



# Girls Who Code At Home

पासवर्ड जनरेटर

## गतिविधि अवलोकन

हम डिजिटल रूप से किए गए लगभग हर काम के लिए पासवर्डों का उपयोग करते हैं। आपके पास अपने ईमेल, शॉपिंग वेबसाइटों, इंस्टाग्राम, फूड डिलीवरी ऐप्स आदि के लिए पासवर्ड होंगे। क्या आप जानते हैं कि आपके नाम, जन्मतिथि, या स्थान जैसी व्यक्तिगत जानकारी को सुरक्षित रखने के लिए, आपको अपने सभी खातों के लिए एक अलग पासवर्ड बनाना चाहिए? 2019 में, 15 में से लगभग 1 व्यक्ति [पहचान की धोखाधड़ी](#) का शिकार हुआ था जहाँ हमले के सबसे आम कारणों में से एक था कमजोर पासवर्ड।

साइबर सुरक्षा विशेषज्ञ किसी भी कंप्यूटर में सुरक्षा उपायों को लागू करने और संभावित खतरों को पहचानने के लिए जिम्मेदार हैं। [साइबर सुरक्षा](#) एक तेजी से बढ़ता उद्योग है जो सभी अमेरिकी तकनीक नौकरियों का लगभग 32-45% है जिस में **\$83,000** का औसत मूल वेतन होता है। घर में अपने सिस्टमों को साइबर हमले से बचाने के लिए कई तरीके हैं! इस गतिविधि में, हम 10 वर्णों के ऐसे पासवर्ड रैंडम रूप से जनरेट करेंगे जो इतने मजबूत हैं कि वे आपकी व्यक्तिगत जानकारी को साइबर हमलों से बचा सकते हैं।

यदि आप साइबर सुरक्षा के बारे में अधिक जानना चाहते हैं,  
तो हमारी यह गतिविधि देखें: [साइबर जासूस](#)

## सीखने के लक्ष्य

इस गतिविधि को पूरा कर लेने पर आप...

- व्यक्तिगत जानकारी को सुरक्षित रखने के लिए मजबूत पासवर्ड बनाने के महत्व को समझ पाएंगे.
- जानकारी स्टोर करने के लिए पायथन में [सूचियाँ](#) बनाना और उपयोग कर पाएंगे.
- किसी विशिष्ट रेंज में संख्याओं को यादृच्छिक करने के लिए [रैंडम लाइब्रेरी](#) का उपयोग कर पाएंगे.

## सामग्रियां

- [Repl.it एडिटर](#)
- [पासवर्ड जनरेटर नमूना प्रोजेक्ट](#)
- [पासवर्ड जनरेटर संदर्भ मार्गदर्शिका](#)

## पूर्व ज्ञान

इस प्रोजेक्ट से शुरुआत करने से पहले, हमारी सलाह है कि आप:

- अपने शब्दों में यह स्पष्ट कर सकते हों कि चर राशि या [variable](#) क्या होता है और यह बता सकते हों कि किसी प्रोग्राम में उनका उपयोग कैसे किया जा सकता है।
- अपने शब्दों में यह स्पष्ट कर सकते हों कि सशर्त कथन या कंडीशनल स्टेटमेंट ([conditional statement](#)) क्या होता है और यह बता सकते हों कि किसी प्रोग्राम में उनका उपयोग कैसे किया जा सकता है।

यदि आपको Python के बारे में क्विक रिविज़न चाहिए हो तो हमारा सुझाव है कि आप हमारी यह गतिविधि देखें: [क्या मैं आपकी मदद कर सकती हूँ?](#)

