ترفندها و تکنیک‌های مطرح‌شده در کتاب

۱. مجاز نبودن استفاده از ارقام در شروع نام متغیر

۲. مجاز نبودن استفاده از عملگرها در شروع نام متغیر

۳. مجاز نبودن استفاده از رقم ۰ در شروع اعداد

۴. مجاز بودن استفاده از کاراکتر _ در هر جای نام متغیر

۵. هرگز از کلمات متفاوت در نام‌گذاری شناسه‌ها استفاده نکنید

۶. تقسیم اعشاری و تقسیم صحیح

۷. استفاده از مقادیر منطقی در عملیات حسابی

۸. تخصیص زنجیره‌ای

۹. مقداردهی چندگانه

۱۰. مقداردهی چندگانه‌ی پیشرفته‌تر

۱۱. تابع print پیشرفته‌تر

۱۲. f-Strings برای قالب‌بندی رشته‌ها

۱۳. باز کردن (Unpacking) مقادیر

۱۴. جابه‌جا کردن مقادیر دو متغیر در یک سطر

۱۵. تخصیص مقادیر یک لیست در چند متغیر جدید

۱۶. کاربرد عملگر _

۱۷. انواع داده‌های بولی که در پایتون False هستند

۱۸. داده‌های صحیح

۱۹. نمایش اعداد صحیح در مبناهای ۲، ۸ و ۱۶

۲۰. تبدیل اعداد صحیح مبنای ۱۰ به مبناهای ۲، ۸ و ۱۶ و بلعکس

۲۱. تابع bit_length()

۲۲. محاسبه میزان حافظه RAM مصرف‌شده اشیا

۲۳. دیباگ کردن

۲۴. استفاده از _ برای جداسازی ارقام اعداد بزرگ

۲۵. ذخیره رشته‌های چندخطی

۲۶. عملگرهای رشته

۲۷. اعداد مختلط

۲۸. خواندن داده به‌صورت کلمه عبور

۲۹. خواندن چند داده در یک خط

۳۰. کاربرد تابع id()

۳۱. کاربرد تابع ord()

۳۲. کاربرد تابع chr()

۳۳. کاربرد تابع type()

۳۴. کاربرد تابع eval()

۳۵. پیدا کردن نام سایت بسته‌های پایتون

۳۶. اجرای یک برنامه خارجی در پایتون

۳۷. تعیین نام و مسیر فایل فعلی در حال اجرا

۳۸. تعیین تعداد CPUها در پایتون

۳۹. نمایش اطلاعات کپی‌رایت پایتون

۴۰. پلت فرم - اطلاعات نسخه سیستم

۴۱. مفسر

۴۲. platform

۴۳. اطلاعات سیستم‌عامل و سخت‌افزار

۴۴. معماری اجرایی

۴۵. توابع min و max

۴۶. نمایش مقادیر متغیرهای محیطی

۴۷. اندازه فضای حافظه موردنیاز برای رشته‌ها

۴۸. چاپ شکلک

۴۹. استفاده از عملگر >> برای محاسبه ضرب

۵۰. استفاده از فرمت n+nn+nnn

۵۱. نکات ماژول struct

۵۲. نمایش آرگومان‌های خط فرمان و تعداد آن‌ها

۵۳. جداسازی نام فایل از مسیر

۵۴. تعیین مقدار nامین بیت یک عدد

۵۵. یک کردن مقدار nامین بیت یک عدد

۵۶. صفر کردن مقدار nامین بیت یک عدد

۵۷. معکوس کردن مقدار nامین بیت یک عدد

۵۸. معکوس کردن تمام بیت‌های یک عدد

۵۹. زمان در سیستم‌های کامپیوتری

۶۰. زمان فعلی همراه با تاریخ امروز

۶۱. تبدیل ثانیه به ساعت و تاریخ روز

۶۲. تبدیل زمان و تاریخ خاص به ثانیه

۶۳. چاپ زمان با فرمت خاص زمان در پایتون

۶۴. تبدیل زمان با فرمت به ثانیه

۶۵. ایجاد وقفه در اجرای برنامه با تابع sleep زمان در پایتون

۶۶. تعریف delay تصادفی در پایتون

۶۷. راه‌کار ذخیره زمان در برنامه‌نویسی

۶۸. محاسبه زمان اجرای برنامه در پایتون

۶۹. ماژول datetime

۷۰. اضافه نمودن n ثانیه به زمان فعلی

۷۱. استفاده از Help()

