱۸} عموماً می توان مسئله هایی له از توالی انجام پلیر کار بیسان
^{۱۹} تابل حل هستند را به صورت بازگشتنی پیاده سازی کرد .

نکته: میراثیت اتوال تابع (رئیسه بازیستی) را جاگذب با استفاده از ساختهان راه پسته (stack) انجام می شود.

- هر تابع بازیستی شامل دو بخش عمده است:

۱. یک عبارت حاوی فراخوان خود تابع

۲. یک سرط بار اسما بین فراخوان بعد و پایان

نمونه کد: مقدار فاکتوریل عدد ۵ :

1 def factorial(n):

2 if n <= 1:

3 return 1

4 else:

5 return n * factorial(n - 1)

6 factorial(5) \rightarrow 120

مثالی له factorial(5) فراخوان می شود ($n = 5$). شرط

$n <= 1$ نداش مقدار close اجرای شود. در این مرحله نمونه

دیگر از تابع بازیستی $n = 4$ فراخوان می شود ($n = 4$)

و اجرای تابع factorial(4) نتصار پایان اجرای factorial(4)

و رسایفستا نیسبت آن می صافد. به همین ترتیب چنین نمونه

از لیک نایم اجرایی شوند که منتظر دریافت نیهان نمودند.

بعد از خود هستند. بر تابیت اشتبه $1 = < n$ بتواند

محی شود و نمود (1) factorial یعنی خود را ب (2) factorial

بر عیار آنند. به همین ترتیب بازگشت راهه محی شفود

تابه نمود نهست اجرای شده یعنی (5) factorial برسد و ترتیب

اجرای صور نظر کاربر به پایان برسد

$12. \text{ factorial}(5)$

$13. \text{ Return } 5 * \text{ factorial}(4)$

$14. \text{ Return } 4 * \text{ factorial}(3)$

$15. \text{ Return } 3 * \text{ factorial}(2)$

$16. \text{ Return } 2 * \text{ factorial}(1)$

$17. \text{ Return } 1 * \text{ factorial}(0)$

$18. \downarrow$

تجربه می‌باشد سلف شیوه بازگشتی شاید ب نظر یه جان انتیز باشد اما عالموش

تلرد که میزان حافظه (Memory) زیادی مصرف اجرای

آن زمان برای درست جواب اجرای اغلب سنت و اشغال زیادی

۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

(debug) آن ساده نتوهه بود!

استفاده از Decorator در تابع بازگشتی :

همان استفاده از decorator برای تابع بازگشتی باید به این نکته توجه داشت از آنکه decorator برای نتایج نمونه های مراحل شده از تابع اعمال خواهد شد و اینکه تنها یک نمونه از decorator یکبار می شود و تمام نمونه های تابع به همان یک نمونه ارسال می شود.

نمونه کد:

```

1 def logger(func):
2     print("Decorator is created!")
3
4     def func_wrapper(number):
5         print(f'new fact call with param: {number}')
6         result = func(number)
7         print(f'fact({number}) ==> {result}')
8         return result
9
10    @logger
11    def fact(n):
12        if n <= 1:
13            return 1
14        else:
15            return n * fact(n-1)
  
```

8. 15 fact (5)

9. >>> Decorator is created !

new fact call with param : 5

10. new fact call with param : 4

11. new fact call with param : 3

12. new fact call with param : 2

13. new fact call with param : 1

14. fact (1) => 1

15. fact (2) => 2

16. fact (3) => 6

17. fact (4) => 24

18. fact (5) => 120

+ تنظیم عمق بازگشتی :

در زبان برنامه نویسی پایتون در عمق پیاده سازی توابع بازگشتی
(تعداد نمونه های فراخوانی شده از تابع و موجود در پست)یک محدودیت مابین تنظیم وجود دارد . تابع ()
getrecursionlimit از مسائل کدیک این مقدار را برمی آوردند . این مقدار

۸ به صورت پس مرض برابر با ۱۰۰۰ می باشد لذا با استفاده از تابع `getrecursionlimit()` از مانند کد زیر می توان آن را تغییر داد.

۹ بارگذاری این محدودیت معمولی توابع بازگشتی باید استثنای `Recursion Error` خواهد داشت.

۱۱ ۱ Factorial (9) \Rightarrow 362880

۱۲ ۲ sys.getrecursionlimit(10)

۱۳ ۳ Factorial (9) \Rightarrow Recursion Error



۱۴ نکته: علاوه بر این محدودیت، یک محدودیت جی نریزی نیز وجود دارد و آن هم میان فضایی است که توسط سیستم عامل پردازش نظر گرفته شده است. بارگذاری از این مقدار فضای برنامه با خطای زمان اجرا مواجه می شود.

(Runtime Error)

۱۷ + استفاده از Generator در توابع بازگشتی:

۱۸ در میان سانچی توابع Generator و Coroutine نیز می توان شیوه بازگشتی را در نظر گرفت، درین صورت میان است نتایج کمی بخلاف انتظار شما باشد.

نون کد :

```

1 def flatten (lists):
2     for sub in lists:
3         if isinstance (sub , list):
4             for i in flatten (sub):
5                 else:
6                     print (sub) → yield sub
7 items = [[1, 2, 3], [4, 5, [5, 6]], [7, 8, 9]]
8 flatten (items)
  1 2 3 4 5 5 6 7 8 9
  
```

المن بلک تبديل تابع flatten به يك Generator
 است به جاي yield از print استفاده ننمی.
 آتفاق نیافردا !! وحیجی يك لیست حالی است . مرا خوانی
 تابع genflatten ترا باعث ایجاد يك سی Generator
 می شود و می باشد در نقطه ای له تابع خودش را مرا خوانی می کند .
 نیز مفهوم است براحت خوبی يك سی Generator با
 ناصم ننمی .

اصلاح کر جملہ:



```

1 def genflatten(lists):
2     for sub in lists:
3         if isinstance(sub, list):
4             for item in genflatten(sub):
5                 yield item
6         else:
7             yield sub
8 items = [[1, 2, 3], [4, 5, [5, 6]], [7, 8, 9]]
9 list(genflatten(items))
>>> [1, 2, 3, 4, 5, 5, 6, 7, 8, 9]

```

// + تلیف - Memoization

یا یادآوری، یعنی تلیف برش نمایندگی از

نیام تبیست آمده به منظور جلوگیری از تلاش محاسبات است.

ین تلیف را بیوان زبان برنامه نویسی پایمود با استفاده از

decorator بیاره سازی می کرد.

برای توضیح این بخش مثال حساب زدن باله فیبوناچی نیک عذرخواه

بررسی می نمیم:

1 def memoize - fibonachi (func) :

2 memory = { }

3 def func - wrapper (number) :

4 if number not in memory :

5 memory [number] = func (number)

6 return memory [number]

7 return func - wrapper

8 @memoize - fibonachi

9 def fibonachi (n) :

10 if n <= 1 :

11 return n

12 else :

13 return fibonachi (n - 1) + fibonachi (n - 2)

14 for number in range (10) :

15 print (fibonachi (number))

به لطف Decorator در این مثال نتیجه حاصل از فراخوانی هر

نونه تابع در جایی ذخیره می شود (شی) پیشنهاد (memory)

بیش از فراخوانی مجدد نموده تابع جدید از تابع پیشنهاد می شود

ایا جبلا این مقدار حساب شده است یا خیر. در صورت وجود جواب
 ان تبلان مرا خوان تابع صرف نظر و مقدار آن هست موجود به عنوان
 سیبز برگ رانده می‌سود. بنابراین بدین است که با جلوگیری
 از ایجاد نوون توابع جدید و حسابات تماریس، سرعастا اجل
 افزایش) می‌یابد.

Function Attributes

اُشیا در پایتون بر حسب نوع خود شامل یک سری صفات یا ویژگی‌ها
 پیش فرض حستند؛ برای مثال صفت name — که در برگردان نام تابع است.
 علاوه بر این، توابع در پایتون می‌توانند صفات دلخواه کاربر را نیز دریافت کنند به این صورت می‌توان
 یک سری اطلاعات اضافی را به توابع پیوست کرد.
 با استفاده از سینتیس زیر می‌توان یک Attribute به تابع اضافه کرد:
`function_name.attribute_name = attribute_value`
 همین برای این منظور می‌توان لز تابع آماده setattr استفاده
 کرد. این تابع سه ارگومان دریافت می‌کند؛ یکی ای که می‌خواهد
 یک Attribute به آن اضافه کنید (درینها تابع)
 ۳. ۲) راز نوع شسته - Attribute موردنظر مقدار ۳ (string)

`setattr(object, name, value)`

```

1 def foo():
2     pass
3     setattr(foo, "is_done", True)
4 if foo.is_done:
5     print('Done!')
6 >>> Done!

```

این صفات مقالب یک سی دیلسترنی ذخیره گی شوند لہ با استفاده از صفت dict در رسس هستند.

بلو دریافت مقدار یک Attribute شخصی توانید آن تابع getattr اماده نیز استفاده نمود. این تابع دو پارامتر اجباری name و یک پارامتر اختیاری default دارد.

در صورتی که مورد نظر (در اینجا تابع) ناتوجه صفت مورد نظر باشد مقدار default (در صورت ارسال) برقرار شده خواهد شد.

getattr(object, name, [default])

در صورت تلاش برای دریافت صفت که بلو تابع مورد نظر تعریف نشده باشد یک استثناء AttributeError نوش خواهد شد.

برای جلوگیری از بروز این استثنایی توان پیش از استفاده از حفظ،

وجود آن را با استفاده از تابع آماده hasattr(object, name)

بررسی کر.

با حذف یک Attribute نیز می توان از تابع آماده

delattr(object, name) استفاده کرد و یا از دستور del

توجه: در صورت استفاده از Attribute بلوی توابع خود واستفاده از

@functools.wraps استفاده از decorator ملحوظ

نشود. (۹ مر)

.....

.....

.....

.....

.....

.....

.....

۸ : Additional methods on Integer Types *

۹ یک جفت عدد صحیح $\leftarrow \text{int_as_integer_ration}()$. ۱

۱۰ نسبت آنرا می‌توان با عدد اصلی است و با مخرج مشتبه برابر کرد.

۱۱ new in version 3.8

۱۲ نسبت اعداد صحیح (اعداد کامل) همیشه عدد صحیح به عنوان صورت و ۱ به عنوان مخرج است.

۱۳ مخرج م به صورت $x = 0.5$

۱۴ $x.\text{as_integer_ration}() \ggg (1, 2)$

۱۵ تعداد بیت‌های را نمایش $\leftarrow \text{int_bit_count}()$. ۲

۱۶ باینری (دوبوی) قد مطلق عدد صحیح برمی‌گرداند.

۱۷ ۱ $n = 19$

۱۸ ۲ $\text{bin}(n) \ggg 0b10011$

۱۹ ۳ $n.\text{bit_count}() \ggg \underline{3}$

۲۰ ۴ $(-n).\text{bit_count}() \ggg \underline{3}$

۲۱ تعداد بیت‌های لام بلو $\leftarrow \text{int_bit_length}()$. ۳

۲۲ نمایش یک عدد صحیح را به صورت باینری (دوبوی) .

۲۳ باستثنی علامت و صفرهای ابتدایی.

۲۴ ۱ $n = -37$

۲۵ ۲ $\text{bin}(n) = 0b100101$

۲۶ ۳ $n.\text{bit_length}() \ggg \underline{6}$

8 int.to_bytes(length, byteorder, signed) : 4

9 آرایه ای از بایت‌ها را که یک عدد صحیح نشان می‌ردد برمی‌گرداند.

10 آرومان length ← عدد صحیح با استفاده از بایت‌ها

11 طول نمایش را در می‌سیند و به طور پیش‌فرض ۱ است.

12 اگر عدد صحیح با تعداد بایت‌ها را شده تا بلن نمایش نباشد،

13 خطای overflow Error نمایش می‌گیرد.

14 آرومان byteorder ← ترتیب بایت‌ها صورت

15 استفاده بلک نمایش عدد صحیح را تعیین می‌کند و به طور

16 پیش‌فرض نوع 'big' قرار می‌گیرد. اگر byteorder

17 'big' باشد، هم‌زین بایت در ابتدا از انتهای آرایه است و اگر

18 'little' باشد، هم‌زین بایت در انتهای آرایه است.

19 آرومان signed ← تعیین می‌کند آیا از ممکن دو بایت

20 نمایش عدد صحیح استفاده می‌گردد. اگر signed

21 باشد و یک عدد صحیح منفی را در می‌گیرد، خطای overflow Error

22 نمایش می‌گیرد. (نحوه پیش‌فرض False می‌باشد).