## वुमन इन टेक स्पॉटलाइट: जया बालू



**तस्वीर स्रोत:**  
[कोलंबिया पत्रकारिता की समीक्षा](#)

वॉयलेट ब्लू (Violet Blue) हैकिंग व सायबर अपराध के क्षेत्र में कार्यरत एक पत्रकार, लेखक, सलाहकार और शिक्षिका हैं। वॉयलेट ने पाया कि महिलाओं और **LGBTQIA** समुदाय को सायबर अपराधों के कहीं अधिक जोखिम का सामना करना पड़ता है। हैकर आपकी व्यक्तिगत जानकारी का उपयोग कर सकते हैं या उसे बेच सकते हैं जिससे अजनबियों द्वारा आपको **ट्रोल किए जाने**, आपका रूप धरे जाने, या ऑफ़लाइन दुनिया में आपका पीछा किए जाने व आप पर नज़र रखे जाने की संभावना बढ़ जाती है। वॉयलेट अपने ऑनलाइन प्लेटफ़ॉर्म का उपयोग सायबर अपराधों के बारे में जागरूकता फैलाने, लोगों को उनकी जानकारी की सुरक्षा करने के तरीकों के बारे में शिक्षित करने, और कंपनियों एवं व्यक्तियों को सायबर सुरक्षा के उपायों के बारे में सलाह देने के लिए निरंतर करती रहती हैं।

2015 में, वॉयलेट (Violet) ने **[द स्मार्ट गर्ल्स गाइड टू प्राइवेसी \(The Smart Girl's Guide to Privacy\)](#)** नामक एक पुस्तक लिखी। इस पुस्तक ने अपने पाठकों को इस बारे में शिक्षित किया कि कैसे:

- वेबसाइटों से व्यक्तिगत सामग्री हटाएं
- वेबसाइट और ब्राउज़र के निजता नियंत्रणों का उपयोग प्रभावी ढंग से करें
- पहचान चोरी हो जाने के धक्के से उबरें और उसकी रोकथाम करें
- पता करें कि कानून कहां आपको सुरक्षा देता है—और कहां नहीं
- सुरक्षित ऑनलाइन प्रोफ़ाइल्स बनाएं
- लोगों को ढूँढने वाली वेबसाइट्स से खुद को हटाएं

**हैकिंग की संस्कृति पर वॉयलेट के विचारों पर यह वीडियो** देखें (टाइमस्टाम्प 3:20 तक)। वॉयलेट के बारे में अधिक जानना चाहती हैं? साइबर अपराधों से सुरक्षित बनने के बारे में **[साथी महिला पत्रकारों के लिए सुझावों पर यह लेख](#)** पढ़ें। इंटरनेट ब्राउज़ करते समय अपने डेटा को गुप्त बनाए रखने के लिए यहाँ **[पासवर्ड मैनेजरों का उपयोग करने के बारे में अधिक जानें](#)** या **[VPN](#)**, वर्चुअल प्राइवेट नेटवर्क का उपयोग करना सीखें।

## झलक

एक कंप्यूटर वैज्ञानिक होना, कोडिंग में बेहतरीन होने की तुलना में अधिक है। इस बात के बारे में सोचने में थोड़ा समय बिताएं कि कैसे वॉयलेट और उनका काम उन शक्तियों से संबंधित है जिन पर महान कंप्यूटर वैज्ञानिक - बहादुरी, लचीलेपन, रचनात्मकता और उद्देश्य के निर्माण के दौरान ध्यान केंद्रित करते हैं।



### रचनाशीलता

वॉयलेट (Violet) निरंतर अपने ऑनलाइन प्लेटफ़ॉर्म के जरिए सायबर सुरक्षा के बारे में लेख लिख रही हैं और दूसरों को शिक्षित कर रही हैं। यदि आपके पास खुद का कोई प्लेटफ़ॉर्म होता, तो आप दूसरों को किस बारे में शिक्षित करते?

परिवार के किसी सदस्य या मित्र के साथ अपनी प्रतिक्रियाएँ साझा करें। चर्चा में शामिल होने हेतु दूसरों को वॉयलेट के बारे में अधिक पढ़ने के लिए प्रोत्साहित करें!

