

خواندن کد (Code Reading)

1. پیش‌بینی خروجی یک مثال از دستور `if/else`
2. پیش‌بینی خروجی یک حلقه شامل `break` و `range()`
3. با توجه به یک قطعه کد مربوط به دستکاری لیست یا دیکشنری، ساختار نهایی را بنویسید
4. پیش‌بینی مقدار بازگشتی یک تابع

جای‌خالی را پر کنید (Fill in the Blanks)

- کدی را کامل کنید که میانگین یک لیست را محاسبه می‌کند
- جای خالی را پر کنید:

```
def greet(name): _____ "Hello " + name
```
- جای خالی‌ها را در تعریف یک دیکشنری پر کنید

نوشتن کد (Writing Code)

1. تابعی به نام `add(a, b)` بنویسید که مجموع دو عدد را برگرداند
2. یک حلقه بنویسید که اعداد ۱ تا ۱۰ را چاپ کند
3. کدی بنویسید که:
• یک دیکشنری ایجاد کند
• یک کلید جدید به آن اضافه کند
• تمام کلیدها را چاپ کند

حل مسئله عملی (Practical Problem Solving)

مسئله ۱

تابعی به نام `count_characters(text)` بنویسید که تعداد حروف موجود در یک رشته را برگرداند

مسئله ۲

با توجه به لیست نمرات دانش‌آموزان زیر:
[53, 96, 85, 32, 70, 90]

کدی بنویسید که:

- تعداد نمراتی که بزرگتر یا مساوی ۶۰ هستند را بشمارد
- بالاترین نمره را پیدا کند

خروجی کد را مشخص کنید

سوال اول –

```
x = 10
y = 5

if x > y:
    print("A")
else:
    print("B")
```

سوال دوم –

```
total = 0
for i in range(1, 5):
    total += i
print(total)
```

سوال سوم –

```
data = {"a": 1, "b": 2}
data["c"] = 3
data["a"] = 5

print(data)
```

سوال چهارم –

```
def square(n):
    return n * n

result = square(4)
print(result)
```

سوال پنجم –

```
my_dictionary = dict()
for i in range(1,3):
    name = input("please enter your name: ")
    age = int(input("please enter your age: "))
    my_dictionary[name] = age

with open("final_file.csv", 'w') as f:
    f.write("NAME,AGE\n")
    for key in my_dictionary.keys():
        f.write(f"{key},{my_dictionary[key]}\n")
```

```
def my_complex_function_1(grades, file_name):
    try:
        with open(file_name, "w") as file:
            for i in grades:
                if i == grades[-1]:
                    file.write(f"{i}")
                else:
                    file.write(f"{i},")
    except:
        print("Too Bad! Exception happened!")

def my_complex_function_2(file_name):
    try:
        with open(file_name, "r") as file:
            text = file.read()
            print(text)
    except:
        print("Too Bad! Exception happened! Maybe the file does not exist?!")

def my_complex_function_3(file_name):
    try:
        with open(file_name, "r") as file:
            text_lines = file.readlines()
            for line in text_lines:
                print(line)
    except:
        print("Too Bad! Exception happened! Maybe the file does not exist?!")

grades = [1, 2, 3, 4, 10, 12, 15, 16, 3]
my_complex_function(grades, "my_text_file.txt")
my_complex_function_2("my_text_file.txt")
my_complex_function_3("my_text_file.txt")
```

بخش — C جای خالی ها را با کد مناسب پر کنید

سؤال ۱

تابع زیر را کامل کنید تا طول یک لیست را برگرداند:

```
def list_length(items):
    return _____
```

سؤال ۲

حلقه زیر را کامل کنید تا اعداد ۰ تا ۴ را چاپ کند:

```
for i in _____:  
    print(i)
```

سؤال ۳

تعریف دیکشنری زیر را کامل کنید:

```
student = {  
    "name": "Sara",  
    "age": _____  
}
```

بخش — D نوشتن قطعه‌کدهای کوتاه

سؤال ۱

تابعی به نام `is_even(n)` بنویسید که اگر `n` زوج بود مقدار `True` و در غیر این صورت مقدار `False` را برگرداند.

سؤال ۲

یک حلقه `for` بنویسید که تمام عناصر لیست زیر را چاپ کند:

```
numbers = [3, 6, 9, 12]
```

سؤال ۳

کدی به زبان پایتون بنویسید که:

- یک مجموعه (`set`) شامل سه عدد ایجاد کند
 - یک عدد جدید به مجموعه اضافه کند
 - مجموعه را چاپ کند
-

بخش — E حل مسئله

سؤال ۱

تابعی به نام `count_positive(nums)` بنویسید که:

- یک لیست از اعداد دریافت کند
 - تعداد اعدادی که بزرگتر از ۰ هستند را برگرداند
-

سؤال ۲

با توجه به لیست زیر:

```
scores = [45, 78, 88, 60, 32, 95]
```

کدی بنویسید که:

1. تعداد دانش‌آموزانی که قبول شده‌اند (نمره ≥ 60) را بشمارد
2. کمترین نمره را پیدا کند

همه اینها + تمرینات آخر هر هفت در اسلایدها