۷۲. if else درون‌خطی

۷۳. a or b

۷۴. a and b

۷۵. مقایسه زنجیره‌ای (a < x < b)

۷۶. ساده‌سازی if با عملگرهای in و not in

۷۷. ساده‌سازی if elif else با Match...case

۷۸. ساده‌سازی if elif else با دیکشنری

۷۹. عبارت انتساب: عملگر Walrus

۸۰. استفاده از دستور pass در دستورات شرطی

۸۱. استفاده از دستور pass در حلقه‌های تکرار

۸۲. تفاوت بین دستورات pass و continue چیست؟

۸۳. حلقه‌های تک‌خطی

۸۴. حلقه‌های تکرار تودرتو خلاصه‌شده

۸۵. نوشتن حلقه‌های پایتونیک

۸۶. استفاده از تابع enumerate() در حلقه برای بازیابی اندیس

۸۷. استفاده از تابع range() برای ایجاد حلقه‌های به سبک C و جاوا

۸۸. بخش else در حلقه for

۸۹. بخش else در حلقه while

۹۰. مباحث پیشرفته‌تر از for و if جهت ایجاد لیست‌ساز و عبارت مولد

۹۱. دستور pass در تابع

۹۲. قرار دادن جانگه‌دار

۹۳. توابع اشیاء هستند

۹۴. در پایتون توابع را می‌توان به‌عنوان آرگومان به توابع دیگر ارسال کرد

۹۵. توابع می‌توانند تابع دیگری را برگردانند

۹۶. کلمه کلیدی yield

۹۷. تفاوت بین yield و return پایتون

۹۸. آرگومان‌های کلمه کلیدی

۹۹. آرگومان‌های با مقدار پیش‌فرض

۱۰۰. برگرداندن چندین مقدار با تابع

۱۰۱. args * و kwargs ** در توابع

۱۰۲. *args چیست؟

۱۰۳. **kwargs چیست؟

۱۰۴. استفاده از args * و kwargs ** در هنگام فراخوانی یک تابع

۱۰۵. استفاده از *args و **kwargs برای تنظیم مقادیر شیء

۱۰۶. توابع تودرتو چیست؟

۱۰۷. بستارها و توابع Factory

۱۰۸. توابع بی‌نام (لامبدا)

۱۰۹. متغیرهای محلی

۱۱۰. متغیرهای سراسری

۱۱۱. دستور global

۱۱۲. نام متغیر سراسری global

۱۱۳. باز کردن آرگومان‌های تابع با استفاده از عملگر splat (*)

۱۱۴. استفاده از دیکشنری و تابع برای پیاده‌سازی switch/case

۱۱۵. محاسبه فاکتوریل عدد در یک خط

۱۱۶. تعریف چند متد هم‌نام در یک بلاک

۱۱۷. پیاده‌سازی سری فیبوناچی در یک خط

۱۱۸. هرگز بیش‌ازحد از if-else تودرتو در توابع استفاده نکنید

۱۱۹. "__main__" == __name__ if چیست؟

۱۲۰. ارسال آرگومان پیشرفته به توابع

۱۲۱. تابع یک‌خطی تعریف‌شده توسط کاربر برای پیدا کردن کوچک‌ترین عدد بین سه عدد

۱۲۲. کپی لیست با استفاده از عملگر انتساب (=)

۱۲۳. تابع copy() کم‌عمق

۱۲۴. تابع deepcopy()

۱۲۵. فرق بین عملگر =، توابع copy() و deepcopy()

۱۲۶. فراخوانی min() و max() با یک آرگومان تکرارپذیر

۱۲۷. استفاده از توابع min() و max() با رشته‌های تکرارشونده

۱۲۸. پردازش دیکشنری با min() و max()

۱۲۹. بهینه‌سازی رفتار استاندارد توابع min() و max() با آرگومان‌های key و default

۱۳۰. استفاده از min() و max() با لیست‌سازها (Comprehensions) و عبارت مولد (Generator)