## चरण 1: साइबर सुरक्षा क्या है? (2-5 मिनट)



**साइबर सुरक्षा** साधारण तौर पर कंप्यूटरों (कोई भी इलेक्ट्रॉनिक डिवाइस जो डेटा को स्टोर और प्रोसेस कर सकती है), सर्वरों, नेटवर्क, और डेटा को साइबर हमलों से बचाने की प्रथा है। लगभग सभी प्रकार की इलेक्ट्रॉनिक डिवाइसों के **वाईफाई** या **5G** के माध्यम से इंटरनेट से जुड़े होने के कारण साइबर हमलों से हमारी जानकारी की रक्षा करना अब पहले से भी अधिक जरूरी है। आप प्रसिद्ध व्यक्तियों की निजता के लीक होने, सरकारों की संवेदनशील जानकारी के जारी होने से हुए कुछ मशहूर हमलों से लेकर टीवी पर क्राइम शोज़ से परिचित होंगे, या आप व्यक्तिगत रूप से ऐसे किसी व्यक्ति को जानते होंगे जिस पर साइबर हमला हुआ है।

2019 में 15 में से लगभग 1 लोग **पहचान की धोखाधड़ी के शिकार** हो गए थे। सबसे आम हमले का कारण किसी ऑनलाइन खाते के लिए मजबूत और सुरक्षित पासवर्ड का नहीं होना पाया गया है। इस गतिविधि में हम एक मजबूत पासवर्ड के गुणों और महत्व पर चर्चा करेंगे, फिर आप सीखेंगे कि पायथन में पासवर्ड जनरेटर को कैसे कोड करते हैं!


### मजबूत/सुरक्षित पासवर्ड (2 मिनट)

यदि किसी हैकर ने आपके खाते को एक्सेस कर लिया है, तो इसका सबसे संभावित कारण एक **कमज़ोर/अरक्षित पासवर्ड** है। पासवर्ड बनाते समय, एक वैध खाता बनाने में सक्षम होने से पहले कुछ सरल आवश्यकताएं हो सकती हैं। कुछ प्रतिबंधों में शामिल हैं:

- कम से कम 6-8 वर्ण
- अपरकेस और लोअरकेस अक्षरों का मिश्रण
- कम से कम एक संख्या का उपयोग
- कम से कम एक विशेष वर्ण

दायीं ओर दिए चार्ट को देखें और तुलना करें कि हैकरों को आपका पासवर्ड समझने में कितना समय लगता है।

TIME IT TAKES FOR A HACKER TO CRACK YOUR PASSWORD					
Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	1 sec	5 secs
7	Instantly	Instantly	25 secs	1 min	6 mins
8	Instantly	5 secs	22 mins	1 hour	8 hours
9	Instantly	2 mins	19 hours	3 days	3 weeks
10	Instantly	58 mins	1 month	7 months	5 years
11	2 secs	1 day	5 years	41 years	400 years
12	25 secs	3 weeks	300 years	2k years	34k years
13	4 mins	1 year	16k years	100k years	2m years
14	41 mins	51 years	800k years	9m years	200m years
15	6 hours	1k years	43m years	600m years	15 bn years
16	2 days	34k years	2bn years	37bn years	1tn years
17	4 weeks	800k years	100bn years	2tn years	93tn years
18	9 months	23m years	6tn years	100 tn years	7qd years

 **Cybersecurity that's approachable.**  
Find out more at [hivesystems.io](https://hivesystems.io)

तस्वीर स्रोत: [Reddit](#)

अक्सर लोग फेसबुक, एप्पल से लेकर बैंक खातों तक, अपने सभी ऑनलाइन खातों के लिए एक ही पासवर्ड का उपयोग करते हैं। हालांकि इससे लोगों के लिए अपने पासवर्डों को याद रखना आसान हो जाता है, वे साइबर हमलों के प्रति और भी अरक्षित हो जाते हैं।