23 1 (1024).to_bytes(2, byteorder = 'big')

24 >>> b'\x04\x00'

25 2 (1024).to_bytes(5, byteorder = 'big')

26 >>> b'\x00\x00\x00\x04\x00'

27 3 (1024).to_bytes(5, byteorder = 'little', signed = True)

28 >>> b'\x00\xfc\xff\xff\xff'

`int.from_bytes(b'1x001x10', byteorder='signed')` ۱۵

عدد صدقہ نشان را رہ شدہ توسط آرائی بایت را رہ شدہ طریقے
برجی لر انداز

۱ `int.from_bytes(b'1x001x10', byteorder='big')`

`>>> 16`

۲ `int.from_bytes(b'1x001x10', byteorder='little')`

`>>> 4096`

۳ `int.from_bytes([255, 0, 0], byteorder='big')`

`>>> 16711680`

نکتہ : ارلومن bytes باید یا یک سی بایت مانند پاسن یا
یک بایت ا تو لیکہ نہ کہ تکرار شوند

: Additional Methods on Float *

اگر مقدار نوٹ میز نہ اور `float.is_integer()` ۱۶

مشتمل ہی مقدار int باشد True برجی لر انداز و در غیر این

حالت False برجی لر انداز

۱ `(-2.0).is_integer()` `>>> True`

۲ `(3.2).is_integer()` `>>> False`

شهادت ایت الله قاضی طباطبائی اولین شهید محرب به دست منافقان (۱۳۸۸ هش) /
 شهادت ایت الله قاضی طباطبائی اولین شهید محرب به دست منافقان (۱۳۸۸ هش)
 ۳۱ October 2020
 ۱۴ ربیع الاول ۱۴۴۲

نیاں) ان یا عدد صحیح شناور را ← float.hex() . ۲
 به عنوان یک رشته همان دسیمال برعکس رواند . برای امعاند صحیح
 شناور مساحتی ، این نیاں) همیشہ در شامل یک $0x$ پیش رو و یک
 م دیگر توان خواهد بود .

1 (4.5). hex()

>> $0x1.200000000000p+2$

2 class.method float.fromhex() . ۳ ← روش کلاس

بلک بر رواندن float نشان داده شده بوسطه یک رشته همان دسیمال ک . رشته ۵ محلن است را از فضای سفیر

پیش رو و اسراحت باشد .

نتتیه متد hex() یک متند نویس ، در حالی که () متد کلاس است .

الدوی رشته همان دسیمال :

[sign] ['0x'] integer [: fraction] ['p' exponent]

\downarrow \downarrow
 + or - رشته هایی از اینجا همان دسیمال توانا

float.fromhex('0x3.a7p10')

>> 3740.0 $(3 + 10.16 + 7.16 \times 2) \times 2.0 \times 10^10$

٨ خلاصه سانچ (List Comprehensions) *

بـ هـنـلـیـات لـفـتـه بـیـشـوـدـه درـطـی آـنـمـیـ توـانـیـتـه تـابـعـ رـاـبـطـه
 گـذـرـتـه اـعـضـاـی یـكـنـوـمـه نـوـعـشـیـ لـیـسـتـه (List) اـعـمـالـ وـسـیـه
 اـمـرـ مـالـسـبـاـ یـكـنـوـمـه نـوـعـشـیـ لـیـسـتـاـ جـرـیدـه دـرـیـافـتـه کـرـدـ

١١ numbers = [1 , 2 , 3 , 4 , 5] ✓

١٢ squares = []

١٣ for n in numbers :

١٤ squares.append (n * n)

١٥ print (squares) => [1 , 4 , 9 , 16 , 25]

١٦ numbers = [1 , 2 , 3 , 4 , 5]

١٧ squares = [n * n for n in numbers]

١٨ print (squares) => [1 , 4 , 9 , 16 , 25]

١٩ سـیـلـسـ لـیـ list comprehensions بـعـدـهـتـه زـیرـهـ

٢٠ [expression for item₁ in iterable₁ if condition₁
 ... for item_N in iterable_N if condition_N]

٢١ s = []

٢٢ for item₁ in iterable₁ :
 ٢٣ if condition₁ :

٢٤ for item_N in iterable_N :

٢٥ if condition_N : ~ append (expression)

نکته: چنانچه نیز اعمال list comprehensions در حرف زیر شامل بیش از یک عضو باشد، می‌باشد مقارین نتایج در داخل یک پرانتز مترابه شفوند (به صورت Tuple)

$[(x, y) \text{ for } x \text{ in } a \text{ for } y \text{ in } b \text{ if } x > 0]$

نکته ②: اگر list comprehensions حوزه خود را دارد، در نیز مقدار متغیر خارج از آن بعنوان تیپس باقی ماند.

- 1 $x = 'before'$
- 2 $a = [x \text{ for } x \text{ in } (1, 2, 3)]$
- 3 $\text{print}(x) \rightarrow \text{before}$

* Generator Expressions

عملدر یک Generator Expressions مشاهده باشد استاد این است اولی با خاصیتی که شیء و برای اینها آن کافی است بجای برای استفاده از list comprehensions

برای اینها اینجا براحتی استفاده نمی‌شود.

- 1 $a = [1, 2, 3, 4]$
- 2 $b = (10 * i \text{ for } i \text{ in } a)$
- 3 $b.__next__() \rightarrow 10$
- 4 $b.__next__() \rightarrow 20$

لک تعدادت - Generator Expressions و list comprehensions

لک تعدادت بسیار حجم است . خوب ج

لک list comprehensions . حقیقاً همان نسبت نباشد

عملیات در مقابل لک شی لبیست است . در حالی که خوب

لک Generator Expressions شی ای است که من را زد

چلوونه نتایج را محاسبه به صورت تولید نماید . در لک این است

موضوعات دفعہ صفحی در بالا بردن کاریکا (performance)

بنام و مصروف حافظه (Memory) خواهد داشت

لک

- نیلارش جزوی : صدری مابدی

- دوره متخصص پلیتودن سینزیلت

- مدرس دوره : رضا دولتی

16 sabzLearn.ir

17

18

19

20

* مفهوم اسلوبیت و مائل :

اسلوبیت یک بنام است دیا دستور العمل های نوشته شده با استفاده از زبان اسلوبیت است. اسلوبیت‌ها بنام‌های لوچلی است لبه وسیله آنای این بخش وظایف تبلواری یا بد وظیف طولانی را به طور خود کار اینما رار در واقع اسلوبیت به کدهای بنام‌نویسی لفت می‌شود که کامپیل نمی‌شوند و محمول تفسیری شوند. در پایتون، محمول یک مایل متن ساده حاوی اند پایتون ره قرار است مستقیماً توسط کابر اجرا شود، اسلوبیت نامیده می‌شود، یک اصلاح غیررسمی است و به معنی مایل بنام سطح بالا است. از اصطلاح دیگر، یک مایل متن ساده، ره حاوی اند پایتون است. ره براي ولر لورن واستفاده از یک مایل پایتون ریلر صراحی

شده است، مارول نامیده می‌شود. بنابراین تفاوت اصلی بین مائل و اسلوبیت در این است که مائل‌ها باید import شوند، در حالی که اسلوبیت‌ها باید طور مستقیم اجرایی شوند. اسلوبیت و مائل اصطلاحاتی هستند که مسلم به جای همچ استفاده شوند.

- اسکریپت (Script) : یک فایل پایتون است که قابل استفاده باشد. این برای محناسنای اجرای شود و حقیقت آن را اجرای نماید، باید کاری انجام دهد. این برای محناسنای اسکریپتها اعملای حاوی درهای هستند که خالج از محدوده هر لیس پایتابغی نوشته شده.

- مارکول (module) : یک فایل پایتون است که قابل استفاده باشد. اسکریپتها یا مارکول های دیگر وارد شود. اعملای اعضا مانند لیسها، توابع و متغیرهای را به برای استفاده در فایل های دیگری که آن را import می کنند تعریف می کند.

- پلیچ ها (package) : گروهی از مارکول های مرتبط است که باهم کاری کنند تا عملکردن خاصی را ایجاد کنند. این مارکول ها در یک پوشه قرار دارند و می توانند هر مارکول دیگری وارد شوند. این پوشه اعلی جاکی یک فایل init پایه است که به پایتون می کند این یک پلیچ است و احتمالاً حاوی مارکول های بیشتر است که در زیر پوشه ها قرار گرفته اند.

- کتابخانه (library) : یک اصطلاح است که به طور ازدیاد معنای بسته ای ازدیاد است. این های می توانند رده هایی ایجاد کنند که جداگانه راست، باشند و می توانند طیف لسترهای از عملکردهای ایجاد کنند. مانند کتابخانه استاندار پایتون که شامل صدها مارکول



بلکه اینها کارهای پلیچ است. (که همراه با نصب و بدون نیاز به دانلود)

سورس لد یک پروژه به زبان پایتون در قالب یک یا چند ماثل توسعه دی یابد که در سورس لد های با پیش از یک ماثل، بین اسسه ماثل هایی که از نظر منطقی با یکدیگر مرتبط هستند ۱۰. این رایترستورپ های بجز املاک دهم که به این نوع رایترستوری ۱۱. زبان پایتون پلیج (package) گفته می شود.

بر تعریف زبان پایتون دونوع ماثول وجود دارد:

۱. ماثول عادی (pure Module) که همان تعریف عادی از ماثول پایتون است. (نایل های با پسوند .py) ۱۲. که بزرگ زبان پایتون را نوشت شده) ۱۳.

۲. ماثول توسعه (Extension Module) ۱۴. ماثل هایی که توسط زبان های برنامه نویسی دیگری به غیر از پایتون تحریمه شده اند. پایتون یک زبان توسعه بزرگ است ۱۵. و در آن می توان از لدهات توسعه شده با دیگر زبان های برنامه نویسی استفاده نمود. به مانند C و C++ در پیاده سازی Iron python Jython یا C# در پیاده سازی Python ۱۶.

بنامه بوسی استفاده نمود. به مانند C و C++ در پیاده سازی Cpp

Iron python Jython یا C# در پیاده سازی Python ۱۷.

۱۸. (python 3.6 Extending and Embedding) برا مطالعه بیشتر لیاب

Guido van Rossum

نوشت

+ برنامه نویسی ماتریال ۸

لیکن اینکه توصیه شده در برنامه نویسی، توسعه برنامه در واحد هایی کوچکتر آن دارد است. به عبارت دیگر واحد هایی که هر واحد دارای نقش مستقل و مستقل از یکدیگر واحد های را دارد باشد. مستقل برابر با معنی آنست که تغییر یا جایگزینی یک واحد باید برای هر واحد دیگر دلیل واحد های تاسیکدار باشد. به این ترتیب مراحل خطاب این تکمیل و توسعه است. برنامه به عبارت بسیاری یا بسیار باید. لیکن از اسلامات پایان یافته در رسیدن به این اصل استفاده درست از معاهیم پلیم و مارول است.

در پاییون بلکه ایجاد دسترسی به یکدیگر ماتریال هایی باشیست ابتدا لازم های import کرو و این عمل بر هرجایی از بین ماتریال یا اسلوپیت جاگذاشتن است. لیکن باید توجه داشت که اجسام اندھایی پاییون اس طور بسط از بالا به پایین انجام بگیرند و در صورت نیاز به هر ماتریال می باشیست تا سیستم آن را import نماید.

بلکن import هر ماتریال می باشیست از آنکه تمام بسته های موجود از ابتدای تا ماتریال مورد نظر به ترتیب از آن لرید به صورتی که هر چیز باشیم کار است. باید بیکار متصفح سده باشند:

import actors.names actor.1
from actors.names import actor1

حالت اول



۱۰. انتشار پروژه :

در پیک پروژه، کامل بدی خواهیم آن را در کامپیوئی پایتون منتشر ساختم،
جدا از سورس‌کد لازم است تا موارد زیر را ساختار آن را نظر گرفت

۱۱. my-project

۱۲. pyproject.toml

۱۳. LICENSE.txt

۱۴. README.rst

۱۵. requirements.txt

۱۶. src/

۱۷. unique-package-name/

۱۸. init_.py

۱۹. main.py

۲۰. docs/

۲۱. test/

۲۲. ساختار ابتدایی تنها شامل سورس دارد ولی این ساختار تمام

۲۳. سورس‌کد در قالب یک پلیس پایتون بخشی آن جمهوری بزرگتر

۲۴. است. (در حال حاضر نیاز به روایت این ساختار نداشتم)

۲۵. تنهای این نتایل ها نیستند)

۱. \leftarrow pyproject.toml ← تمام اطلاعات صورتیان

بلاک خلاینه تولید (Build) یک Distribution یا یک سیستم قابل انتشار از پروژه صورت نظر توسط راین فایل تعریف شده است.

۲. \leftarrow README.rst ← تمام پروره‌های بایستا سمل سندی

بلاک توضیف خود باشد. در پایتون بلاک ایجاد اسناد معمولاً از زبان نشان داری reStructuredText استفاده می‌کرد.

و به همین دلیل این اسناد پسوند .rst مارک‌دات است، اجباری

۳. \leftarrow requirements.txt ← این فایل بایکا معرفی می‌کند و محتواهای

خاصی که در پروژه استفاده شده‌اند و زمان نصب آنها یا اجرای آنها را مشخص می‌کند و اعلیاً احتمالی یک لیست از متن بروانه‌های

وجود یا نصب بود آنها نیز ضروری است. استفاده می‌کردد.

۴. \leftarrow LICENCE.txt ← این فایل پروانه انتشار پروژه را

مشخص می‌کند و اعلیاً احتمالی یک لیست از متن بروانه‌های

حقوقی یا رایج به مانند MIT، GPL، BSD و پاشرد

۵. \leftarrow docs ← در این دایرکتوری اسناد (راهنما، امنیتی و...) پروره متن دارند می‌شوند. ایجاد این اسناد در کامپیوت پیش‌تر

معمولاً توسط sphinx انجام می‌شود.

۶ - test ← این رایتلر توکن ممل نتیجای لد های مربوط

بر تست پلقد من باشد . تست انویسی یک امر ضروری در اوند تو سحر هر پروژه بر نامه نویسی است . این رایتلر توکن می تواند هم در بالاترین ساخته از پروژه و هم در داخل رایتلر توکن سهی دو مرار دارد .

۷ - المن می توان از روی این پروژه یک توزیع (Distribution) ایجاد و آن را با استفاده از اینزرهایی مانند Twine یا بریک (Github) یا popl منتشر ساخت .

