Errata for
Python and Algorithmic Thinking
for the Complete Beginner

1.2 What is Hardware?

- The Central Processing Unit (CPU)
  This is the part of a computer that actually performs all the tasks defined in a program (basic arithmetic, logical, and input/output operations.)

5.10 Review Exercises

2. Match each element from the first column with one element from the second column.

<table>
<thead>
<tr>
<th>Value</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The name of a person</td>
<td>a. Boolean</td>
</tr>
<tr>
<td>2. The age of a person</td>
<td>b. Real</td>
</tr>
<tr>
<td>3. The result of the division 5/2</td>
<td>c. Integer</td>
</tr>
<tr>
<td>4. Is it black or is it white?</td>
<td>d. String</td>
</tr>
</tbody>
</table>

13.1 Introduction

**Exercise 13.1-2 Finding the Sum of Digits**

**First Approach**

<table>
<thead>
<tr>
<th>Step</th>
<th>Statement</th>
<th>Notes</th>
<th>digit1</th>
<th>digit2</th>
<th>digit3</th>
<th>digit4</th>
<th>r</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>total = digit1 + digit2 + digit3 + digit4</td>
<td>6753</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>3</td>
<td>53</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>print(total)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Second Approach**

Once more, let’s try to understand the second approach using an arithmetic example. Take the same number, 6753, for example.

- **Fourth digit = 3**
  The fourth digit can be isolated if you divide the given number by 10 to get the integer remainder
  
  \[ \text{digit4} = 6753 \mod 10 \]

- **Remaining digits = 675**
  The remaining digits can be isolated if you divide the given number by 10 to get the integer quotient
  
  \[ r = 6753 \div 10 \]

Value 21 is displayed
### 13.2 Review Exercises

5. Write a Python program that prompts the user to enter an integer representing an elapsed time in seconds and then displays it in the format "WW weeks DD days HH hours MM minutes and SS seconds." For example, if the user enters the number 2000000, the message "3 weeks 2 days 3 hours 33 minutes and 20 seconds" should be displayed.

### 14.3 Retrieving a Portion from a String

**Exercise 14.3-1 Displaying a String Backwards**

**Third Approach**

<table>
<thead>
<tr>
<th>file_14_3_1c</th>
</tr>
</thead>
<tbody>
<tr>
<td>s = input(&quot;Enter a word with four letters: &quot;)</td>
</tr>
<tr>
<td>s_rev = s[::-1]</td>
</tr>
<tr>
<td>print(s_rev)</td>
</tr>
</tbody>
</table>

**15.5 Python’s Membership Operators**

- s not in ["a", "b"]. This can be read as ‘test if s is not equal to letter “a”, nor equal to letter “b”.

### 22.5 Exercises of a General Nature with Decision Control Structures

**Exercise 22.5-3 Is the Number a Palindrome?**

**Second Approach**

<table>
<thead>
<tr>
<th>file_22_5_3b</th>
</tr>
</thead>
<tbody>
<tr>
<td>import re</td>
</tr>
<tr>
<td>IS_NUMERIC = &quot;^[+-]?\d+.\d+</td>
</tr>
<tr>
<td>inp = input()</td>
</tr>
<tr>
<td>if not re.match(IS_NUMERIC, inp):</td>
</tr>
<tr>
<td>print(&quot;You entered non-numeric characters&quot;)</td>
</tr>
<tr>
<td>else:</td>
</tr>
<tr>
<td>x = int(inp)</td>
</tr>
<tr>
<td>if x &lt; 10000:</td>
</tr>
<tr>
<td>print(&quot;You entered less than five digits&quot;)</td>
</tr>
</tbody>
</table>
elif x > 99999:
    print("You entered more than five digits")
else:
    inp_reversed = inp[::-1]
    if inp == inp_reversed:
        print("Palindrome")
    else:
        print("Not Palindrome")

24.1 The Pre-Test Loop Structure

Exercise 24.1-3 Designing the Flowchart and Counting the Total Number of Iterations

Now, let's create a trace table to observe the flow of execution.