आपके खातों को सुरक्षित करने का सबसे अच्छा तरीका है विविध प्रकार के अक्षरों, संख्याओं, और विशेष वर्णों वाले अद्वितीय पासवर्ड बनाना। लेकिन जैसे कि आप कल्पना कर सकते हैं, उन्हें याद रखना बहुत ही मुश्किल होता है। पेश हैं पासवर्ड मैनेजर! ये सुरक्षित सॉफ्टवेयर सभी खातों के लिए मजबूत पासवर्ड बनाने और याद रखने में लोगों की मदद करते हैं, आपको बस एक पासवर्ड याद रखना है।

## चरण 2: अपने पासवर्ड की योजना बनाएं (15-20 मिनट)

इसका अन्वेषण करें! (10-15 मिनट)



इस प्रोजेक्ट के लिए हम [Repl.it](https://repl.it) वेब एडिटर का उपयोग करेंगे। Repl.it एक मुफ्त, सहयोगात्मक, ब्राउज़र पर आधारित एडिटर है जो अनेक प्रोग्रामिंग भाषाओं का समर्थन करता है। यह शक्तिशाली टूल आपको कोड करने और उसी समय अपने कई मित्रों के साथ बात करने की अनुमति तक दे सकता है!

- ❑ निम्नलिखित पासवर्डों को 1-5 तक रैंक करें, जहाँ 1 का मतलब सबसे अधिक सुरक्षित और 5 का मतलब सबसे कम सुरक्षित पासवर्ड है। पासवर्डों को रैंक करने में आपकी मदद करने के लिए, हमने कुछ कॉलम शामिल किए हैं: वर्णों की संख्या और वर्ण में भिन्नता।
  - ❑ प्रत्येक पासवर्ड के लिए वर्णों की संख्या और वर्ण की भिन्नता कॉलम भरें।
  - ❑ पासवर्डों को 1-5 तक रैंक करें।
  - ❑ [इस वेबसाइट](#) का उपयोग करके अपने उत्तर जाँचें और प्रत्येक विकल्प टाइप करें। अपनी रैंकिंग की तुलना करने के लिए अंतिम कॉलम, हैकर को क्रैक करने में लगने वाला समय, भरें।

रैंक करें 1: सबसे सुरक्षित से लेकर 5: सबसे कम सुरक्षित तक	पासवर्ड	वर्णों की संख्या पासवर्ड में कितने वर्ण हैं?	वर्णों की भिन्नता क्या पासवर्ड में अपर और/या लोअर केस अक्षर शामिल हैं? संख्याएं? प्रतीक?	हैकर के द्वारा खोलने में लगने वाला समय इस वेबसाइट, किसी हैकर को पासवर्ड को खोलने में कितना समय लगेगा?
उदाहरण पासवर्ड	ku8@}:'\$	8	लोअरकेस अक्षर, प्रतीक, संख्याएं	4 घंटे
	hcVESx			
	vWESp3Tt			
	Sg3Jpezyhv			
	पासवर्ड1			
	jG/8ab{s			



अपने विचारों को पृष्ठ 2 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।

## चरण 2: अपने पासवर्ड को नियोजित करें (जारी)

- अपने खुद के सुरक्षित पासवर्ड बनाने में **2 मिनट** लगाएं और उन्हें नीचे दी गई जगह में लिखें। आपके पासवर्डों को खोलने में हैकरों को *कम से कम 1 वर्ष* लगना चाहिए।
- पासवर्ड कितने सुरक्षित हैं यह जाँचने के लिए पिछले चरण में हमने जिसका उपयोग किया था [उस वेबसाइट](#) तक नेविगेट करें।

### अपने पासवर्ड जनरेटर का पैटर्न बनाएं (2-3 मिनट)

इस गतिविधि में आप एक प्रोजेक्ट बनाएंगे जो एक 10 वर्ण का पासवर्ड रैंडम रूप से जनरेट करता है। अपने पासवर्ड में *वर्णों का क्रम* आप तय करेंगे।

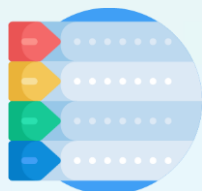
#### उदाहरण

उदाहरण के लिए इस नमूना प्रोजेक्ट में, जनरेट किए गए सभी पासवर्ड निम्नलिखित पैटर्न से युक्त होंगे:

- **0-5वां वर्ण:** अपर या लोअर केस वाला एक रैंडम अक्षर
- **6वाँ वर्ण:** एक रैंडम संख्या
- **7वाँ वर्ण:** एक रैंडम विशेष वर्ण
- **8वाँ-9वाँ वर्ण:** अपर या लोअरकेस वाला एक रैंडम अक्षर

q1Qe1B7-En  
ihhniK2&kR  
iGVRNz3^Hb  
oYJTJC2+s0  
bTedeQ7&Rm

ये एक रैंडम रूप से जनरेट किए गए पासवर्ड के 5 उदाहरण हैं  
जो ऊपर जैसे पैटर्न का ही उपयोग करता है!



आप सोच सकते हैं, हमने 0 से क्यों शुरू किया? वास्तव में, हमारा पासवर्ड बनाने के लिए हमें पायथन में **सूचियाँ** नामक एक वस्तु का उपयोग करने की जरूरत पड़ेगी। **सूची** एक क्रमवार डेटा सेट है जिसका उपयोग विभिन्न प्रकार की जानकारी को स्टोर करने के लिए किया जा सकता है। **सूचियाँ** भीतर मौजूद प्रत्येक तत्व को संदर्भित करने के लिए इंडेक्सों, या एक संख्या का उपयोग करती है। इसके कारण **सूची** के भीतर जानकारी को पुनर्प्राप्त करना, जोड़ना, डिलीट करना, और संशोधित करना आसान हो जाता है। इंडेक्स 0 पर शुरू होते हैं और 1 के द्वारा बढ़ते हैं, जिसके कारण हमने उपरोक्त उदाहरण में पासवर्ड के लिए हमारे वर्ण पैटर्न में शुरू करने के लिए 0 को चुना! हम इस गतिविधि में बाद में इस बारे में अधिक जानेंगे कि **सूचियाँ** डेटा को कैसे स्टोर करती हैं।

## चरण 2: अपने पासवर्ड को नियोजित करें (जारी)

- ❑ नीचे दी गई तालिका में अपने पासवर्ड के पैटर्न की योजना बनाने में 2 मिनट बिताएं। सरलता के लिए, हमारा सुझाव है कि आप अपर और लोअरकेस अक्षरों को एक साथ संयोजित करें। यदि आप अपने पैटर्न को अधिक पेचीदा बनाना चाहते हैं तो आप उन स्थानों को निर्दिष्ट कर सकते हैं जहाँ पर अपर या लोअरकेस अक्षर होंगे।

वर्ण #	वर्ण का प्रकार <i>अक्षर (अपर या लोअरकेस), संख्या, या विशेष वर्ण</i>
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

**ध्यान दें:** आपके पासवर्ड का पैटर्न इस सारी गतिविधि में दिए गए उदाहरण में प्रदर्शित पैटर्न से अलग हो सकता है। जब आप अपने कोड की तुलना संदर्भ मार्गदर्शिका में हमारे उदाहरणों के साथ करते समय यह बात ध्यान में रखें।