۸ - * نتائج در صور پلیج ها :

از نسخه ۳.۳ پایتون با افزوده سدن ویتل جدید پنهان شده است . namespace package یعنی بسته نصانم

پایتون به دو ساخته بسته عاری Regular package همان تعریف صیغی از پلیج می باشد . بسته فضای نام نستش یافته است : در واقعیت تا قبل از پایتون ۳.۳ هر پلیج پایستگی با پایسته عاری یک فایل init.py باشد . ولی دلیر نیازی به این فایل نیست . مفسر پایتون المن می تواند

در پایتون هر Regular package می‌باشد تا حاوی فایل `init.py` باشد. این فایل را می‌توان درون آن فایل و جزو ندارد. این فایل را برای سورس خود را به عنوان یک پلیج (ملحق برای مقاله‌های مرتبط) بر منسق پایتون معرفی می‌کند.

`my-model`

نحوه پلیج:

`init.py`

`training`

`submission`

`metrics`

`init.py`

`init.py`

`init.py`

`dataset.py`

`submit.py`

`precision.py`

`training-Loop.py`

`run-context.py`

`recall.py`

`loss.py`

`f1-score.py`

لکته all در پایتون لیست را در شرط هایه تعریف می‌کنند.

چند متغیرهایی باید به فایل رسیلری وارد شوند. متغیرهایی که

یدک (لیست مخصوص سده اند فقط پس از وارد کردن این فایل

در فایل رسیلر متابل استفاده هستند و بقیه متغیرها در صورت

حال آن بگوییم ان * باتی import لردن ماثل و پیچ ها استفاده نیم، باید درون نایل .py - init - مربوط به آن پیچ، متغیر های مربوط بر دهن all شخص نیم در غیر این صورت با خطا name Error مواجه می شویم.

* متغیر محیط path و مشناهای پایتون :

برای آنکه یافتن از نسخه موجود پایتون محابتوانید دستور python --version را در خط فرمان سیستم عامل وارد کنید؛

نایم؛ البته در ویندوز باید اینکه محله این دستور دایسته باشد؛ تغییر path می باشد، بسته است از مسیر

ادام نمایید Control panel > programs and Features

نکت - path یعنی از متغیر های محیط (Environment Variables)

سیستم عامل است. این متغیر حافظ خواست رایتلری هایی می باشد که سیستم عامل در آن ها به دنبال نایل اجرایی همیزی باشد. با دستور خارج شده در خط فرمان می تردد.

در هنگام نصب اسخ ای از قصد ندارید از آن به صورت نسخه

بیس (base) نصب یا توں خود استفاده نمایید، به این نکته توجه

روانسته نمایید که در صرحد سفارشی سازی (Customize

لرین، اندروید) خود کار میتوانید میتوانید path

ویندوز را نعال نمایید.

در این صورت با وارد کردن `python` در خط فرمان ویندوز،
 منسق پایتون را نخواهد مرا خواهد می شود.
 در صورت تنظیم بودن path می باشد تا از طریق پنجره
 خط فرمان و سرور `cd` به مسیر نصب پایتون وارد شوید.
 البته اماکن استوار `path` در هر زمان وجود را در مسیر
`control panel > system > Advanced system settings > Advanced environment variables`
 را طی کرده و سپس با لایک روی `System variables`
 پنجره جدید باز می کنند که در متدها
 آن `path` را بپیما و انتخاب نمایند. بر روی `Edit` در پایین
 صنان پنجره لایک کرده و مسیر `Scripts` را
 وارد کرده و سپس بر روی `OK` لایک نمایند.
 در نتیجه بروی یک دیندون از هر دو شاخه پایتون انتخاب
 یا حتی چند نسخه متفاوت از هر یک به مانند نسخه های
`2.7`, `3.4`, `3.3` راهنمایی
 خوب بدل ترا خواهی منسق پایتون هر یک از آن ها در خط فرمات
 ویندوز، اسفاره از راه اندازی پایتون (`python launcher`)
 می باشد. در این راهنمایی به دستاری متغیر `path` نیست
 و از سرور `cmd` به جای سرور `python` استفاده می شود؛ بنابراین
 صریحت با دستور `2 - py` منسق اخیرین نسخه موجود از شاخه ۲۲

و باستور ۳ رم نیز مفسر اخرين (نفعن) محو زدن از مشاهده ۳۰
پايتون ضرائح اسپود و برانک فلاحوان مفسريل نعمت هنام
م بايستاد نعمت هنامها بع صراحتاً نيل لورد (۳-۳-pg)

* سیستم مدیریت پایه (pip) :

pip اینلری است مسی برخط فرمان که ازان بلکن نص
خف ، بفرسانی و درکن مدیریت پسته های (یا تابعه
های سصر ثالث) پایتون استفاده می لورد . برنامه نویس
لیس ان یافتن ستد مونظر خود را pip یا وب سایتها
و سرویس های دیلری به مانند bitbucket.com.github.com و
عنواند به وسیله ستر pip برخط فرمان ، اتفاق به نص
آن در پایتون نماید .

آشایی یا ماذن همچون pip و استفاده از pip در توسعه پروژه

های پایتون اهمیتاً بالای دارد . بلکن مثال صریح نماییم در پروژه

خود می بايستاد آن تاریخ با فرمات جداول نیاش دهید . راه حل

ابتداً ، توسعه کدها یا مارکی برای تبدیل تاریخ صیادری

به جداول توسط خود را می باشد . راه حل دیلر اما جستجو برای

یافتن تابعه یا مارک های است که پس متوسط دیلران

توسعه یافته و در مشارک همانند pip میسر یافته است .

در این سوابط برگ رسترسی باین تابعه تابعه های شخص ثالث آشنا کافی /

است با استفاده از pip از های را به معه تابعه تابعه های را اند خود اضافه نماید .

۱. pip از زمان انتشار نسخه ۳.۴ به بسته نصبی پایتون افزوده شده.

۲. در بر همراه آن نصب کاری شود لذا در صورت نیاز ب pip براحتی

۳. نسخه های معمولی تن، می باشد که آن را به صورت جداگانه نصب

۴. می توانیم براحتی نصب pip. لازم است تا مایل .pg - get - pip -

۵. را دانلود نماییم. سپس آن را به وسیله دستور کمتر از ۱۰ ثانیه نزدیک

۶. دستور python -m pip install pip - get - pip - pg را در خط فرمان، با سطح کاربری

۷. محدود نماییم. می توانیم اقدامات امنیتی را در نصب pip نماییم

۸. bottle pip بسیار آسان است. به عنوان نمونه براحتی نصب

۹. می توانیم این را به وسیله دستور (web framework) براحتی نصب کنیم

۱۰. از دستور pip install bottle استفاده می کرد. با وارد کردن

۱۱. این دستور، Bottle در پایپلاین جسمی سود دیگر نداشته باشد

۱۲. ابتدا دانلود، دیگر نصب کنیم و براحتی دستور site-packages

۱۳. پایتون افزوده می شود:

۱۴. در ادامه برف دستورات زیر را با pip اورده شده است.

۱۵. ۱. نصب آخرین نسخه از بایک بسته:

۱۶. # pip install [package name]

۱۷. ۲. نصب بایک نسخه خاص از بایک بسته:

۱۸. # pip install [package name] == [version]

٣. خصیفیل بسته :

pip uninstall [package name]

٤. بروزرسانی بسته :

* pip install --upgrade [package name]

٥. بروزرسانی اخود pip

* python -m pip install --upgrade pip

٦. لوقتن معرفت تمام بسته های نصب شده و آنها را حذف کنید

* user > pip list

٧. لوقتن معرفت تمام بسته های اینجا باشد اینها را بروزرسانی کنید

* user > pip list --outdated

٨. مشاهده چیزی است بسته نصب شده :

* pip show [package name]

٩. نصب تمام بسته های آنچه سعی بیکمالی میکنی به مانند

مشخص requirements.txt شو :

root > pip install -r requirements.txt

۸ مارکل pip به صورت پیش نظری تایی بسته های دریافت را
 ۹ در مسیری تبار می بندد لیه دریل رایانه (تایی کاریل) تایل
 ۱۰ رسئرس باشید این روپ یعنی و مریسیتا بسته به صورت
 ۱۱ حرف ایک پیشنیاد نبی سود جمله سبلند مدت و در هنلام توسعه
 ۱۲ پر نامه های لوناون، پر زامه نویس را رجای مسلسل خواهد لد
 ۱۳ کے برترین راه حل یا سیوه میزست بروئه در یا سیون، ایجاد محیط مجازی
 ۱۴ در این حالت پر ای هر پروژه یک محیط یا سیون کامل ایزو یا
 ۱۵ مسئل ایجاد می لورد، بنابراین مارکل pip هر بسته صورت نیاز
 ۱۶ در هر بروئه را آنرا در همان پروژه صادر می رله،
 ۱۷ کے سیوه دیلو استفاده از user - در میان دستور مارکل pip
 ۱۸ می باشد، این توسعه ساده برای این رهیز از نیاز به دسترسی root
 ۱۹ یا Administrator می باشد، در این سرایط مارکل pip هر
 ۲۰ بسته صورت نیاز را در محیط کاربری، کاربر حقوق تأثیراتی می نماید
 ۲۱ # user > pip install --user [package name]

* ایجاد محیط مجازی *

۲۲ حالی را در نظر بگیرید که بعد ایجاد بروئه های مختلف ابه نسخه های
 ۲۳ متفاوت از بخش تابعه های نیاز دارید؛ در این صورت جلوه هی تواید
 ۲۴ چندین نسخه متفاوت از یک تابعه را در یا سیون نصب نمایید؟

برای نمونه، فرض نماید که بخواهیم برای این توسعه حکم داد و دلیل ساخت آن
 بیل نسخه جدید (1.8) و ب فریم‌ورک جنلو و نیلز برای این نسخه مخصوص
 نسخه جدید (0.96) ازان کار نمی‌کرد، ولی نتوانیم!
 زیرا لذتی شود هر دوی این نسخه‌ها را با هم در یا پستون داشتیم!

راهنمایی برای این توسعه حکم داد و دلیل ساخت آن نسبتاً مخصوص است.
 معمولاً نسخه توسعه و اجرای هر بروزه یا پستون را به همان شیوه واپسی کرده
 از آن دان بروزه‌ها را یک‌باره جبرایلی اینکه می‌دانم.

- این از اینکه این توسعه حکم داد و دلیل ساخت آن نسبتاً مخصوص است.
 استفاده می‌شود که منظور از EN2 در آن، نشانی دایلیستوری
 نخواهد بود بلکه قصد داریم عیط صیانت ازان ایجاد کرد.

* user > virtualenv / Documents / SampleEn2 /

رسانی می‌کنیم که این توسعه حکم داد و دلیل ساخت آن نسبتاً مخصوص است.
 این توسعه حکم داد و دلیل ساخت آن نسبتاً مخصوص است.

می‌شود و جایی که بخواهیم عیط صیانت خور را برای نسخه
 مخصوص دلیلی از یا پستون ایجاد نماییم، لازم است با استفاده از
 نشانی مفسر آن مخصوص از آن دلیلی ایجاد کرد.

* user > virtualenv -- python = python 2 EN2

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

virtualenv -- python = C:\python25\python.exe # ویندوز

السون می توانیم در پروژه خود به تابعهای ها، pip، دایلسوئی site-package و مفسر اختصاصی دسترسی داشته باشیم، البته بیش از شروع کار با یک سیط بخاری می باشیم که آن را activate (فعال) و پس از آن که نیز آن را deactivate (غیرفعال) نمایم.
حال لرن را نیبا به معنای تنظیم سیستم path سیستم عامل بر روی مفسر سیط بخاری مورد نظر است از آن با غیرفعال لرن این وضاحتی از بنده می بود.

cd Documents\sampleEnv\Scripts

> activate.bat فعال لرن

> deactivate.bat غیرفعال لرن

از نسخه ۳.۳ پایتون به بعد مارکی با نام venv برای ایجاد سیط بخاری به تابعه اس آندر پایتون افزوده شده است از آن به جای نصب virtualenv استفاده نمود؛ برای این منظور از دستور python -m venv مسابه Env ایجاد می شود.

c:\python34\python -m venv Documents\...

* نایل ها (Files) :

نایل ها جزو منابع منطقی کامپیوتر به شماری بودند . توسط سیستم عامل مدیریت ای شوند و املاک برای نیازمند طولانی مدت از اطلاعاتی بی باشند . نایل ها در واقع بسته هایی متواتی از صفر و یک هستند که بر روی دیسک ذخیره کشیده اند و محموله ای رو قالب " جداگانه زیر ساخته ای شوند :

نایل باینری (binary File) ← نایل های ویدیو ، تصویر ، نایل zip ، نایل بایت الکترونیک و ...

نایل متنی (text File) ← نایل HTML ، نایل مانیل پایتون

نایل ها چیزی جز مجموعه ای از مقادیر های باینری (یادگاری)

نیستند و نایل های متنی نیز در واقع یک زیرمجموعه از نایل های باینری

است با این تفاوت در بسته های یک نایل متنی در کنار هم

بسته به نوع لغدادی این مساحت را برابر باشد ای ان کاراکتر هایی اسما مخصوص

و مقابل چاپ هستند . محتوا (این نوع خایل در سطر هایی جداگانه

قلدرداره ای شوند و با استفاده از برنامه های ورایسلر صنعتی برای

انسان خوانایی بشوند . برای نایل متنی باید لاین ASCII (اسلی)

هر هشت بیت را (یک بایت) برابریک کاراکتر مجاور باشد .

