



Girls Who Code At Home

Password Generator
Reference Guide

Meteor Catcher Game: Part 4 - Reference Guide



In this document you will find all of the answers to some of the questions in the activity. Follow along with the activity and when you see this icon, stop and check your ideas here.

Step 2: Plan Your Password

Create the Pattern of Your Password Generator (2-3 mins)

Rank <i>Rank the passwords from 1-5. 1: most secure to 5: least secure</i>	Password	# of characters <i>How many characters are in the password?</i>	Character Variation <i>Does the password include upper and/or lower case letters? Numbers? Symbols?</i>	Time for Hacker to Crack <i>Based on this website, how long would it take for a hacker to crack the password?</i>
<i>Example Password</i>	ku8@}:'\$	8	Lowercase letters, symbols, numbers	4 hours
4	hcVESx	6	→ Uppercase letters → Lowercase letters	4 hundred milliseconds
3	vWESp3Tt	8	→ Uppercase letters → Lowercase letters → Numbers	1 hour
1	Sg3Jpezyhv	10	→ Uppercase letters → Lowercase letters → Numbers	7 months
5	password1	9	→ Lowercase letters → Numbers	Instantly
2	jG/8ab{s	8	→ Uppercase letters → Lowercase letters → Numbers → Symbols	12 hours

Step 4: Intro to Python

```
# This is a comment  
# Here we can write anything without the computer reading our messages  
message = "hi"
```

In this example, the first two lines of code are code comments. We can tell because there is a # symbol at the start of the line. You may notice that the color of these lines of code are gray. The editor in Repl.it helps identify code comments by graying these lines of code, if you use a different editor code comments might be a different color. The only line of code that the computer will read is the one written on the third line! We will talk about what is written on the third line later in this activity.

Step 5: Meet Lists in Python

Assigning Values at a List Index

```
programmers = ["Ada", "Grace", "Katherine", "Roya"];  
programmers[2] = "Violet"
```

In our example we stored a **List** of names in a variable called **programmers**. In order to change "Katherine" to "Violet" we tell the computer that at index 2 of the **programmers** List, reassign the value to "Violet".

Step 6: Create All Possible Characters

Create List variables to store types of character possibilities

```
# Lists for possible types of characters
letters =
['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q',
'r','s','t','u','v','w','x','y','z','A','B','C','D','E','F','G','H','I',
'J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z']
sChars = ['!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '+']
numbers = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
```

Test Your Code

Add these optional `print()` statements to check that your Lists have been implemented correctly.

```
# Lists for possible types of characters
letters =
['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r',
's','t','u','v','w','x','y','z',
'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S',
'T','U','V','W','X','Y','Z']
numbers = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
sChars = ['!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '+']

# Print out Lists to check that content is correct
print(letters)
print(numbers)
print(sChars)
```

Step 8: Generate the 1st random character

In this example we implement a password layout where the first character is a letter.

```
# Import the random library
import random

# Lists for possible types of characters
letters =
['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p',
'q','r','s','t','u','v','w','x','y','z',
'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P',
'Q','R','S','T','U','V','W','X','Y','Z']
numbers = ['0','1','2','3','4','5','6','7','8','9']
sChars = ['!','@','#','$','%','^','&','*','(',')','-', '+']

# Print out Lists to check that content is correct
#print(letters)
#print(numbers)
#print(sChars)

# Create password List
pw = [0,0,0,0,0,0,0,0,0,0,0]

pw[0] = random.choice(letters)

print(pw)
```

Step 9: Create Your Random Password

In this example we implement a password layout where the first character is a letter.

```
# Import the random library
import random

# Lists for possible types of characters
letters =
['a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p',
'q','r','s','t','u','v','w','x','y','z',
'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P',
'Q','R','S','T','U','V','W','X','Y','Z']
numbers = ['0','1','2','3','4','5','6','7','8','9']
sChars = ['!','@','#','$','%','^','&','*','(',')','-', '+']

# Create password List
pw = [0,0,0,0,0,0,0,0,0,0]

pw[0] = random.choice(letters)
pw[1] = random.choice(letters)
pw[2] = random.choice(letters)
pw[3] = random.choice(letters)
pw[4] = random.choice(letters)
pw[5] = random.choice(letters)
pw[6] = random.choice(numbers)
pw[7] = random.choice(sChars)
pw[8] = random.choice(letters)
pw[9] = random.choice(letters)

print(pw)
```