**सुझाव:** इस पेज को प्रिंट कर लें ताकि आप बाद में इसका आसानी से संदर्भ ले सकें!



## चरण 3: Repl.it के साथ शुरू करें (15 - 20 मिनट)



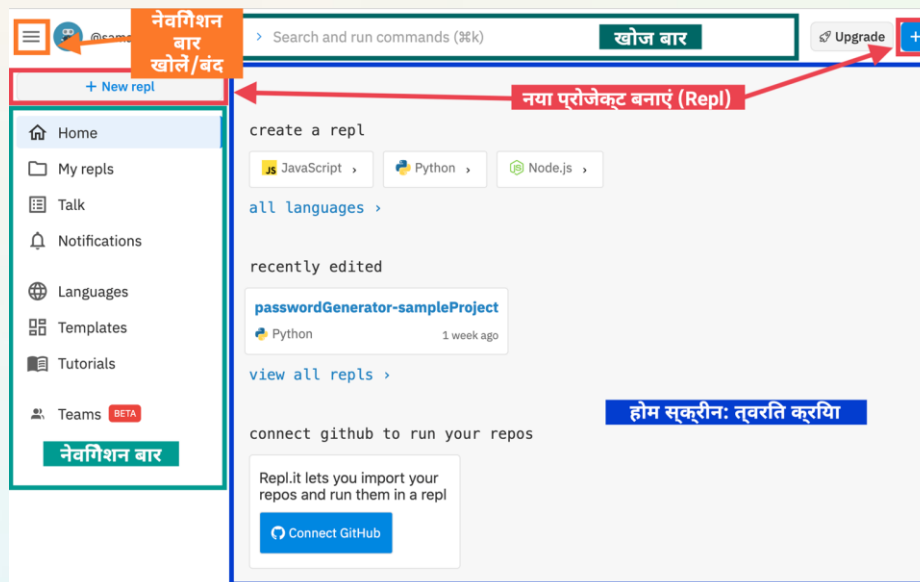
इस प्रोजेक्ट के लिए हम [Repl.it](https://repl.it) वेब एडिटर का उपयोग करेंगे। Repl.it एक मुफ्त, सहयोगात्मक, ब्राउज़र पर आधारित एडिटर है जो अनेक प्रोग्रामिंग भाषाओं का समर्थन करता है। यह शक्तिशाली टूल आपको कोड करने और उसी समय अपने कई मित्रों के साथ बात करने की अनुमति तक दे सकता है!

### एक खाता बनाएं (3-5 मिनट)

- ❑ **Repl.it में साइन अप या लॉगिन करें।** अपने काम को सहेजने के लिए आपको एक खाता बनाने की जरूरत पड़ेगी। खाता बनाने के लिए साइन अप फॉर्म पर दिए निर्देशों का पालन करें। यदि आप 13 वर्ष से कम उम्र की हैं, तो आपको साइन अप करने के लिए अपने माता/पिता के ईमेल पते की आवश्यकता होगी।
- ❑ **खाता बनाने के लिए निर्देशों का पालन करें।** अधिक तेज लॉगिन के लिए आप अपने Google, GitHub, या Facebook खाते के साथ साइन अप करने का चुनाव कर सकते हैं।

### Repl.it मंच का अन्वेषण करें (3-5 मिनट)

Repl.it मंच में स्वागत है! चलिए शुरू करने के लिए उपलब्ध कुछ मुख्य विशेषताओं का अन्वेषण करते हैं।



आपको अपने सारे Repl.it ब्राउज़र में GitHub का कई बार उल्लेख होता दिख सकता है। **GitHub** एक लोकप्रिय सॉफ्टवेयर डेवलपमेंट मैनेजर है। इस गतिविधि में हम इस टूल का उपयोग नहीं करेंगे, लेकिन अधिक जानने के लिए इस [GitHub ट्यूटोरियल](https://github.com) देखें।

- **नेविगेशन बार:** आपकी स्क्रीन के बायीं ओर स्थित यह कॉलम आपको उन आम कार्रवाईयों को एक्सेस करने देता है जिन्हें आप करना चाह सकते हैं, जैसे अपने Repls. बनाना या देखना।
- **खोजें बार:** नेविगेशन बार का उपयोग करने की बजाय, आप कमांड खोजने और चलाने के लिए खोजें बार का उपयोग भी कर सकते हैं।
- **New Repl:** नया प्रॉजेक्ट बनाने के दो तरीके हैं। पहला तरीका नेविगेशन बार के माध्यम से जाकर **+New repl** बटन पर क्लिक करना या विंडो के शीर्ष दाहिने ओर स्थित **+** बटन को क्लिक करना है।
- **मुख्य क्षेत्र:** आपके स्क्रीन के केंद्र में आपके व्यू के अनुसार भिन्न होगा। जब आप Repl.it में पहली बार लॉग इन करते हैं तो आपको डिफाल्ट रूप से होम स्क्रीन पर ले जाया जाएगा।