قرفته

برنامه : (لدهای اسلی هفت بایتی) هستند ولی در این کامپیوترها برای هر دو آن یک بایت

برنامه : (لدهای اسلی هفت بایتی) هستند ولی در این کامپیوترها برای هر دو آن یک بایت

برای نمونه در `python` برشل شس بایت که هر کدام به ترتیب
برای مقدار باینیک که اسلی کاراکترهاست باشد،
ذخیره می شود.

- اسلی تنها از ۱۲۸ کاراکتر (از کد ۰ تا ۱۲۷ در پایه ده) پشتیبانی می کند و به همین دلیل امروزه بسیار از کوادرات های استاندار `Unicode`
(یونیکد) استفاده می کرد. در یونیکد مجموعه بسیار بزرگتری از
کاراکتر تنها بود که اسلی پشتیبانی می شود به لوندگی این اسلی را نمی
دانند و بزمی بینند. برای نمونه در کوادرات `UTF-8` این اسلی استاندارد،
استاندارد `128` دنخستی برای درجهان `128` که اسلی می باشد.
کوادرات `UTF-8` کاراکترهای یونیکد را در یک شبکه پشتیبانی می کند
با طول مستحسن (از ۱ یکت تا چهار بایت) اینکه می رهد که تر
ین کوادرات همچنین برای کاراکترهای اسلی تنها یک بایت را نظر
لرفته می شود.

- دریک فایل باینیک وجود سطح مفروض نماید و بایت های آن همان
استاندارهای را نمایش رهند. نمی توان آن ها را توسط هیچ
یک از کوادرات های کاراکتر `ASCII`، `UTF-8` و ... خواهید
چاپ کرد یا حتی صملن است در صورت چاپ آن ها نیز حاصل
"برای انسان غیرقابل فهم باشد."

تابع open :

پایتون یک نوع شی اماده برای دستیابی فایل ها در برنامه هاست
که در آن این شی با استفاده از تابع اماده (open(file, mode))
یک اندیشه درود به آردوینو نیز یک شی است که حاوی نام کامل
فایل (نشانی به نام + پسوند)، موزنخانه بر روی رسالت کامپیوت
است و آردوینو روم نیز بیت رشت، با مقادیر خاص است و
حالی را که این فایل می باشد باز شود را مشخص می سازد. این

آردوینو (اخیراً) است و مقادیر پیش فرض آن 'r' می باشد
('r' و 'w' در open('C:\Users\saeid\Sample.txt'))

نکته: آردوینو file را می توان نوشتابا نام خالی فایل و بعنوان ذکر مسیغ آن
مقطر رهی لرز؛ در این صورت مفسر پایتون خسیر را برگزیری جایز

را برای آن در نظر می نماید. در اینجا مسیر را برای آن می نمایم

نکته ② با استفاده از تابع (os.getcwd() لزمانیم

مسیر را برگزیری جایز برآورد کرد. اینجا آوریدم

```
1 import os
2 os.getcwd()
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

- آرکومان mode نیز بر حسب نیاز بی تواند یعنی ان مقدار های پایین

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

٢٠- نایل (تالیت متنی ترا به منظور خواهند) از آن

باز می سسوزد و اشنازه در دن ابتد استخوان می لیرد چنانچه، نایلی با

ایسا نام موجود نباشد یا خطأ (پارسیش) File Not Found Error.

نراش بی مردگان متعادل 'rt'

۱۰۷) میخانه‌ای را می‌مالد اینسته این سایه‌مند خواهد بودشست و بولند

سازمان اسناد و کتابخانه ملی ایران

سار ہو تو بہرہ ان منانی سیر . چاچھے مایلیں باہیں

مودود بہاسند، ابتدائی بلا و پسیس بار می اسنود۔ معاشر

۱۵- نهایی نظر قالب متن: سهایه منظرون اصرار (متن)

در اینجا متن موجود در آن بایدی سروده، اشاره‌لر در اینجا

خایل می‌دارد. چنانچه می‌لی باین نام موجود نباشد، لبته ایجاد

و سپس باز می شود . محال 'at'

(¹⁷exclusive creation) $3x$ فقط ربانيون $\leftarrow x'$.

نیاں در مالب متین ایجاد و به منظور نوشتن در آن باز می شود.^{۱۸}

چنانچه، این نایاب از پیش موجود نباشد یک خط (یا استثنا)

١٩ 'xt' لیزنس می لرد .. محداں File Not Found Error

۸. $'rb'$ و $'wb'$ و $'ab'$ و $'xb'$

۹. همان توصیع لزینهای مسابه قبل را دارند ولی با این تفاوت که برای کار با فایلنهای در قالب (با اینک) استفاده می‌شوند.

۱۰. $'r+'$ فایل در قالب متنی به منظور خواندن و نوشتن

۱۱. باز می‌شود و اشاره‌لن در استانی آن تبار می‌گیرد. چنانچه فایل با این

۱۲. نام موجود نباشد یک خطای (یا استانی) `File not Found Error`

۱۳. لرash می‌گردد. توجه (لائسنس باسیمه نه زاین) حالت اعمال نوشتن از

۱۴. همان استانی فایل، باید جایگزین سند متنی یجدید باشند حافظ

۱۵. می‌شود. معادل $'rt+'$ یا $'r+t'$

۱۶. $'w+'$ فایل نویسی در قالب متنی به منظور خواندن و نوشتن

۱۷. باز می‌شود. هنن درون آن در صورت وجود پالس می‌شود و اشاره‌لن در

۱۸. استانی آن تصریح می‌گردد. چنانچه فایلی با این نام موجود نباشد

۱۹. ابتدا ایجاد و سپس باز می‌شود. معادل $'wt+'$ یا $'w+t'$

۲۰. $'a+'$ فایل در قالب متنی به منظور افزودن متن در

۲۱. انتها متن (موجود در آن) و همین خواندن باز می‌شود؛ اشاره‌لن

۲۲. در انتها فایل قرار گارد. چنانچه فایلی با این نام موجود نباشد،

۲۳. ابتدا ایجاد و سپس باز می‌شود. معادل $'at+'$ یا $'a+t'$

۲۴. $'x+'$ فقط در پایتون x . فایل در قالب متنی ایجاد و

۲۵. به منظور نوشتن و خواندن باز می‌شود؛ چنانچه این فایل از پیس موجود

$'xt+'$ $1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29$

۲۶. باش یک خطای (یا استانی) `File not Found Error` لرash می‌گردد معادل

$'xt'$

۸. $w+b$ یا $wb+$ و $r+b$ یا $rb+$...

۹. $x+b$ یا $xb+$ و $a+b$ یا $ab+$ ←

۱۰. همان توصیح لزینه‌های مسابقه بدل را درین دلیل با این تفاوت

۱۱. نه بلی کار با مسائل های در غالب باینی استفاده می‌شوند.

- پس :-

۱۲. ۱۱. r خواندن مایل (read)

۱۳. w نوشتن مایل (write)

۱۴. a ایجاد در زیر (appending)

۱۵. t تعیین مایل متن (text) برای مایل صورت نظر (پس معرفی)

۱۶. b تعیین مایل باینی (binary) برای مایل صورت نظر

۱۷. $+$ مایل به منظور بروزرسانی (updating) باز می‌شود

۱۸. ۱۸. u حالت Universal Newline (منطقی شده)

۱۹. در صار راد سیستم‌های اعمال نونالون از رشته‌های مختلف (نهی اتواند

۲۰. از نیپ یا جنر کارکتر تشکیل شده باشد) برای نشانه دلای انتها سطر

۲۱. های یک مایل استفاده شده است و برای نمود در ویندوز آن CRLF

۲۲. (یا '۱۳۱۰') و در لینوکس از LF (یا '۱۰') استفاده می‌شود.

۲۳. سی مایل بایسون به صورت پیش نرف از همان صار راد سیستم عمل

۲۴. میزبان ابری تعیین رشته newline خود استفاده می‌کند. ونی

۲۵. چنانچه، تصدیق در بایسون غایلی با تراک راد سیستم عامل دلیری

را به منظور خواندن باز نمایید. می بایست این کار بر حالت

خواندن از پلیس فایل پایتون تمام رشته های newline موجوب

برنایل را به کار است 'n' نداشتی لند.

نکته ۱: درستون os.linesep از مثقال دو رشته newline

repr(os.linesep) سیستم عامل میزبان را برمی برواند.

نکته ۲: باز لوین یک فایل به منظور خواندن ازان حلم یک منبع ورودی (input) و باز لوین یک فایل به منظور نوشتن ازان حلم یک منبع خروجی (output) را در بزرگساز دارد.

* متدهای شی فایل پاسخ:

۰۱. write(string) یک شی از نوع str یا bytes را دیافت می لند و آن را در عنوان شی فایل مورد نظر می نویسد.

نکته: متدهای write تعداد کاراکتر های مالکیتی نویسید را برمی برواند.

```

1 text = ' 1) python\n2) programming\n'
2 print(text)
>>> 1) python
2) programming
3 output = open('textfile.txt', 'w')
4 output.write(text)
>>> 25

```

نکتہ ②: نبی توان از چیزی str بٹی نوئیں دیگر مایلی لے رہا ہے جس کا باسی (b) باشہ نہ اسست است اسٹفادہ لئے۔

