

List numbers one by one

```
20 # Get input from the user
21 number = int(input("Please enter a number 7 to 18 digits: "))
22 # Determine the number of digits in the integer
23 num_digits = len(str(number))
24 # Ensure the number is between 7 to 18 digits
25 if 7 <= num_digits <= 18:
26     # Loop to print out each digit
27     for i in range(num_digits):
28         # Print the digit
29         print(num_digits - num_digits, "18 ** num_digits", 10 ** num_digits)
30 # Calculate the power of 10 for the current position
31 power_of_10 = 10 ** (num_digits - 1)
32 # Print the power of 10, the number, the number // power_of_10, number // power_of_10
33 print("power_of_10", power_of_10, "number", number, "number // power_of_10", number // power_of_10)
34 # Extract the leftmost digit
35 digit = number // power_of_10
36 # Print the digit
37 print(digit)
38 # Remove the leftmost digit from the number
39 number %= power_of_10
40 # Decrease the number of digits left to process
41 num_digits -= 1
42 else:
43     print("Number was not 7 to 18 digits.")
```

sentinel loop

```
114 # Initialize total miles and total gallons to zero
115 total_miles = 0
116 total_gallons = 0
117 overall_avg = 0
118 # Start the sentinel-controlled loop, meaning it repeats until it is broken by user input
119 while True:
120     # Get the gallons used from the user, convert to float
121     gallons = float(input("Please enter the gallons used (-1 to stop the loop): "))
122     # Check for the sentinel value and if so escape
123     if gallons == -1:
124         break
125     # Get the miles driven from the user
126     miles = float(input("Please enter the miles driven: "))
127     # Calculate miles per gallon for this tankful
128     mpg = miles / gallons
129     print("The miles/gallon for this tank was {:.format(mpg)}")
130 # Update the total miles and total gallons for avg calculation later
131 total_miles += miles
132 total_gallons += gallons
133 # Calculate the overall average miles per gallon
134 if total_gallons != 0:
135     overall_avg = total_miles / total_gallons
136 print("The overall average miles/gallon was {:.format(overall_avg)}")
```

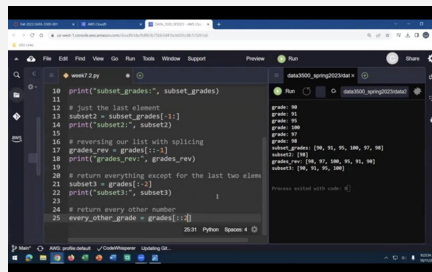
palindrome

```
149 # 3.12 (Palindromes) A palindrome is a number, word or text phrase that reads the same
150 # Get input from the user
151 number = int(input("Please enter a five-digit integer: "))
152 # Ensure the number is five digits
153 if 10000 <= number <= 99999:
154     # Extract the digits
155     digit1 = number // 10000
156     digit2 = (number // 1000) % 10
157     digit3 = (number // 100) % 10
158     digit4 = (number // 10) % 10
159     digit5 = number % 10
160     # Check if the number is a palindrome
161     if digit1 == digit5 and digit2 == digit4:
162         print("{} is a palindrome.".format(number))
163     else:
164         print("{} is not a palindrome.".format(number))
165 else:
166     print("Please enter a five-digit integer.")
```

consecutives in list

```
190 i = 1
191 total = 0
192 consecutive_3_14 = 0
193 consecutive_3_141 = 0
194 iteration_for_3_14 = 0
195 iteration_for_3_141 = 0
196
197 for denominator in range(1, 6000, 2):
198     if i % 2 == 0:
199         total += 4/denominator
200     else:
201         total -= 4/denominator
202
203 # Check for consecutive 3.14 approximations
204 if str(total)[4] == '3.14':
205     consecutive_3_14 += 1
206     if consecutive_3_14 == 2 and not iteration_for_3_14:
207         iteration_for_3_14 = i
208         print('hit 3.14', i)
209     else:
210         consecutive_3_14 = 0
211
212 # Check for consecutive 3.141 approximations
213 if str(total)[5] == '3.141':
214     consecutive_3_141 += 1
215     if consecutive_3_141 == 2 and not iteration_for_3_141:
216         iteration_for_3_141 = i
217         print('hit 3.141', i)
218     else:
219         consecutive_3_141 = 0
220
221 # Print the total, iteration_for_3_14, iteration_for_3_141
222 i += 1
223
```

splicing



moving avg

```
def moving_avg(data, window_size):
    # ...
```

arithmetic

```
def arithmetic(a, b, c):
    # ...
```

pi

```
def pi_approximation(n):
    # ...
```

odd or even

```
def is_odd(n):
    # ...
```

multiples

```
def multiples(n, m):
    # ...
```

formatting

```
def formatted_output(x, y):
    # ...
```

C By chase2981

cheatography.com/chase2981/

Published 10th June, 2025.

Last updated 10th June, 2025.

Page 1 of 1.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>