<table>
<thead>
<tr>
<th>Step</th>
<th>Statement</th>
<th>Notes</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>i = 1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>while i != 6:</td>
<td>This evaluates to True</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>i += 2</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>while i != 6:</td>
<td>This evaluates to True</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>i += 2</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>while i != 6:</td>
<td>This evaluates to True</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>i += 2</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>while i != 6:</td>
<td>This evaluates to True</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

1st Iteration

2nd Iteration

3rd Iteration

Exercise 24.1-5 Finding the Sum of 4 Numbers

Using a pre-test loop structure, write a Python program that lets the user enter 4 numbers and then calculates and displays their sum.

24.6 Review Exercises

3. Design the corresponding flowchart and create a trace table to determine the values of the variables in each step of the next Python program. How many iterations does this Python program perform?

```python
i = -5
while i < 10:
    i = i + 1
print(i)
```

27.8 Converting from a While-Loop to a For-loop

Exercise 27.8-3 Converting the Python Program

```python
file_27_8_3b
s = 0
for i in range(1, 6):
    s = s + (i + 1) ** 2
```
29.6 Exercises of a General Nature with Loop Control Structures

**Exercise 29.6-4 Is the Message a Palindrome?**

```python
message = input("Enter a message: ").lower()
#Create a new string which contains letters except spaces, commas, periods or question marks
message_clean = message
for c in " ,.?":
    message_clean = message_clean.replace(c, "")
if message_clean == message_clean[::-1]:
    print(message_clean, "The message is palindrome")
```

35.2 Data Validation with Lists

**Exercise 35.2-3 Displaying Odds in Reverse Order – Validation with Individual Error Messages**

```python
import re
IS_NUMERIC = "^\[-+]?\d+\.\d+\$"
ELEMENTS = 20
```

35.4 Sorting Lists

**Exercise 35.4-1 The Bubble Sort Algorithm – Sorting One-Dimensional Lists with Numeric Values**

Fifth pass

1st Compare

Elements at index positions 4 and 5 are compared. Since the value 49 is not less than the value 25, no swapping is done.

38.3 Formal and Actual Arguments

**Remember!** There is a one-to-one match between the formal and the actual arguments. The value of argument `a` is passed to argument `n1`, the value of argument `b` is passed to argument `n2`, and so on.

41.1 Simple Exercises with Subprograms

**Exercise 41.1-5 How Many Times Does Each Number of the Dice Appear?**

```python
#Variable n1 is assigned the number of times that value 1 exists in list a
n1 = search_and_count(1, a)

#Variable n2 is assigned the number of times that value 2 exists in list a
n2 = search_and_count(2, a)
```
Variable n6 is assigned the number of times that value 6 exists in list a

n6 = search_and_count(6, a)

Display how many times each of the six numbers appears in list a

print(n1, n2, n3, n4, n5, n6)

Find maximum of n1, n2,... n6

maximum = n1
max_i = 1

if n2 > maximum:
    maximum = n2
    max_i = 3

if n3 > maximum:
    maximum = n3
    max_i = 3

41.2 Exercises of a General Nature with Subprograms

Exercise 41.2-3 Progressive Rates and Electricity Consumption

```python
file_41_2_3

def find_amount(kwh):
    if kwh <= 400:
        amount = kwh * 0.08
    elif kwh <= 1500:
        amount = 400 * 0.08 + (kwh - 400) * 0.22
    elif kwh <= 2000:
        amount = 400 * 0.08 + 1100 * 0.22 + (kwh - 1500) * 0.35
    else:
        amount = 400 * 0.08 + 1100 * 0.22 + 1500 * 0.35 + (kwh - 3000) * 0.5

    amount += 0.26 * amount
    return amount

...