```

1 data = ' 1) python\n2) programming\n'
2 bdata = bytes(data, 'utf-8') # str → bgt
3 print(bdata)
>>> b' 1) python\n2) programming\n'
4 output = open('binaryfile.bin', 'wb')
5 output.write(bdata)
>>> 25

```

در زمانه لینوکس تفاوت بین حالت متنی و باینری وجود ندارد :

النون الـ هر دو فایل binary file . bin و text file . txt

را در لینوکس توسط برنامه ویرایش متن بازنمایید، خواهید دید که محتوا این دو فایل مشابه یکدیگر نباشد اما می شود.

python ۱) >>>

۲) programming

وکی در لینوکس به هنام حالت متنی، الـ یک فایل را به منظور خواندن بازنمایید که انتها \n موجود در آن به شکل '۱۲۱۷'

برگردانده می شود و الـ دیگر یک فایل را به منظور نوشتن بازنمایید، هر بار رست، '۱۲۱۸' برجای '۱۲۱۷' نوشته می شود.

(در حالت باینری این کارها انجام نمی شود.)

باز لینوکس فایل text file . txt :

۱) programming

: binary file . bin

>>> ۱) python ۲) programming

۱۸

۱۹

۲۰

۱۰. ۱) عده‌ها ← پس از پایان کار با هر مایل راهی باشیست آن

را بینید، این متدهای باز شده را بینید، شیخ مایل مورد نظر

پس از فراغت این متدهای همچو شده را بینید، توانید فراغت

لینید، زمانی که سراسر ارجاع به یک شیخ مایل، به صفر برسد

یا وقتی متغیر مایل به شیخ مایل رسیده است، پایتون

۱۱. شیخ قبلی را به صورت خودکار بیند، همچو همیشہ برقرار است

۱۲. لر بستن تایل به صورت صریح توسط بزرگ نویسن انجام شود

همچنین برای بررسی این لیست یک مایل بیاز است، یا اینکه

۱۳. بسته شده است، یعنی توانید از صفت closed استفاده

نمایید؛ این صفت در صورت این مایل بسته شده باشد

حاوی مقدار True بی‌باشد.

۱۴. ۲) read ← تمام جواب مایل را خواند و در قالب یک

۱۵. شیخ انواع str (برای مایل‌های سفی) یا bytes (برای

۱۶. مایل‌های باینری) برمی‌گرداند.

۱۷. این متدهای اریومان اختیاری نیز دارند، این اریومان یک عدد

۱۸. صحیح است، لر تعباد کار است (یا بایستی)، لر بی‌باشیتا از مایل

۱۹. خوانده و برگردانده شوند لر تعیین نمایند.

۲۰. نت: هنایی از مایل بر حالت rt، باز شده است، اشاره کن در اینجا

۲۱. مایل تعداد لفته و با هر میزان خواندن از مایل، موقوع است اشاره لر

۲۲. نیز به خلو حالت راست است.

۸ در هر من اخوان یک سطر از `readline()` . ۹

۱۰ فایل را تاشت `newline` (برچ لردن) . ۱۱

۱۲ `readlines()` ← تمام سطرهای یک فایل را در قالب ۱۳ یک شی لیست برج لردن . ۱۴

۱۵ `writelines(List)` ← یک شی لیست را تمام ۱۶

۱۷ اعضای آن از نوع `str` هستند و به عنوان اردوغان لفست . ۱۸

۱۹ و اعضای آن را به ترتیب (در فایل) موردنظر کیانویسی ۲۰

۲۱ `L = ['a', 'b', 'c', 'd\n', 'e']`

۲۲ `f = open('textfile.txt', 'w')`

۲۳ `f.writelines(L)`

۲۴ `f.close`

۲۵ `>>> abcd`

۲۶ `e`

۲۷ `seek(offset)` ← اردوغان یک سرا

۲۸ `offset` و این مقدار ممیخته اسالر فایل را برآن

۲۹ نسبت به ابتدای فایل تغییر می رهن . ۳۰

۳۱ `f = open('textfile.txt')`

۳۲ `f.seek(3)`

۳۳ `1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30`

۳۴ `f.read(6)` `>>> Python`

۸. PLUS^{TM} بسیار باعث اخیره محتواست. با پیش در فایل می شود.
 هنگامی که فایل را به منظور نوشتن، باند buffer تا پیش از زمان
 بستن فایل، هر آنچه در آن محتوی می باشد در پیش از این زمان داره می شود.
 مرا خوان این متد بهم می کند تا بدون بستن فایل، اطلاعات
 از پیش به فایل منتقل نگردند. با پیش را خلو، حالی شود.

۹. Buffering ، open از ابزارهای اختیاری تابع Buffering ، open می باشد. هنگام تعریف اندازه پاس با استفاده از پارامتر پاس، می توانیم
 از مقادیر مختلف برای تنظیم حالت Buffering استفاده نمود.
 $\text{buffering} = 0$ باید خاصیت ترین پاس (نمک

۱۰. در حالت پاس (معنی است)
 $\text{buffering} = 1$ هر خط به عنوان یک پاس جزو
 در نظر گرفته می شود، زیرا راه ها به اندازه یک خط در فایل خوانده
 می شوند. (فقط در حالت متن مناسب و مقابل استفاده)
 $\text{buffering} > 1$ نایل در قطعات برابر با اندازه پاس
 را به سده خوانده می شوند. این حالت باید صریح است
 فایل های جیم یا پسند سازی از استفاده از حافظه مناسب است.

$\text{buffering} = -1$ default (default) سیستم عامل
 و لایه ای از I/O underlying I/O از صورت خودکار
 استفاده می کند. باید مشاهده مقادیر آن با استفاده از



10 DEFAULT - BUFFER-SIZE

نکته: از تابع `open()` نیز می‌توان برای حلقه `for` استفاده کرد، در این صورت در هر بار تکرار سطرها را فایل پیمایش کنیم:

۱. `for line in open('textfile.txt':`

۲. `print(line, end='')`

`>>> 1) python`

`2) programming`

از آنچه در متون مذکور خود جزوی `newline` (در اینجا `\n`) است، ابرویات `end` تابع `print()` را برای جلوگیری از درج `\n` در آنها هر سطر تغییر داردیم

۱۳. `print()` با `end = '\n'` از درج `\n` در آنها هر سطر تغییر داردیم

۱۴. دستور `*with / as`:
یک دستور مربوط است که از اسکیمی له با نوع `context Manager` در پایتون ساخته می‌شوند، پشتیبانی می‌کنند. برخی از این اشیاء در پایتون

لر مانند شی‌فایل (`Context Manager`) تابلیتی (با `__enter__` و `__exit__`) باشند. برای این اتفاق این قابلیت دارند. شی‌های موردنظری بایستی جزوی دو متد ویره (`__enter__` و `__exit__`) باشند. لب ب دریبا در زمان‌های "ورود به" و "خروج از" برند دستور `with / as` فراخوان می‌شوند.

دستور `with` ساختار مشابه پایین دارد:

`with expression as variable :`
`statement (s)`

در این ساختار، `expression` نام بخشی از دستور است که بدل
شی لزومی (`Context Manager`) را برای درون آن؛ یعنی شی
با استفاده از `__enter__` و `__exit__` به یک متغیر (رچاع) داده می‌شود.

1. `with open('textfile.txt', 'w') as output:`
2. `output.write('text')`

پس از ایجاد شی فایل، این شی می‌بایست با دستور `with / as` دستور
باشد که این کار را با منابع خواهد مستر.

(۱) `__enter__` انجام می‌دهد. در واقع این متدها نمایل
ایجاد شده را برای درون آن درآورده و توسط `as` به متغیر
ارجاع داده می‌شود. همین‌طور با استفاده از این ساختار بدلیل
نیازی به منابع متر `(.) close` بدل شی فایل نیست.

(۲) `__exit__` توسط `as / with` به انجام می‌رسد؛ در واقع `as / with`
بسیز فایل را بر پایان اجرای دستورهای بدل خود تضمین می‌نماید.

همین‌طور این ساختار همانند کار با فایل خطی (را استنادی) رخ
ردد، بیش از زارش آن، ابتدا فایل بسته خواهد شد.

نکته ۱: نوجہ داشت، باشد یک شی مایل همیشہ باید بسیه شود.

در مواقعی له مقصود استفاده از حلقه for بلکه شی مایل باریه.

بینه لست آن (بروگ) ساختار with/as بجهه بلیورید.

نکته ۲: ان دستور as می توان به شال توں تو بجهه لفظت.

نکته ۳: از نسخه ۳.۱ به بعد پایتون سیتماس جینیک افزوده شده.

توسط آن می توان همزمان از چند context manager را بجهه لفظت.

with A() as a and B() as b :

statements

5 1 with open('data') as fin, open('res', 'w') as fout:

6 2 for line in fin:

7 3 if 'key' in line:

8 4 fout.write(line)

نمونه کد بالا معرفایل را باز می کند؛ از یکی می خواهد و بقیه

از سطر های آن را در فایل دیگر می نویسد.

نکته ۲۰: از اینجا پس از اینکه کد اجرا شود، فایل res.txt ایجاد شده باشد.

* شی نایل استاندارد :

سنه نوچ شی نایل تو سطح مفسر پاپتون دایجاده می لرده هرسه

آن ها تو سطح مائل کارک در رشیس هستند.

لے `sys.stdin` بروی استاندارد؛ برای

بستیابی تایپ ورودی ها ز حالت تعاملی پاپتون (مانند

فراخواخ) تابع `(input()`) از این شی استفاده می لرده.

لے `stdcout` کو کارک خروجی استاندارد تو سطح

صورت استفاده قرار می کنید لیکن `print`

ملے `sys.stderr` بروی خطاهای استاندارد؛ سی ای

لے خطاهای دریافت سایی لند.

نه همیشه ولی ماتولن صبع `stdin` را صفحه، ملیک کامپیوئر

و مینیپ `stdout` کو `stderr` را صفحه نمایش

در نظر گرفت.

17

18

19

20

* تابع `input()` :

از این تابع در پایتون برای لفظی خودی از کاربر در حالت تعاملی که استفاده می‌کرد. تابع `input()` یک سطر از کاربر می‌خواند و در نتیجه یک شی از نوع `str` برمی‌گرداند.

با اجرای رستوری مشابه `s = input()` مفسر پایتون منتظر ورود متنی مانند سپس با فشردن کلید `Enter` (پیان) خود را ذهنی و سپس مفسر پایتون تمام کاراکترهای در ریاضی را در قالب یک شی رشته (نوع `str`) به صفتی موردنظر (که ارجاع می‌کند). همچنین می‌توان متنی برای مشاهده کاربر به صورت آن دوستان در فرایون تابع قرار دار.

نکته: تابع `eval()` تابع آماده دیلری که در پایتون استاند یک شی رشته (نوع `str`) را ریاضی تراویث آن را به شکل کد پایتون تفسیر می‌کند.

1. $x = 1$

2. $y = eval("x + 2")$

3. $print(y)$ عدد ۳

- لفون می‌توان بایک خط دیک ماسین حساب ساده ایجاد کرد:

print(`eval(input('Type your formula'))`)

* تابع `print()`:

این تابع `التوییب` به شلک پایین را دارد:

```
print(*objects, sep = '\n', end = '\n',
      file = sys.stdout, flush = False)
```

* `objects` ← بیانلر اشیائی است که خواهیم در خروجی قرار بدهیم.

هر چند همچ باشد (نه برابر بازیش یک سطر

خالی) یا اینکه می تواند یک یا چند شی (نه درین صورت اشیاء می باشد) بوسطه یک کاما (از یکیلر جدا شوند) را در بردارد.

* `sep` ← رشتنهای نه بایست بین اشیاء قرار گیرند.

نیز می تند را جدا شوند اشیاء از یکدیگر). مقدار این اردومن

در حالت پیش فرض یک فاصله (space) است.

مقدار ارسالی به این اردومن می باشد ایکس (شتر) یا `None` باشد.

```
1 print('python', '3x', sep = '-')
```

```
>>> python - 3x
```

```
2 print('python', '3x', sep = None)
```

```
>>> python 3x
```

* `end` ← رشتہ ای نه بایست اند اسکی هر سطر متولیه را تجییش می تند. مقدار این اردومن در حالت پیش فرض `'\n'`

مقدار ارسالی به این اردومن می باشد یک تئی رشتہ یا `None` باشد.

→ خوبی را تحسین کی اند می بایس شایک ہی بہ file
ھماراہ مت (write (String سفن) کلاس خود، باشد۔ این
ارکومات بہ صورت پیش خرض ہر لمحہ خوبی را استاندار مقدار دھی شدہ
استا، این تابع قابلیت استفادہ نہ حالانکے بایس کے فلیل ہمارا نہیں۔

