

Pseudocode

- an informal way of writing algorithms for humans to read (not computers!)
- Illustrates the logic of an algorithm, but omits details that people can fill in automatically.
- You get to make it up as you go along, as long as you (and other people) can easily understand it.

- Instead of saying
 name = input("What is your name? ")
- Pseudocode might use a line that says name = ask user for name

- · Instead of saying
 - if x >= 0 and x <= 100: print("\$%.2f" % x)
- Pseudocode might use a line that says if x is between 0 and 100: print x with 2 decimal places

The point is to get your ideas down on paper quickly, so you can worry about the programming details and exact syntax later. To write any while loop:

- 1. Write out pseudocode for what the loop does, explicitly repeating lines until you've repeated the code at least twice.
- 2. Include an "if" statement in your code that will be True if you want the loop to keep going.
- 3. Make sure the code repeats the "if" statement at least twice.
- Find the statements between consecutive "if" statements. These statements will become the body of the loop.
- 5. The "if" test will become the "while" test.
- 6. If there's anything before the first "if" test, it will go immediately before the while loop (outside of the body).

Pseudocode for Name Comparison Program

main()



Figure 6-3	The random function returns a value
	Some number number = random.randint(1, 100)
	A random number in the range of 1 through 100 will be assigned to the number variable.
Figure 6-4	Displaying a random number
	print(random.randint(1, 10))
	A random number in the range of

<pre># This prog: # in the rax import rand</pre>	am displays age of 1 thro om	a random number ough 10.
<pre>def main(): # Get a number : # Displ print(')</pre>	random numbe random.rand ty the number The number is	er. dint(1, 10) r. s', number)
<pre># Call the n main()</pre>	ain function	n.

Pseudocode for Guess the Number Game

number = generate a random number between 1 and 100 guess = ask user to guess a number between 1 and 100 if number does not equal guess, then keep going if guess is greater than number, tell user this else if guess is less than number, tell user this guess = ask user to guess a number between 0 and 100 if number does not equal guess, then keep going if guess is greater than number, tell user this guess = ask user to guess a number between 0 and 100 if number does not equal guess, then keep going if guess is less than number, tell user this guess = ask user to guess a number between 0 and 100 if number does not equal guess, then keep going if guess is greater than number, tell user this else if guess is less than number, tell user this

When user guesses correct number, tell them so

hem so	
Sample Outpu	ıt
Guess a number between 1	and 100: 50
Your guess was too high.	
Please try again: 40	
Your guess was too high.	
Please try again: 30	
Your guess was too low.	
Please try again: 35	
You guessed right! Great	work!



Examples of loops that count

• See count1.py in Box.com folder

Practice

- Write a while loop that prints all divisors of 30.
 Your code should print out the following: 1, 2, 3, 5, 6, 10, 15, 30
- 2. Modify this loop to print out all divisors of 30 AND 50
- 3. Now let the user select any 2 integers and print out the common divisors of these 2 integers
- 4. Challenge: Print out only the largest of the common divisors of these 2 numbers