आप इस [संसाधन](#) का उपयोग करके Repl.it मंच और Repls के बारे में अधिक जान सकते हैं।

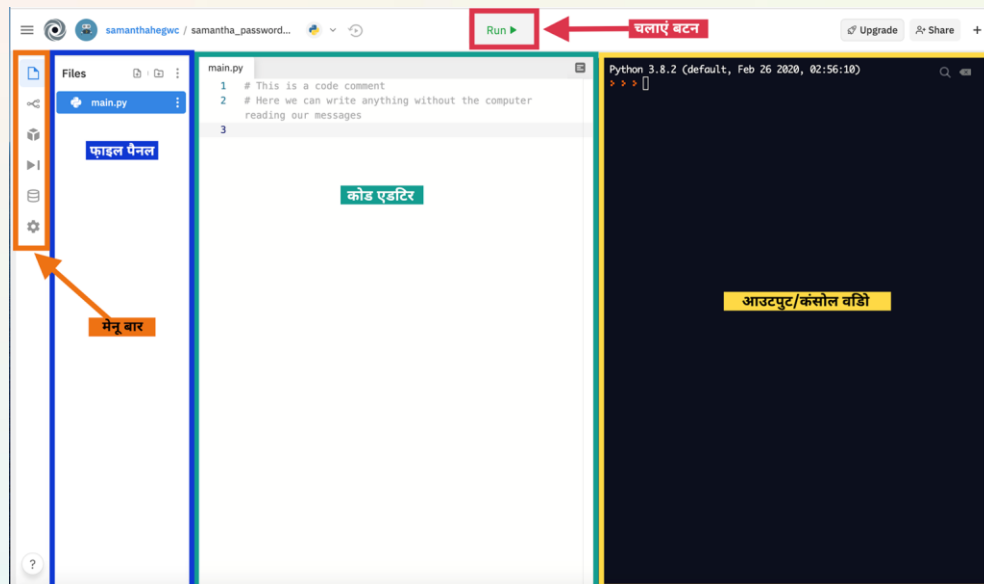


### New Repl.it प्रॉजेक्ट बनाएं (2-4 मिनट)

- ❑ **नया repl. बनाएं** अपने स्क्रीन के शीर्ष बायें कोने पर नीले + पर या अपने नेविगेशन बार में +New repl बटन पर क्लिक करें।
- ❑ **भाषा विकल्प के अंतर्गत पायथन चुनें।** इस प्रॉजेक्ट के लिए हम पायथन में काम करेंगे, लेकिन याद रखें कि भविष्य के किसी भी प्रॉजेक्ट के लिए आप Repl.it पर 50 से अधिक भाषाओं में प्रोग्राम कर सकते हैं।
- ❑ **अपने प्रॉजेक्ट का नामकरण करें।** अपने प्रॉजेक्ट को <yourName>\_pwGenerator जैसा कोई वर्णनात्मक नाम दें। आम तौर पर प्रॉजेक्ट्स में कोई स्पेस नहीं होने चाहिए और शब्दों को अलग करने के लिए **camelCase** या अंडरस्कोरों का उपयोग करें।
- ❑ **Create Repl पर क्लिक करें।**

### एडिटर व्यू का अन्वेषण करें (5-8 मिनट)

चलिए Repl.it के लिए एडिटर व्यू को देखते हैं। अब जबकि हमने एक नया प्रोजेक्ट बना लिया है, हमें यह समझने की जरूरत है कि कहाँ कोड करना है, फाइलों और संपत्तियों के बीच कैसे नेविगेट करना है और आपके कोड को कैसे सहेजना और चलाना है!



