

Putting it together

Python Reference

End-to-end mini-projects

A mini-project combines earlier ideas: data in a list, a function with selection inside a loop, and printed output. This is also the shape of the AP Create Performance Task.

Project: average mark

```
def average(marks):  
    if len(marks) == 0:  
        return 0  
    return round(sum(marks) / len(marks), 1)  
  
print(average([88, 71, 95, 60]))    # 78.5
```

Project: count passes

```
def count_passes(marks, pass_mark=60):  
    passes = 0  
    for m in marks:                # iteration  
        if m >= pass_mark:         # selection  
            passes += 1  
    return passes  
  
print(count_passes([88, 50, 95, 60]))    # 3
```

Project: filter to a new list

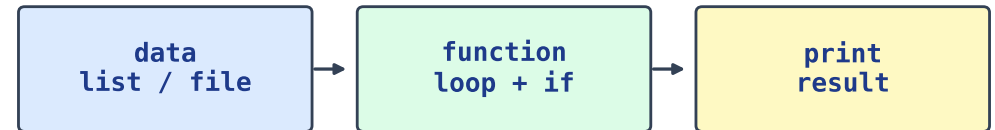
```
def merit(marks):  
    return [m for m in marks if m >= 80]  
  
print(merit([88, 71, 95, 60]))    # [88, 95]
```

The AP Create Task wants a list, a parameterised procedure that uses selection and iteration, and some input/output. Each project above is exactly that shape — build small pieces, then join them.

Common mistakes

- Build in small steps and test each part before moving on — do not write it all at once.
- Read the whole task first, then plan the input → process → output before you code.

Mini-project pipeline



Exam tip: combine list + function + selection + loop + clear output — the Create task shape

Mini-project: data → function → printed result