```
1 output = open ('textfile.txt', 'w')
2 print('Hi :)', file = output)
3 output.close()
```

۱۳ این ارتوان از نسخه ۳.۳ بـ تابع print پیغام را از دستورات خارج می کند.
۱۴ باشد (رسورد قبل) با تنظیم این لزیت، بنحوی مقدار True یا False می باشد.
۱۵ اسماں متن را به تغایر بیرون تلفظ و قسم افعال را پذیرد.
۱۶ اهمان افعال حتمی را با عنوان معرفه خالی () نشاند.

* ساختارهای ذخیره و تبادل داره ؟
زمانهای بسیاری پیش خواهد آمد که برنامه نویس نیاز دارد داره ای
را بر روی فایل بنویسید یا داره هایی مانند پارامترهای پیکربندی نیز
از فایل ها در طول اجرای برنامه بخواند . شما در برنامه خود از هر
ساختاری که توانید برای ذخیره اطلاعات استفاده نماید ولی در علم ریاضی

بر مردم زمان ساختار های استانداری برای این منظور تعریف شده است.
 علاوه بر سرویست در کار با فایل ها و اسلان استفاده از پیشیاری از این
 های اپیشیاری آنرا برای این دست افزایش ها، پیروی
 از این ساختارها در همان سازمان برنامه های با میتوانند های
 بیشتر باشند. چراکه این ساختارها همانند یک ریانه مشترک برای تبادل
 اطلاعات هستند، یک استاندار برای کار با داده های نیاز به
 تبادل یا ذخیره به سلیمانی نمایند. در اینجا به معنی برجسته
 از این ساختار های رایم خواهیم پرداخت.

لیکن JSON ← ساختار object Notation ←

یا JSON در تباری داره بسیار یافج است. به خصوص در پایه سازی
 سرویس ها. این ساختار داره های را به شکل key : value
 سازماندهی می کند و کاملاً شبیه به ساختاری شی دیلشنیزی در

python	JSON
dict	object
list, tuple	array
str	string
int, float	number
True	true
False	false
None	null

۱۰ - تعدادی از متدهای json:

۱۱ - ل ۱: json.loads زیانی داشته باشد زیانی داشته باشد.

۱۲ - ل ۲: شسته json به یک فرمت که پایتون استفاده آن را متوجه شود.

۱۳ - json_string to python_type → deserialize / pars

۱۴ - ل ۳: loads ← دستوراتی که اینجا

۱۵ - ل ۴: دلیل با این تفاوت نه به جای خواندن json از یک شسته.

۱۶ - ل ۵: json.load(file_object). آن را از یک فایل می خواند.

۱۷ - L ۶: json_file to python_type

۱۸ - L ۷: dumps ← json.dumps با نقطه مقابل مخالف با

۱۹ - L ۸: باید شما قبل از شسته json دستوراتی که آن را به

۲۰ - L ۹: بایشتری پایتون تبدیل می کرد حال باید ریاستی پایتون را برد

۲۱ - L ۱۰: باید به یک دسته json تبدیل کنید. (json.dumps(data))

۲۲ - L ۱۱: python_type to json_string → serialize

۲۳ - L ۱۲: json.dump(data, file) این تابع دو آرگومان

۲۴ - L ۱۳: در: data آن هایی که باید در یک فایل json نوشته شوند

۲۵ - L ۱۴: و file یک سی فایل نه تواند برای ذخیره داده ها استفاده شود.

۲۶ - L ۱۵: python_type to json_file

ملے CSV ← ساختار value ← comma-separated

یا CSV داده های مانند یک جدول سازماندهی می شوند که داده های موجود در هر سطر آن با استفاده از یک کاراکتر کاما و از پریلر تغییر شده اند. این ساختار در صورت نخسته اطلاعات یا لذارش ها بر روی فایل بسیار رایج است. سطر یکم، سر ایند (Header) اختیاری است.

در پایتون می توان به سادگی با استفاده از مارپول CSV از لایاغانه استاندارد آن، با این نوع فایل کار کرد.

ملے XML ← ساختار language Extensible Markup

یا همان XML یکی از با سابقه ترین نوع از ساختار تباری داده علم رایانش است. این ساختار براساس element یا node می باشد.

های خود را ایجاد می کردد. از این ساختار بیشتر برای ذخیره یا ارسانی پیلرینگ یا تبادل داده در بین پروتکل ها به مانند SOAP

استفاده می شود. هر element شروع < element-name > و پایان </ element-name > مشخص می کرد و می تواند حاوی

یک ساختار یا یک لیست (راجئه - لیست) element دیگر باشد.

در پایتون می توان به سادگی با استفاده از مارپول

ملے xml.etree.ElementTree از لایاغانه استاندارد آن،

با این نوع فایل کار کرد.

* مفهوم شی درایجی :

برنامه نویسی شی درایجی Object-oriented Programming

یا به اختصار OOP یک الگو یا شیوه تغیر در برنامه نویسی است

که برگرفته از زیستی واقعی بوده و از ده سال ۱۹۶۰ میلادی مطرح گشت

است. به زبان که از این الگو پشتیبانی کند، زبان شی درا گفت

می‌شود؛ Simula 67 و Smalltalk نخستین زبان‌های برنامه

نویسی شی درا هستند. ایده شی درایجی را پاسخ به برخی از نیازها که

الموهای موجود پاسخ لورانسان‌بودن به وجود آورد. نیازهای

صفاته؛ توانایی پیاره سازی سائل پیچیده (complex)،

پنهان‌سازی اطلاعات (Data Hiding)، تابلیت استفاده

تجددیستور (Reusability)، وابستگی کمتر به توابع، انعطاف

بالا و ...

رویکرد برنامه نویسی شی درا از پایین به بالا (Bottom-up)

است؛ یعنی ابتدا واحدهایی کوچک از برنامه ایجاد می‌شوند و

سپس با پیوند این واحدهای واحدهای بزرگتر و در نهایت

شلیل کامل از برنامه ب وجودی آید. برنامه نویسی شی درا در

قالب دو مفهوم کلاس (class) و شی (object) ارائه می‌گردد.

هر للاس واحدی از برنامه است که بخشی از عملیات ها و راه های یک برنامه را تعریف می کند و هر شی یک نمونه پیاره هست که از یک للاس است. میتوان هر تعابیر شی از یک للاس را بارگذاری کرد. هر شی مستقل از پیاره هست و خواهد بود که می تواند مقادیر مستعاری را خود نسباری ایند (چهارچوب را توسط للاس) آن تعریف شده است.

در برنامه نویسی شی را، توانی عملیات های قبل از طاری یک برنامه از تعامل اشیائی تواندن باشد که حاصل می شود.

برای امثال کارخانه قویید یک مدل خود را می توانیم به شکل یک ساختار بزرگ در نظر بگیریم. بدین استفاده این کارخانه شامل نشی های تولیدی (و چیزی که مانند فرسنگ الکترونیکی، سیستم چنگ ها، سیستم سوختا، سیستم خنک کننده، موئین و این دیگرها، صراحی شی را) این کارخانه به این صورت می شود که ابتدا می بایست للاس های تولیدی واحد ها را بارگذاری کرد. این دستور هستند و پس از آن به ترتیب للاس هایی که بزرگتر تعریف شده اند را در آنرا بخشی از عملیات های متصل باشد. با این همانند می شوند، این للاس ها را تعریف خود حاوی اشیائی للاس های تولیدی خواهند بود. با استرس این طرح به تعریف للاس را می

۸. در اینجا همان لایس کارخانه است. اینم که
 لایس با هم‌اصلیت لرگ اسیاک واحد های توجهتر از خود و تعریف
 عملیات های موردنیاز، امداده تولید آمونت، های کامل و ممکن انتشار
 خواهد بود. به این صورت که با ایجاد یک اتفاق ازان، پیش

(صوچوییت اجرایی (خودرو) پذیریده آید.

۱۱. از دید برنامه نویسی، هر لایس از دو بخش اعضا (Members) دارد.

۱۲. توابع عضو (Members Functions) تسلیل شده است. اینها

راده در واقعیت همان متغیرهای رون لایس استفاده مخصوصیات

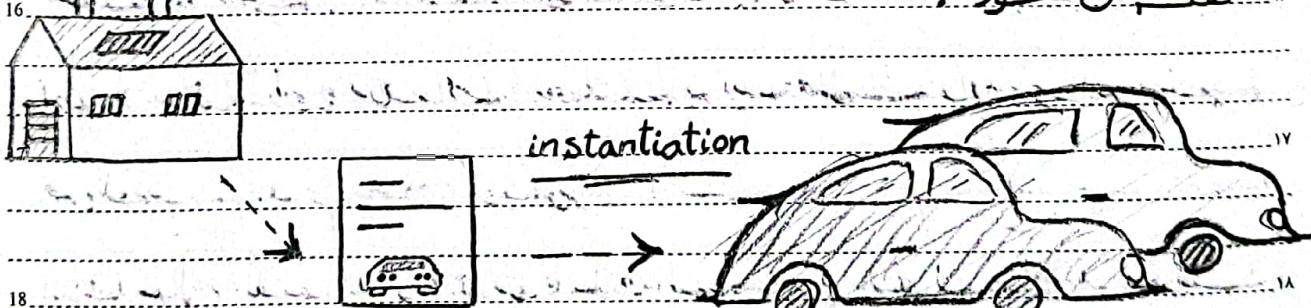
۱۳. و خواص ایشی را بیان می کنند و در شی (Object) با عنوان فیلد (Field)

۱۴. یا صفت (Attribute) از آن های ایجاد می شوند. توابع عضو نیز

۱۵. عملیات های کارهایی هستند که بیش از لایس قادر به انجام

۱۶. آن هایی باشند، در شی (Object) باید تابع صفت (Method)

۱۷. نفته می شود.



۱۸. «CLASS»

۱۹. «object»

۲۰. «2 object»

۲۱. توزیع لایس، گازداری

۲۲. بوق زدن و ...

۲۳. طبل، وزن

۲۴. زبان و ...

- نمونه سازی به بھر شی از یک لاس، یعنی نمونه Instance از آن لاس گفته می شود و به عمل ایجاد شی از لاس بھر شی لاس نمونه سازی Instantiation گفته می شود. در این مفهوم

صف نظری از تعریف لاس ها در برنامه نویسی می باشد.

گاهی نیز وجود یک لاین نظری در بین تعاریف مشترک بین

چند لاس دیگر کاربرد پیدا می کند. درین اساس نوع لاس

در شی دلایل وجود دارد:

۱. لاس های عادی که توانایی

نمونه سازی دارند و به آنها concrete class گفته می شود.

۲. لاس های که توانایی نمونه سازی ندارند و به آنها

Abstract class گفته می شود.

با ایجاد ھر نمونه از لاس یک متغیر خود را به صورت خود کار

اجدا می کرد. این متغیر سازنده "constructor" نام دارد و کار آن

مقدار دهنده اولیه "initialization" نیز است. این کار موجب

اصیان از مقدار دهنده تمامی اعضای داره پیش از استفاده شی در برنامه

می شود.

برای مثال یک لاس خودرو ساره را در نظر بگیرید که در آن صفاتی

زیست بود، طرفیت بالک، بیستین سرعت و متدهای غیر

رانن، دریافت میزان سوخت، تنظیم سرعت، تعریف تعریف

شده است. در همان ابتدا یعنی زمان نمونه سازی کی توانیم با

۱۰ تنظیم صفات توسط متدهای خارجی است، هم نوع و نا جایگزین است.
 ۱۱ صفات ایجاد نماییم، برای همچنان خود را بآن بازنگشتیم
 ۱۲ ۲۰ لیتر و بیشتر بفرعه است ۸۰ لیتر بفرعه برسامن است یا یک خود رفته
 ۱۳ صفتی با اطرافیت بالی ۴۰ لیتر و بیشتر بستگی ۱۶۰ لیتر بستگی
 ۱۴ برسامن در البته هر سه آن های اتمام متدهای دلاس را
 ۱۵ در خود دارند.

۱۶ /
 ۱۷ - پسول سانی (Encapsulation) به در برنامه زنی
 ۱۸ سی درا پسول سانی روشنی است له بجمع دون راههها (data)
 ۱۹ دریک ماقول نمی شوند. پسول سانی، راههها (data) و
 ۲۰ روشن های له راهه هارا مستعاری می کنند (methods)، درینها
 ۲۱ جزو واحد در همی اورده و از داخل خارجی محافظت می کنند.
 ۲۲ پسول سانی عمدها شامل بسته بندی راههها و همین بدلود هایی
 ۲۳ است که از راههها استفاده می کنند.

۲۴ ۲۵ پسول سانی راهه هابه نفعی بسیار صریح پیشان اسانی راهه ها
 ۲۶ منجر می شود: به طور خلاصه پسول سانی برای اتصال راههها
 ۲۷ و عملکرد ها ب دلیل است دریک پسول برای پیشان دون بسیار
 ۲۸ یک دلاس است. در پسول سانی، اعماقی داخل یک دلاس
 ۲۹ می توانند private، protected، public باشند.

۳۰ پسول سانی ادار در جه اول برای پسندیدن دون سیستم استفاده



class (Methods) Variables می شود.

۱- پنهان سازی راهه‌ها (Data Hiding):

در برنامه‌نویسی شی‌گرایی، پنهان کردن راهه‌ها یعنی از مفاهیم بعده است که اینستاکت اعضای یک ملک از این رسانی‌ها عموّج و غیر مالک آن تضمین نمی‌کند، بنابراین پنهان کردن راهه‌ها از اعضای داده در برابر تغییر یا هلاک صفاتی تابعی نموده، همان‌طور که می‌دانیم داروهای حساب (درین) و خواربرین (غمتوان) یک برنامه هستند و سمت‌داران (این راهه‌ها می‌توانند سیمکو نادرستی ایجاد کنند، همچنین می‌توانند برای این جنایت راهه‌ها تأثیر بذلوهند) (درین) (است) برای این جنایت راهه‌ها تأثیر بذلوهند) (درین) (بلات) مثال نباید نماید یک حساب Bank Account در نظر ببرید و یک عضو راهه AC Balance در داخل این مملک ایجاد نمایم، در اینجا موجودی حساب یک اطلاعات حساب (حساب) استاندار سیمکو به مملک خارجی اجازه می‌دهیم موجودی حساب را بدلی نمایم اما اجازه نمی‌دهیم موجودی حساب را تغییر دهیم. از این‌رو در این مرور، با اعلام صفت (attribute) موجودی حساب را محدود می‌نماییم.

۲- اساسی ترین تعاویت بین پنهان سازی راهه‌ها و پیسولو سازی این است که در پنهان سازی راهه‌ها، بر اینستاکت راهه‌ها متمطل است در حالی که پیسولو سازی بر پنهان کردن بحث می‌شود

۳- سیستم صنعتی است.

۱- اساع (Abstraction)

با این سوال خود را آیا خود را کاملاً مخصوص بمنتهی تابعی آید؟

نه، خود را (اساع) است، یعنی مفاهیم و مدل‌های ذهنی هستند و به خود را خود در نیات (ذاتی) ماتحت و جمود ندارند و قابل پیاده‌سازی نیستند.

همچنین روچ رخنه و پراید و بندهای احالتی یعنی و پیاده‌سازی شده

بیک خود را هستند، مفاهیم اساعی (abstract) بیک طرح و اللو (concrete) هستند

برای نمونه عالی واقعی (concrete) هستند

به طور مطلق (Abstraction) ضرایب و جمله های مخفی یا ملزومات بردن برخواهد

ویژگی های غیر ضروری (ایک چیز به منظور کاهش خلاصه شارط)

و به کار بردن آن در محصولاتی از ویژگی های خود را (است)

در برنامه نویسی سئی دار، اساع یعنی از سه مفهوم اصلی ل به همان

در برنامه نویسی سئی دار، اساع یعنی از سه مفهوم اصلی ل به همان

۲- Inheritance و Encapsulation

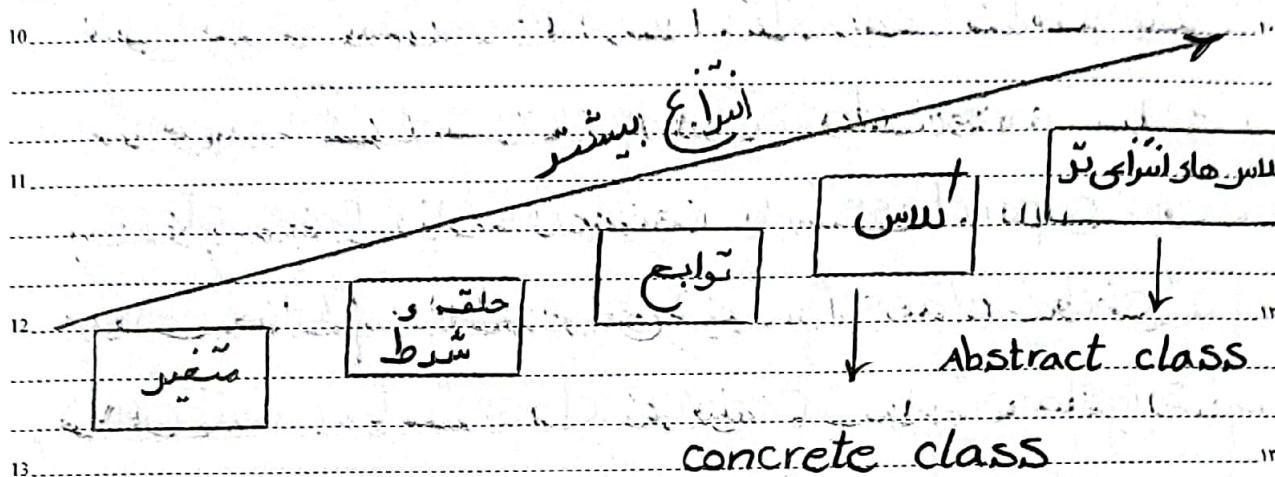
در واقع از طریق نظریه اساع، بیک برنامه نویسی بمحیط از اطلاعات غیر فنلوری مربوط به نیست سئی را به منظور کاهش پیغایل و افزایش کارایی آن بسات می‌کند و اینها نهادی از آن سئی را که حاوی اطلاعات ضروری است، در توضیح برنامه مورخ استفاده آثار خواهد داشت.

لذا می‌توان این اساع بیک هر دو مفهوم (Encapsulation) و

اَسْنَاع - در برنامه نویسی شی بُلّا به همراه چندینی (polymorphism) می آید و توسط دو صورت "للاس های مجرد" و "متدهای مجرد" Abstract Method از اینها می شود.

للاس مجرد، للاسی است که شامل یک یا چند متدهای مجرد باشد و متدهای مجرد، متدهای اعلان (Declare)، شده ولی بین آن تعریف نشده (Define) است. للاس های مجرد، قابلیت نهاد و نوع بتوان از آن سئی ایجاد نمود؛ چرا که هدف او توسعه آنها قرار نرفتن در بالادرین سطح ریاضی بالایی (رخت و راشت کاب عنوان للاس باید براي ارت بری للاس های باین اثر کی باشد. اینde طراحی للاس مجرد در تحسین یک شرکت توسعه بلکه للاس های فرزند آن است؛ تحسین صفات و خصوصیات لازم ولی والدات این تعریف را می توان بر عینde للاس های فرزند به عنوان نفوذ سه للاس ماهی، برابر و بینویس زدن نظر پیرید. این للاس ها از رفتارهای خاص خود (مانند پرواز بردن در بیوت و سفالون در ماهی) یک سری افتاد به مانند سقون (شیخیز)، غذا خواردن و... مستر چستندر راه درست) توسعه این للاس، تحسین یک للاس باید براي هرسدان هایی باشد. ولی از آنجا که هر یکی این رفتارهای مستر را به لونهای بیکر انعام می دهد؛ راه درست آن است.

