



Girls Who Code en casa

Generador de contraseñas

Guía de referencia

Juego del capturador de meteoritos, parte 4: Guía de referencia



En este documento encontrará todas las respuestas a algunas de las preguntas de la actividad. Siga la actividad y cuando vea este ícono, deténgase y revise sus conocimientos aquí.

Paso 2: Planificar su contraseña

Crear el patrón de su generador de contraseñas (de 2 a 3 minutos)

Clasificar <i>Clasifique las siguientes contraseñas del 1 al 5. 1: más segura; 5: menos segura</i>	Contraseña	Cantidad de caracteres <i>¿Cuántos caracteres tiene la contraseña?</i>	Variación de caracteres <i>¿La contraseña incluye letras mayúsculas o letras minúsculas? ¿Números? ¿Símbolos?</i>	Horas para que el hacker descifre la contraseña <i>Según _____, ¿cuánto tiempo tardaría un hacker en descifrar la contraseña?</i>
Contraseña de ejemplo	ku8@}:'\$	8	Letras minúsculas, símbolos y números	4 horas
4	hcVESx	6	→ Letras mayúsculas → Letras minúsculas	400 milisegundos
3	vWESp3Tt	8	→ Letras mayúsculas → Letras minúsculas → Números	1 hora
1	Sg3Jpezyhv	10	→ Letras mayúsculas → Letras minúsculas → Números	7 meses
5	password1	9	→ Letras minúsculas → Números	Al instante
2	jG/8ab{s	8	→ Letras mayúsculas → Letras minúsculas → Números → Símbolos	12 horas

Paso 4: Introducción a Python

```
# Este es un comentario.  
# Aquí podemos escribir cualquier cosa sin que la computadora lea nuestros  
mensajes.  
message = "hi"
```

En este ejemplo, las dos primeras líneas de código son comentarios de código. Lo sabemos porque hay un símbolo `#` al principio de la línea. Puede ver que el color de estas líneas de código es gris. El editor de Repl.it ayuda a identificar los comentarios del código poniendo en gris estas líneas de código; si usted usa un editor diferente, los comentarios del código podrían ser de otro color. La única línea de código que la computadora leerá es la escrita en la tercera línea. Hablaremos sobre lo que está escrito en la tercera línea más adelante en esta actividad.

Paso 5: Conocer las listas de Python

Asignación de valores en el índice de una lista

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

En nuestro ejemplo, almacenamos una **lista** de nombres en una variable llamada **programmers**. Para modificar `"Katherine"` por `"Violet"`, le decimos a la computadora que en el índice 2 de la lista **programmers**, reasigne el valor a `"Violet"`.

Paso 6: Crear todos los caracteres posibles

Crear las variables de la lista para almacenar las posibilidades de tipos de caracteres

```
# Listas de posibles tipos de caracteres
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']
```

Probar su código

Añada estas sentencias condicionales `print()` opcionales para comprobar si las listas se implementaron correctamente.

```
# Listas de posibles tipos de caracteres
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 = ['!','@','#','$','%','^','&','*','(',')','-', '+']

# Imprimir las listas para verificar que el contenido sea el correcto
print(letters)
print(numbers)
print(sChars)
```

Paso 8: Generar el primer carácter aleatorio

En este ejemplo, implementamos un diseño de contraseña en el que el primer carácter es una letra.

```
# Importar la biblioteca aleatoria
import random

# Listas de posibles tipos de caracteres
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 = ['!','@','#','$','%','^','&','*','(',')','-', '+']

# Imprimir las listas para verificar que el contenido sea
el correcto
#print(letters)
#print(numbers)
#print(sChars)

# Crear lista de contraseñas
pw = [0,0,0,0,0,0,0,0,0,0]

pw[0] = random.choice(letters)

print(pw)
```

Paso 9: Crear su contraseña aleatoria

En este ejemplo, implementamos un diseño de contraseña en el que el primer carácter es una letra.

```
# Importar la biblioteca aleatoria
import random

# Listas de posibles tipos de caracteres
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 = ['!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '+']

# Crear lista de contraseñas
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)
```