

# #1

## BAGELS



In Bagels, a deductive logic game, you must guess a secret three-digit number based on clues. The game offers one of the following hints in response to your guess: “Pico” when your guess has a correct digit in the wrong place, “Fermi” when your guess has a correct digit in the correct place, and “Bagels” if your guess has no correct digits. You have 10 tries to guess the secret number.

## The Program in Action

When you run *bagels.py*, the output will look like this:

---

```
Bagels, a deductive logic game.
By Al Sweigart al@inventwithpython.com

I am thinking of a 3-digit number. Try to guess what it is.
Here are some clues:
When I say:      That means:
    Pico          One digit is correct but in the wrong position.
    Fermi         One digit is correct and in the right position.
    Bagels        No digit is correct.
I have thought up a number.
    You have 10 guesses to get it.
Guess #1:
> 123
Pico
Guess #2:
> 456
Bagels
Guess #3:
> 178
Pico Pico
--snip--
Guess #7:
> 791
Fermi Fermi
Guess #8:
> 701
You got it!
Do you want to play again? (yes or no)
> no
Thanks for playing!
```

---

## How It Works

Keep in mind that this program uses not integer values but rather string values that contain numeric digits. For example, '426' is a different value than 426. We need to do this because we are performing string comparisons with the secret number, not math operations. Remember that '0' can be a leading digit: the string '026' is different from '26', but the integer 026 is the same as 26.

- 
1. `"""Bagels, by Al Sweigart al@inventwithpython.com`
  2. `A deductive logic game where you must guess a number based on clues.`
  3. `View this code at https://nostarch.com/big-book-small-python-projects`
  4. `A version of this game is featured in the book "Invent Your Own`
  5. `Computer Games with Python" https://nostarch.com/inventwithpython`
  6. `Tags: short, game, puzzle"""`
  7.
  8. `import random`
  9.

```

10. NUM_DIGITS = 3 # (!) Try setting this to 1 or 10.
11. MAX_GUESSES = 10 # (!) Try setting this to 1 or 100.
12.
13.
14. def main():
15.     print('Bagels, a deductive logic game.
16. By Al Sweigart al@inventwithpython.com
17.
18. I am thinking of a {}-digit number with no repeated digits.
19. Try to guess what it is. Here are some clues:
20. When I say:      That means:
21.   Pico           One digit is correct but in the wrong position.
22.   Fermi          One digit is correct and in the right position.
23.   Bagels         No digit is correct.
24.
25. For example, if the secret number was 248 and your guess was 843, the
26. clues would be Fermi Pico.''.format(NUM_DIGITS))
27.
28.     while True: # Main game loop.
29.         # This stores the secret number the player needs to guess:
30.         secretNum = getSecretNum()
31.         print('I have thought up a number.')
32.         print(' You have {} guesses to get it.'.format(MAX_GUESSES))
33.
34.         numGuesses = 1
35.         while numGuesses <= MAX_GUESSES:
36.             guess = ''
37.             # Keep looping until they enter a valid guess:
38.             while len(guess) != NUM_DIGITS or not guess.isdecimal():
39.                 print('Guess #{}: '.format(numGuesses))
40.                 guess = input('> ')
41.
42.             clues = getClues(guess, secretNum)
43.             print(clues)
44.             numGuesses += 1
45.
46.             if guess == secretNum:
47.                 break # They're correct, so break out of this loop.
48.             if numGuesses > MAX_GUESSES:
49.                 print('You ran out of guesses.')
50.                 print('The answer was {}'.format(secretNum))
51.
52.             # Ask player if they want to play again.
53.             print('Do you want to play again? (yes or no)')
54.             if not input('> ').lower().startswith('y'):
55.                 break
56.         print('Thanks for playing!')
57.
58.
59. def getSecretNum():
60.     """Returns a string made up of NUM_DIGITS unique random digits."""
61.     numbers = list('0123456789') # Create a list of digits 0 to 9.
62.     random.shuffle(numbers) # Shuffle them into random order.
63.
64.     # Get the first NUM_DIGITS digits in the list for the secret number:

```

```

65.     secretNum = ''
66.     for i in range(NUM_DIGITS):
67.         secretNum += str(numbers[i])
68.     return secretNum
69.
70.
71. def getClues(guess, secretNum):
72.     """Returns a string with the pico, fermi, bagels clues for a guess
73.     and secret number pair."""
74.     if guess == secretNum:
75.         return 'You got it!'
76.
77.     clues = []
78.
79.     for i in range(len(guess)):
80.         if guess[i] == secretNum[i]:
81.             # A correct digit is in the correct place.
82.             clues.append('Fermi')
83.         elif guess[i] in secretNum:
84.             # A correct digit is in the incorrect place.
85.             clues.append('Pico')
86.     if len(clues) == 0:
87.         return 'Bagels' # There are no correct digits at all.
88.     else:
89.         # Sort the clues into alphabetical order so their original order
90.         # doesn't give information away.
91.         clues.sort()
92.         # Make a single string from the list of string clues.
93.         return ' '.join(clues)
94.
95.
96. # If the program is run (instead of imported), run the game:
97. if __name__ == '__main__':
98.     main()

```

---

After entering the source code and running it a few times, try making experimental changes to it. The comments marked with (!) have suggestions for small changes you can make. On your own, you can also try to figure out how to do the following:

- Change the number of digits for the secret number by changing `NUM_DIGITS`.
- Change the number of guesses the player gets by changing `MAX_GUESSES`.
- Try to create a version with letters as well as digits in the secret number.

## Exploring the Program

Try to find the answers to the following questions. Experiment with some modifications to the code and rerun the program to see what effect the changes have.

1. What happens when you change the `NUM_DIGITS` constant?

2. What happens when you change the `MAX_GUESSES` constant?
3. What happens if you set `NUM_DIGITS` to a number larger than 10?
4. What happens if you replace `secretNum = getSecretNum()` on line 30 with `secretNum = '123'`?
5. What error message do you get if you delete or comment out `numGuesses = 1` on line 34?
6. What happens if you delete or comment out `random.shuffle(numbers)` on line 62?
7. What happens if you delete or comment out `if guess == secretNum:` on line 74 and return `'You got it!'` on line 75?
8. What happens if you comment out `numGuesses += 1` on line 44?