لهم یک لاس مجبور به عنوان لاس پایه‌ان همان نظر بگیریم؛
در این حالت هر کدام از لاس‌ها ضمن (انستن) رفاه‌های لازم
ساخته شده‌انها را متناسب با باختصار خود تعریف نمایند.



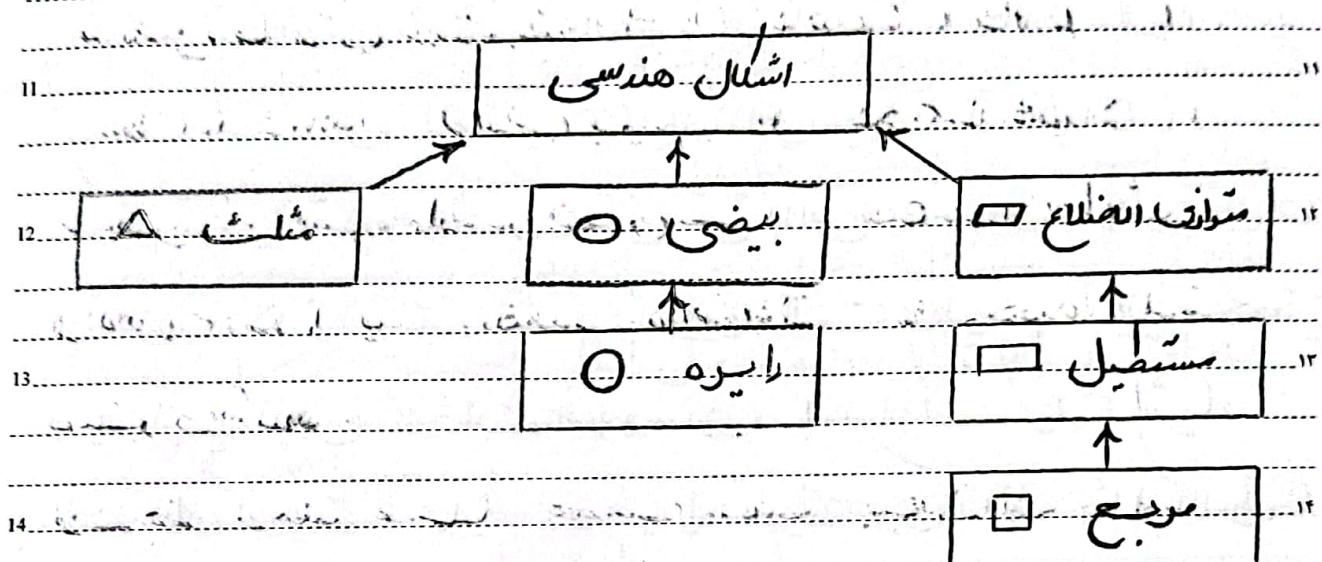
۱۴ - وارثت (Inheritance)

وارثت یکی از اشیاء «قابلیت استفاده مجدد» را بوده‌است.
برنامه‌نویس قادری سازی تابعی از بزرگ‌تر صفات و متدهای
یک یا چند لاس موجوب، لاس‌های جدید را ایجاد نماید.
برای نمونه فرض نیم طرح لاس خود را در مثال نمونه‌سازی،
قصد صراحتی یک مدل خود را جدید با رویکرد باریکی دارد؛ بنابراین
محبایست لاس جدیدی برای تولید آن تهیه نماید. ولی لاس
جدید علاوه بر صفات (ظرفیت‌باریکی و...) و متدهای خاص
خود (الجای باریکی، تعلیم بار و...) به صفات لذک بدنم،
ظرفیت بار و...) و متدهای (راند، توقف، سوخت‌لیری و...).

در این جالت نیاز است ب تعریف مبدأ آن که آنست و می تواند
بر صفاتی و متد های للاس پیش را در للاس جدید بورزش برویم
یک Abstract class، در حق تعاریف مشترک هر دو
للاس ایجاد نموده این مدل خود را می توانند آن را به ارث ببرند.
+ به للاسی که از آن ارث برده شود "parent class" یا
"Base class" و "superclass" (للاس پایه) یا "child class"
به للاسی که اقدام به ارث برده کند "subclass" یا "Derived class" (للاس خرد) یا "Lft" می شود.

ارث بوسط نسبت "هست" یک "بیان می شود
یک از چیزی است که للاس پایه للاس خرد (IS-A Relationship)
ارث بردار؛ در این حالت ای ایم : A is a type of B یعنی دوست است الربلایم: سبب یک نوع میوه است
یا خودرو یک نوع وسیله نقلی است وی توجه داشته باشد
که این یک ارتباط یک طرف از للاس خرد به للاس پایه است.
ونه توایم بلایم: میوه یک نوع سبب است یا
وسیله نقلیه یک نوع خودرو است. X

- کلاس ہامی تواند مستمل باسٹن دلی ھنڈی کے طور رابطہ ھات
 وراثت اسی شوندہ عیلک ساختار سماں صرتیب (Hierarchy)
 بے شکل درختیاً تشیل می رہندے۔ بلکہ نہوں، امانند شکل زیر :
 ل۔ پیلانا ھانشاندرو نسبت اے۔ اے ھستنے)



+ انواع وراثت یوں :

15. 1. وراثت واحد (Single Inheritance) ←

16. 2. وراثت چندگان (Multiple Inheritance) ←

17. 3. وراثت چند سطحی (Multi-Level Inheritance) ←

18. 4. وراثت سلسہ مرتبتی (Hierarchical Inheritance) ←

19. 5. وراثت تولیبی (Hybrid Inheritance) ←

- ترتیب (Composition)

در زمانه نویسی شی لر اشنیست بیلریت بین باعثون نسبت " را در میگیرد " (HAS-A Relationship) وجود را در لئه بیانات مفروض است که بنابراین ترتیب اسنست لئه بیلریت از بیلریت استفاده میگردند که میباشد که مفروض متفاوت با واقع است دارد. این نسبت زمانی بیان میشود که دوست یک لاس (مانند C) از لاس بیلریت (مانند D) بوده بازی سازی شده باشد؛ یعنی سی لاس C را عنوان خودش میگیرد (مانند D) به یاد دارید که خودرو کارخانه خودرو سازی با همها فن درون اشیاء واحد های لوح چیزی، عملیات خود را تعریف میکند مثل لاس صوتی (یعنی درون این لاس یک سی از لاس اصوات را میگیرد) میتوان (میتوان) تولیم بلویسم خودرو یک موتور دارد

وسیله نقایی

حی a

خودرو

has a

موتور

- رابط بین اشیاء رشی دایی :

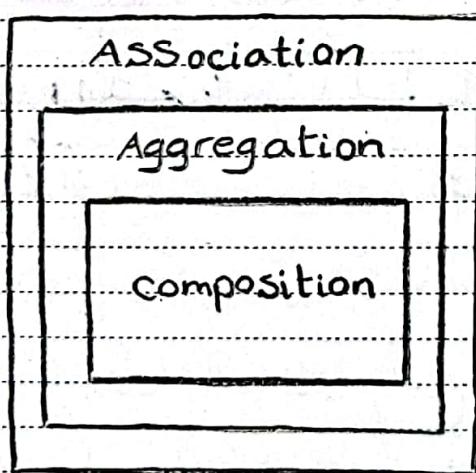
۱) لیه Association \rightarrow اوئی بایلیدیلر رابطه دارند اما
وابستلی فرعی نیست. رابطه حلم نیست و وابستلی شیوه ندارد
بالطبع تعلق و مالیتی وجود ندارد. مثل رابطه صفت، سطرچ و بازیلر.

۲) لیه Aggregation \rightarrow رابطه موک بر است چون رابطه ان
تعلقی است اما وابستلی شدید به بیلیلر ندارد. به عبارت بیلر و قی
للاس اصلی خفی می سود، للاس های بیلر به صورت مستقل باقی
می مانند و از بین آنی روند. مثل رابطه صفت، سطرچ و معرفه ها

۳) لیه Composition \rightarrow بین رابطه رابطه جمله نفتح
می شود زیرا اشیاء رابطه تعلق و وابستلی شدید بهم دارند و قطع
رابطه، روشنی را کاملابی محتاج نمایند. به عبارت وقتی للاس اصلی خفی

می شود، تمامی قسمت های تسلیل دهنده

نیز از بین آنی روند.



مثل رابطه بین اصنفه سطرچ

و خانه های موجود در صفت

۱- چندین‌تی (polymorphism) :

معنی چندین‌تی بسیار توانایی للاس فرزند را باز تعریف کرده‌اید است که در للاس پایه موجود نباشد. للاس نمونه دو للاس ماهی و درب را که هر دو آن‌ها این للاس به نام حیوان ارث برده‌اند را در نظر بگیرید؛ در للاس حیوان ممکن با امنیت خدا خود را اعلیٰ مُستَر کردن می‌توان. تماش حیوانات است وجود دارد ولی از آن‌ها چلولی انجام‌لان در ماهی و درب متفاوت است، بنابراین این دو للاس‌ها از دارند تا ممکن خدا خود را شخصی خود را راست باشند. در این‌است این متد در للاس‌ها فرزند باز تعریف می‌شوند، به این عمل Method overriding لغت، می‌شود.

با override بودن یک متد، صفت للاس پایه زیر سایر صفات مشابه در للاس فرزند خود می‌گیرد و انتظار اشیاء للاس فرزند پنهان می‌شود.

* اشیاء در پایتون :

علاوه بر اینکه یک زبان برنامه‌نویسی شی کلاس است، ساختار آن زیر بر مبنای شی کلاس توسعه یافته است و اینطور بیان می‌شود که هر چیز در پایتون یک شی است، اشیاء اسram پایتون براز ارایه نوع را (Data types). هستند. به بیان دیگر تماش راه‌های یک برنامه پایتون یا به صورت مستقیم یک شی است یا از روابط بین اشیاء ایجاد می‌گردد.

برای نوون : ۵۶ (int str "Hello world!") .

توابع و ... حتی خود للاس هایین توسط یک نوع شی ایجاد می شوند.
هر شی در پایتون حاوی یک شناسه (identity) یک نوع (type) و یک مقادیر (value) است.

شناسه ارزمان ایجاد شی بآن اختصاصی می یابد و غیرقابل

تجزیس است. تابع (def) هم شناسه شی را به صورت یک عدد
صحیح برای آدرس داده این مقادیر بیانگر نشانی (Address) شی است.

حافظه است (Memory)

هر شی در پایتون یاری یک نوع یا type است که در واقع
معرف للاس است که شی از آن ایجاد شده است. نوع

هنئی توسط تابع (type) قابل مشاهده است.

نته ① : هر للاس خود یک شی از للاس type است و همه للاس ها
از للاس به اسم object ایجاد شده اند.

نته ② : در ساختار جدید پایتون اضفیم للاس برابر مفهوم type در نظر

لرفتنی شده است.

نته ③ : صفت class نام للاس یک شی و صفت

نام للاس های پایه یک للاس را نمایش می دهد.

* برنامه نویسی شی کلاس (object-oriented programming) یک پایتون می باشد که بر اساس برنامه نویسی چند الگوی (multi-paradigm) است و از الگوهای مختلف از جمله شی داری، شستاری می باشد. در این شیوه بدهات ابرنامه در مالکا موجود است اها بوجلی برنامه لاس (class) به وجود می آید. لاس ها چنین نیستند جز اینکه توسط آن می توان راه ها و عملیات مرتبط با یک دیگر را دریافت و جدا از سایر بخش های کوچک داد. بالاتر لاس ها رفتار و بعمله هر یکی از افراد شخص است و با سایر بخش های کوچک داخل یکدیگر نباشند. اینداد یک لاس به معنای ایجاد یک نوع (Type) جدید در برنامه می بشود که می توان چنین شی (object) یا نمونه (Instance) از آن نوع ایجاد کرد. یک برنامه نویسی شی کلاس حاصل ارتباط و تعامل اثنا عشر مختلف ایجاد شده ترکان است.

* تعریف لاس :

در پایتون برای تعریف لاس از کلمه کلی class استفاده می کرد. همانند الگو زیر :

`class Class Name :`

`<statement - 1>`

`:`

`<statement - N>`

لیستهای تعریف کلاس (class) یک دستور اخراج پذیر

برای توابع یک کلاس بیش از اجراء دستور خود هم تأثیر

در برنامه دارد. این شرایط اسباب می‌شود که حتی بتوان یک

کلاس را در میان بذلت دستور شرط (if) یا دوی یک تابع تعریف

کرد.

بعد از لیستهای کلاس (class)، نام کلاس (به لغوه کلیه) ذوست

می‌شود. سطر نخست تعریف مانند تما مسورة ای صلب

(compound) که به صورت محمول در چند سطر توییز می‌شوند و

سرازیند رارند، به کراتری ختم می‌شوند. از سطر رقم بازیافت

یکنواخت ارزشی دسواران بذلت کلاس توییز می‌شوند.

نکته ۱: کلاس‌ها نیز همانند توابع، حوزه (scope) خود را دارند.

نکته ۲: یک کلاس تعریف نماینده صفت (Attribute) و رفتار

های (Method) اسیایی است که از آن ایجاد خواهد شد.

در واقع نام کلاس معرف نوع (Type) اسیایی خود است.

به بیان ساده‌تر، هر کلاس می‌تواند شامل تعدادی توابع و متغیر

در داخل بذلت خود باشد. که به متغیرها ← و Attribute

به تابع ← Method ← نکته می‌شود.

توجه داشته باشید که تعریف اهدایی باستینس (زیر نیز از لاس در زبان پایتون صحیح می باشد) به طور کلی برانست (بر جلوی نام لاس رمزانی مرا می کند) قصد بسیار سازنده مفهوم و اشتراک ارائه می باشد و در غیر این صورت نیازی به مراحل را برانست نمی باشد.

`class DerivedClassName (BaseClassName):`

`pass`

* نمونه سازی (Instantiation) :

به عملیات ایجاد یک شی از لاس نمونه سازی (instantiation) می شود لاس چنین جزئی که در نوشتار شده بست و جای در حافظه نظارت این اسیا ایجاد شده از لاس هستند در حافظه (Memory) مرا می کنند. نمونه سازی از یک لاس بر پایتون به صورت زیر انجام می شود:

`class Sample:`

`pass`

`sample_object = Sample()` * instantiation

بر پایتون با ملاحظه نام لاس (همچون فراخوان یک تابع) یک شی از آن لاس ایجاد می کرد. از هر لاس می توان بی نایت نمونه سازی داشت.

هر سئی از یک لیس، حوزه مخصوص به خود را دارد و سجداً از دیگران
لشی، آن لیس خواهد بود. بنابراین اشیا، هر لیس را می‌مستغل
و این‌وقت از یکی‌یکش هستند.

۱ class sample :

۲ class :

۳ obj - ۱ = Sample()

۴ obj - ۲ = Sample()

۵ id (obj - ۱) ۱۳۹۹۳۶۵۱۲۹۶۶۸۴۰

۶ id (obj - ۲) ۱۳۹۹۳۶۵۱۲۹۶۷۰۰۸

همان‌طور که از خروجی تابع آمده id مئضی است، هر سئی جدید
از لیس identity یا هوتی مستغل از دیگلو اشیا را نشان می‌نماید.
در متن جذگاهی از حافظه قابل واده می‌شود.

در پایتون دو شیوه یا برابری باشند نوع یک سی وجود دارد:

۱ تابع آناده type

۲ تابع آناده isinstance (obj, type)

چنانچه سی از نوع دریافت شده باشد \leftarrow True

* سازنده (Constructor)

در بحث شی دریا، هنلار ساختایل شی لایبادیک نوونه جدید ()

به صورت خودکار یک متاد از لالس سور نظر ضرخوانی می شود. به این متاد رسازنده لفته می شود. ضرخوانی خودکار این متاد به برنامه نویس این املا را می دهد له در صورت تایل بتوانه چدونلی ایجاد شی

جدید را میریت یاد همان هنلار ساخت. شخص سازی نماید.

از طرف هر لالس رسازیان برنامه نویس یا پیون شامل یک سرچ متاد خاص می باشد که نام آن ها با دو کاراکتر خط زیرین (underscore or underline) شروع و نیز یا بیان می یابد. همانند `__init__`

(دو کامیزیتی یا پیون) یعنی دو کاراکتر خط زیرین برینان هم به اصطلاح Dunder (Double underscores)

به این متدهای خاص در یا پیون به اصطلاح Special Methods یا Magic Methods یا Dunder Methods

باید توجه داشت که نام این متدها یک هیاهو سازی بیش خوب در یا پیون دارند و الزاماً برای هیاهو سازی وجود ندارد.

در فرایند نوونه سازی از لالس یا پیون، به ترتیب اول متاد خاص در یا پیون است: `__new__` و `__init__`

۸ - سند - new - در زمان ایجاد شی و رئیفیابی ایجاد شی مرا خوانی .

۹ - محتشود - خروجی این متد یک شی جدید از آن کلاس می باشد .

۱۰ - این متد از نوع static Method است . نخستین بار این متد

۱۱ - این متد ، کلاس است که متران است از آن یک شی ایجاد شود .

۱۲ - پارامترهای دیگر را می توانند حاوی مقادیر باشند که در مصلحت

۱۳ - نمونه سازی ارسال شده است .

۱۴ - متد init - بلا ماحله بس از این شی جدید توسط

۱۵ - متد new - ایجاد شود و درست قبل از این شی جدید از

۱۶ - متد new - بر بروز رسانده شود (returned) ، مرا خوانی می کرد .

۱۷ - این متد از نوع Instance Method است . نخستین

۱۸ - پارامتر این متد شی لجای ارسال است (همان شی ای داشت) .

۱۹ - ایجاد شود) و پارامترهای دیگر بعنوان نویس (در زمان نمونه سازی

۲۰ - جست مقدار دهی در شی ارسال می کند . توجه داشته باشید که این

۲۱ - متد خروجی از آن (return) بعون دستور return - خروجی آن none (

۲۲ - و شی جدید حاصل خروجی متد new - خواهد بود .

۲۳ - در فرایند نمونه سازی در زبان برنامه نویسی پاسون (این دو متد

۲۴ - با یکدیگر کار می کنند و نقش نمونه را ایفا می کنند . متد new -

۲۵ - شی را ایجاد (create) و متد init - آن را شخص سازی

۲۶ - 20 - customize () می کند .

نکته ۱: زبان برنامه نویسی پایتون بخلاف فارسی از زبان های شی را به ساخته Java، از املاک های سازنی چنین constructor پیشیانش نمی نماید. البته برنامه نویس با روئش هایی می تواند اب صورت منطقی به هدف خود برسید.

نکته ۲: از زمانی به پیاده سازی دو متده new و init برای نمونه سازی از لالاس یافت می شود. در این موقع new پیاده سازی نمی شود اما زمانی که می خواهیم در زمان نمونه سازی مقادیری در شی تنظیم نماییم، لام است متد init را پیاده سازی نماییم.

نکته ۳: معمولاً new زمانی پیاده سازی می شود که بنواییم صدور می تا های در ایجاد شی لالاس مورد نظر ایجاد نیم. برای نمونه سازی طرح Singleton یک لالاس.

نکته ۴: انسان ارتومن در زمان نیوون سازی اسئی یا همان پیاده سازی متد init ب برنامه نویس این اطمینان را حداچی کرده تئی جدید در یک وضعيت اریست تنظیم شده است.

نکته ۵: ارتومن های متناظر با پارامترهای متده init را به جز Self که توسط مفسر پایتون (مقدار دهنده می تردد) می باشند در زمان نمونه سازی و غیر خواندن لالاس ارسال نردد.

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

def __init__(self, [arguments])