۱۳۱. قرار دادن توابع min() و max() در عملیات

۱۳۲. حذف کوچک‌ترین و بزرگ‌ترین اعداد در یک لیست

۱۳۳. ایجاد لیست‌های حداقل و حداکثر مقادیر

۱۳۴. برش مقادیر در لبه‌های یک بازه اعداد

۱۳۵. یافتن نزدیک‌ترین نقاط

۱۳۶. شناسایی ارزان‌ترین و گران‌ترین محصول

۱۳۷. بررسی شیء zipشده

۱۳۸. عدم ارسال آرگومان به تابع zip

۱۳۹. ارسال یک آرگومان به تابع zip

۱۴۰. ارسال چندین آرگومان به تابع zip

۱۴۱. تابع zip و رشته‌ها به‌عنوان آرگومان

۱۴۲. فراخوانی min() و max() با یک آرگومان تکرارپذیر

۱۴۳. استفاده از توابع min() و max() با رشته‌های تکرارشونده

۱۴۴. پردازش دیکشنری با min() و max()

۱۴۵. بهینه‌سازی رفتار استاندارد توابع min() و max() با آرگومان‌های key و default

۱۴۶. استفاده از min() و max() با لیست‌سازها (Comprehensions) و عبارت مولد (Generator)

۱۴۷. قرار دادن توابع min() و max() در عملیات

۱۴۸. حذف کوچک‌ترین و بزرگ‌ترین اعداد در یک لیست

۱۴۹. ایجاد لیست‌های حداقل و حداکثر مقادیر

۱۵۰. برش مقادیر در لبه‌های یک بازه اعداد

۱۵۱. یافتن نزدیک‌ترین نقاط

۱۵۲. شناسایی ارزان‌ترین و گران‌ترین محصول

۱۵۳. بررسی شیء zipشده

۱۵۴. عدم ارسال آرگومان به تابع zip

۱۵۵. ارسال یک آرگومان به تابع zip

۱۵۶. ارسال چندین آرگومان به تابع zip

۱۵۷. تابع zip و رشته‌ها به‌عنوان آرگومان

۱۵۸. تابع zip و مجموعه‌ها به‌عنوان آرگومان

۱۵۹. تابع zip و لیست‌ها به‌عنوان آرگومان

۱۶۰. خارج کردن مقادیر فشرده از حالت فشرده با استفاده از تابع zip

۱۶۱. ایجاد حلقه روی داده‌های تکرارپذیر و تابع zip

۱۶۲. استفاده از تابع filter

۱۶۳. تابع filter و لامبدا

۱۶۴. تابع filter و تابع map()

۱۶۵. تابع filter و تابع reduce()

۱۶۶. جایگزین‌های تابع filter()

۱۶۷. جایگزینی تابع filter() با عبارت مولد

۱۶۸. استفاده از map() با توابع دیگر

۱۶۹. تابع map() با تابع لامبدا

۱۷۰. استفاده از توابع map() و Reduce()

۱۷۱. استفاده از توابع map() و filter()

۱۷۲. پردازش چندین آرگومان با map

۱۷۳. تبدیل تکرارهای رشته‌ای با map

۱۷۴. تبدیل تکرارهای عددی با استفاده از map()

۱۷۵. جایگزین‌های تابع map()

۱۷۶. فرق بین توابع map() و starmap()

۱۷۷. ساخت نمونه‌هایی از Counter

۱۷۸. به‌روزرسانی شیء Counter

۱۷۹. دسترسی به محتوای Counter

۱۸۰. Counter با لیست

۱۸۱. Counter با دیکشنری

۱۸۲. Counter یک تاپل

۱۸۳. Counter و عملیات حسابی

۱۸۴. Counter با متدهای مختلف

۱۸۵. تابع all()

۱۸۶. مقادیر True و False در پایتون چیست؟

۱۸۷. تابع all() با تاپل و مجموعه

۱۸۸. تابع all() با دیکشنری

۱۸۹. all با رشته‌ها و اعداد صحیح جداگانه عمل می‌کنند

۱۹۰. پیاده‌سازی تابع All با حلقه for

۱۹۱. نحوه استفاده از تابع all() برای ترکیب چند شرط با and منطقی

۱۹۲. تابع any()

۱۹۳. تابع any() و رشته‌ها

۱۹۴. تابع ()any و لیست‌ها

۱۹۵. تابع ()any و تاپل‌ها

۱۹۶. تابع ()any و دیکشنری‌ها

۱۹۷. بررسی وجود ارقام در یک رشته با استفاده از تابع ()any

۱۹۸. بررسی وجود حروف الفبا در یک رشته با استفاده از تابع ()any

۱۹۹. ترکیب چند شرط or با استفاده از تابع ()any

۲۰۰. فرق بین توابع ()all و ()any

۲۰۱. تابع ()sum

۲۰۲. پیاده‌سازی تابع ()sum

۲۰۳. تابع ()sum و یک تاپل به‌عنوان آرگومان

۲۰۴. تابع ()sum و یک مجموعه به‌عنوان آرگومان

۲۰۵. تابع ()sum و یک دیکشنری به‌عنوان آرگومان

۲۰۶. تابع ()sum و اعداد اعشاری به‌عنوان آرگومان

۲۰۷. تابع ()sum و رشته‌ها و اعداد به‌عنوان آرگومان

۲۰۸. تابع ()sum و رشته offset همراه با لیست اعداد تکرارشونده به‌عنوان آرگومان

۲۰۹. تابع ()sum و انواع داده‌های مختلف به‌عنوان آفست

۲۱۰. تابع ()sum و لیستی از مجموعه به‌عنوان آرگومان

۲۱۱. تابع ()sum و لیستی از لیست و تاپل به‌عنوان آرگومان

۲۱۲. fsum و عدد اعشاری

۲۱۳. مغلوب کردن رشته

۲۱۴. حذف فضاهای غیرضروری بین کلمات رشته

۲۱۵. ساده‌ترین راه برای کپی کم‌عمق یک لیست

۲۱۶. معکوس کردن یک لیست

۲۱۷. عملگر+ در لیست‌ها

۲۱۸. عملگر* در لیست‌ها

۲۱۹. ادغام دو دیکشنری

۲۲۰. پیدا کردن آیتمی با بیش‌ترین تکرار در یک لیست

۲۲۱. تابع enumerate()

۲۲۲. پیمایش چندین لیست با یک حلقه در یک زمان

۲۲۳. تبدیل اولین حرف هر کلمه به حرف بزرگ در یک جمله

۲۲۴. بررسی رشته‌های فرعی

۲۲۵. بررسی این‌که آیا یک لیست خالی است یا خیر؟

۲۲۶. مرتب کردن دیکشنری بر اساس مقدار
۲۲۷. تبدیل چند لیست ذخیره‌شده در یک متغیر به یک لیست تخت
۲۲۸. ایجاد یک رشته از آیتم‌های یک لیست
۲۲۹. گرفتن مقادیر از دیکشنری
۲۳۰. حذف آیتم‌های تکراری لیست: (تابع set)
۲۳۱. تعویض کلیدها و مقادیر دیکشنری
۲۳۲. روش‌های ادغام دیکشنری‌ها
۲۳۳. اجتماع دو مجموعه
۲۳۴. ترکیب لیست‌ها – ادغام لیست‌ها
۲۳۵. عملگر اشتراک (&) در مجموعه‌ها
۲۳۶. عملگر تفاضل (-) در مجموعه‌ها
۲۳۷. عملگر تفاضل متقارن (^) در مجموعه‌ها
۲۳۸. عملگر زیرمجموعه (<) در مجموعه‌ها
۲۳۹. پیدا کردن تفاضل بین دو لیست با تابع symmetric_difference
۲۴۰. تعیین این که آیا همه عناصر یک لیست یکی هستند یا خیر؟
۲۴۱. مقایسه کارا دو لیست نامرتب
۲۴۲. چگونه بررسی کنیم که آیا همه عناصر یک لیست منحصربه‌فرد هستند یا خیر؟
۲۴۳. تبدیل دو لیست به یک دیکشنری با تابع zip
۲۴۴. مرتب‌سازی عناصر یک رشته یا لیست بر اساس تعداد تکرار عناصر
۲۴۵. پیدا کردن عنصری با بیش‌ترین تکرار در یک لیست فقط در یک خط
۲۴۶. حذف تمام عناصر از یک لیست، مجموعه یا دیکشنری
۲۴۷. شمارش تعداد دفعاتی که یک عنصر در یک تاپل ظاهر می‌شود
۲۴۸. شمارش تعداد دفعاتی که یک عنصر در لیستی ظاهر می‌شود
۲۴۹. بررسی این که آیا در یک رشته تمام کلمات آن با حروف بزرگ شروع می‌شوند یا خیر؟
۲۵۰. بررسی این که آیا یک رشته نشان‌دهنده یک عدد چینی است یا خیر؟
۲۵۱. بررسی این که آیا تمام حروف یک رشته ارقام هستند یا خیر؟
۲۵۲. حذف فضاهای خالی سمت چپ یا راست یک رشته یا حذف کاراکترهای سمت چپ یا راست یک آرگومان
۲۵۳. بررسی این که آیا تمام کاراکترهای یک رشته حروف الفبا هستند یا خیر؟
۲۵۴. بررسی این که آیا تمام کاراکترهای یک رشته حروف الفبا یا ارقام عددی هستند یا خیر؟
۲۵۵. بررسی این که آیا تمام کاراکترهای یک رشته فضای خالی هستند یا خیر؟
۲۵۶. رشته‌ها و تاپل‌ها تغییرناپذیرند.