```

8 1 class person:
9
10 2     def __init__(self, name): # constructor
11
12 3         self.name = name # attribute
13
14 4     def say_hi(self): # sample Method
15
16 5         print('Hello, my name is', self.name)
17
18 6 p1 = person('Mahdi')
19
20 7 p2 = person('Karen')
21
22 8 p1.say_hi() >>> Hello, my name is Mahdi
23
24 9 p2.say_hi() >>> Hello, my name is Karen

```

* صفات (Attributes)

به بیان ساده، متغیرهایی را به یک کلاس یا یک شی اتصاب
ناره می‌شود. صفت یا ویژگی خوانده می‌شود. در بحث‌اشی دو ای
نیان برنامه‌نویسی یاد موند. نوع وجود دارد:

Instance Attribute ← .

class Attribute ← .

Instance Attribute -

به Attribute های خاص یا شی لفته می‌شود. به هر شی در
پایین می‌توان با استفاده از سینتакс زیر یک Attribute اتصاب داد:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

| object.attribute-name = value |

```

1 class Sample:
2     pass
3 sample_obj = Sample()
4 sample_obj.a_new_attribute = new_att!
5 print(sample_obj.a_new_attribute)
6 >>> new_att!

```

هر چیزی در پایتون یک شی است و ممکن است مفسر پایتون برای

بُخ اشیاء محدود است های در نظر نرفته باشد و شما نتوانید به هر

شی ای در پایتون Attribute افای نمایید. در اینجا جاداره اشاره

شود به بخش Function Attribute (۳۰ صر) که در واقع

کافی جزوی از Attribute به شی تابع نبود.

بر مثال قبل، این تسویه از افزودن Attribute به اسیا لاس

های ای خودمان آن را تعریف می کنیم چنان جالب نیست و

ممکن است باعث بروز خطاهای منطقی در برنامه شود، بجز

است این که توسط متode init که در واقع

اسیا پایتون است انجام نماید. (همانند مثال ۳۰ دی)

لسته: مقادیر این لسته از Attribute ها به ای ارزشی مخصوص به خود است.

بلطفاً نهاده: صفت ای هایی همچون نام و نام خانوادگی، سن، برآمدگی از

اسیاء لاس شخص قابل تعریف است و هر چونه شی از این لاس

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	۲۵	۲۶	۲۷	۲۸	۲۹	۳۰
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

می بایست شامل مقادیر مخصوص به فرد از این Attribute ها باشد.

class Attribute

۸ های خاصی یک لاس لفته می شود و در واقع متغیر Attribute
 ۹ های است از سمعن لاس و خارج از متدها تعریف می شود.
 ۱۰ کاربر این ها ب استرال لذیثن یک یا چند
 ۱۱ مقدار یکسان درین تمام اشیاست.
 ۱۲ تمام اشیای یک لاس به class Attribute های آن کامل
 ۱۳ مسترسی را دارد.

1 class Sample :

2 class_attribute = 0

3 Pass

4 print('Line 4 : ', sample.class_attribute)

5 sample_obj1 = Sample()

6 Sample_obj2 = Sample()

7 print('Line 7 : ', sample_obj1.class_attribute)

8 sample.class_attribute = 1

9 print('Line 9 : ', sample_obj1.class_attribute)

10 sample_obj2.class_attribute = 2

11 print('Line 11 : ', sample_obj1.class_attribute)

12 print('Line 12 : ', sample_obj2.class_attribute)

نکت ۱: مقدار class ها هم با استفاده از نام کلاس

قابل دسترسی است (سطر ۴) و هم با استفاده از هر یک از

اسیاد آن کلاس (سطر ۷)

نکت ۲: برای تغییر مقدار class Attribute در داخل کلاس از

کلاس (سخ در این) استفاده می شود و در پیورن

کلاس با استفاده از نام کلاس به صورت زیر (سطر ۸) :

`class Name . class _ attribute = new _ value`

نکت ۳: برای تغییر مقدار class Attribute از شی استفاده نمی شود!

با این کار تنها یک Instance Attribute برای آن شی ایجاد

می شود (سطر ۱۰)

* متد (Method) :

متدها در واقع توابع هستند که داخل هر کلاس تعریف می شوند.

هر کلاس پایتون میتواند شامل سه نوع متده باشد :

۱. Instance Method ← هر زمان در پیاده سازی یک کلاس،

به شی جاتی از کلاس یا Instance Attribute ها نیاز داشتم

می بایست این نوع متده را پیاده سازی نیم.

۲. Class Method ← هر زمان در پیاده سازی یک کلاس،

ب خود کلاس یا Class Attribute ها نیاز داشتم می بایست

۳. ان نوع متده را پیاده سازی نیم.

۴. ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

(توضیح هر یک از متد ها در ارامه ...)

۵

پیشنهاد
بهمن

۱۳۹۹

24 January 2021
۱۰ جمادی الثانيه ۱۴۴۲

۳ متد static Method ← هر زمان رسیداده میانی یا لاس .
۴ به لاس و به اسیاں آن لاس نیاف نداشیم یا صدم پیاده سازی کار
۵ مستعمل آن رفیق لی لاس مورد نظر را داشیم ، می بایستی این نوع متد با
۶ پیاده سازی نمیم .

۷ class sample :

۸ def instance _ method (self) :

۹ pass

۱۰ @class Method

۱۱ def class _ method (cls) :

۱۲ pass

۱۳ @ static method

۱۴ def static _ method () :

۱۵ pass

* وراثت (Inheritance) :

۱۶ وراثت به معنای املاک است که یک لاس بتواند صفات و
۱۷ رفتارهای یک لاس دیگر را تذییب همراه خود ورثت باشد . (۱۴ می)
۱۸ پیاده سازی وراثت (رسیداده) حافظ بودن لاس نیاز دارد :
۱۹ superclass / base class ← لاس اصلی یا لاسی که
۲۰ می خواهد لاس یا لاس های دیگر آن را به ارث ببرد .
۲۱ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

۲۲ - drived class / sub class ← لاس یا لاس هایی که از لاس
۲۳ - ارث بری خواهند داشت super class