۲۵۷. لیست‌ها، مجموعه‌ها و دیکشنری‌ها قابل‌تغییر هستند

۲۵۸. دستور pass در هنگام تعریف کلاس

۲۵۹. خاصیت __name__ کلاس

۲۶۰. تعیین نوع شیء

۲۶۱. صفات کلاس

۲۶۲. بازیابی مقادیر متغیرهای کلاس

۲۶۳. مقداردهی به صفات کلاس

۲۶۴. حذف صفات کلاس

۲۶۵. بررسی وجود صفتی در یک کلاس

۲۶۶. ذخیره‌سازی صفات کلاس

۲۶۷. ویژگی‌های قابل فراخوانی کلاس

۲۶۸. چه زمانی از ویژگی‌های کلاس در پایتون استفاده می‌کنیم

۲۶۹. متغیرهای نمونه

۲۷۰. متدهای نمونه

۲۷۱. متدهای استاتیک

۲۷۲. متدهای استاتیک در مقابل متدهای کلاس

۲۷۳. مثال‌های متد استاتیک پایتون

۲۷۴. متدهای __init__() و __new__()

۲۷۵. متد __str__()

۲۷۶. متد جادویی __repr__()

۲۷۷. متدهای استاتیک

۲۷۸. متدهای استاتیک در مقابل متدهای کلاس

۲۷۹. مثال‌های متد استاتیک پایتون

۲۸۰. متدهای __init__() و __new__()

۲۸۱. متد __str__()

۲۸۲. متد جادویی __repr__()

۲۸۳. متد __eq__()

۲۸۴. تابع hash()

۲۸۵. متد __bool__()

۲۸۶. متد __len__()

۲۸۷. متد __del__()

۲۸۸. متد __call__()

۲۸۹. کپسوله سازی(بسته‌بندی)

۲۹۰. ویژگی‌های خصوصی

۲۹۱. نام‌هایی با دو خط زیر (__)

۲۹۲. متدهای دریافت‌کننده (بازیابی کننده) و تنظیم‌کننده (مقدار دهنده)

۲۹۳. دکوراتور property

۲۹۴. دکوراتورهای setter

۲۹۵. ویژگی فقط خواندنی

۲۹۶. تعریف ویژگی‌های قابل‌محاسبه در حافظه نهان (کش)

۲۹۷. حذف ویژگی

۲۹۸. مقایسه شیء:"is" و "=="

۲۹۹. تبدیل رشته (هر کلاس به __repr__ نیاز دارد)

۳۰۰. تفاوت بین __str__ و __repr__

۳۰۱. چرا هر کلاس به __repr__ نیاز دارد

۳۰۲. تابع __unicode__

۳۰۳. ایجاد یک مقدار ثابت در کلاس

۳۰۴. ایجاد سازنده کلاس چندگانه

۳۰۵. ایجاد Enum

۳۰۶. کلاس‌های Iterators

۳۰۷. دسترسی به یک کلاس به‌عنوان یک لیست

۳۰۸. پیاده‌سازی وراثت

۳۰۹. اصطلاحات وراثت

۳۱۰. نوع در مقابل نمونه

۳۱۱. پیاده‌سازی مجدد متد

۳۱۲. پیاده‌سازی مجدد عملگرها