- **फाइल्स पेन:** डिफाल्ट रूप में, यह क्षेत्र आपको प्रोजेक्ट से संबद्ध सभी फाइलें दिखाएगा। इस गतिविधि में हम केवल एक फाइल, main.py में काम करेंगे। आप अन्य प्रोजेक्ट में शामिल करने के लिए अपनी इच्छानुसार डेटा फाइलें, छवियाँ, या अन्य पायथन फाइलें पा सकते हैं। इस पेन को बंद करने के लिए, फाइल्स आइकन पर क्लिक करें जो मेनू बार के शीर्ष पर स्थित है।
- **मेनू बार:** यह बार आपको फाइल पेन में व्यू को बदलने की अनुमति देगा। कुछ विकल्पों में संस्करण नियंत्रण बदलना, पैकेज जोड़ना, डीबगिंग, डेटाबेस जोड़ना और सेटिंग्स अपडेट करना शामिल है। सेटिंग्स में आप लेआउट, थीम, फॉन्ट का आकार, और अन्य टेक्स्ट सेटिंग्स बदल सकते हैं।
- **कोड एडिटर:** यह वह जगह है जहाँ आप अपना कोड लिखते हैं!
- **चलाएं बटन:** अपने कोड में परिवर्तन करने के बाद, एडिटर के शीर्ष पर चलाएं बटन पर क्लिक करें। आपको दायीं ओर स्थित आउटपुट/कंसोल विंडो में अपना परिणाम दिखना चाहिए।
- **आउटपुट/कंसोल विंडो:** यह आपके कोड में किसी भी आउटपुट को प्रदर्शित करती है। सभी आउटपुट दिखाई देंगे, इसलिए यदि आप चलाएं बटन पर कई बार क्लिक करते हैं, तो यहाँ प्रत्येक परिणाम दिखाया जाएगा।

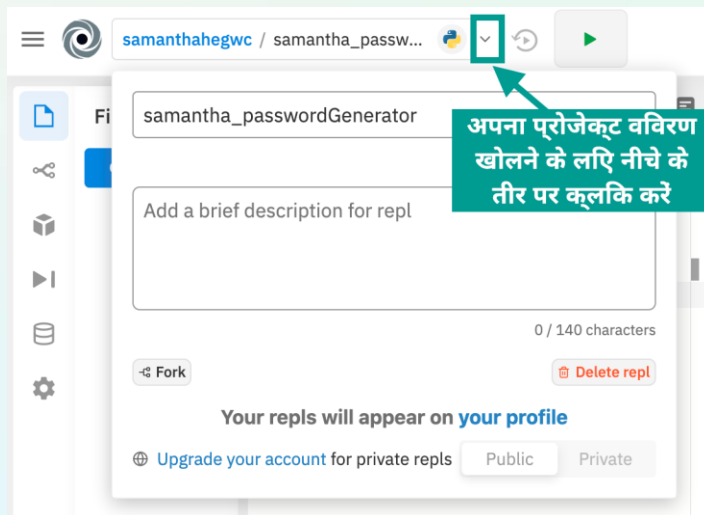
आपने देखा होगा कि Repl.it में कोई **सहेजें** बटन नहीं है। जब तक आपके पास इंटरनेट कनेक्शन होगा, आपके कोड के सभी परिवर्तन स्वचालित रूप से सहेजे जाएंगे! जैसे ही आप चलाएं बटन पर क्लिक करते हैं आपका Repl सहेज लिया जाता है, इसलिए सुनिश्चित करें कि अपने एडिटर को बंद करने से पहले आप हमेशा अपने कोड को चलाएं!



आपके कोड को बैक अप करना भी एक अच्छा विचार है क्योंकि वह आपके कोड में कोई त्रुटि होने पर या अगर Repl.it को बंद करने पर वह ठीक से न सहेजा जाए तो काम आएगा। आप इस [संसाधन](#) के साथ इस बारे में अधिक सीख सकते हैं कि अपने काम को बैक अप करने के लिए GIT भंडार सेट अप करने के लिए मेनू बार में वर्शन कंट्रोल विकल्प का उपयोग कैसे करते हैं। हम इस प्रोजेक्ट के लिए इस विकल्प का उपयोग नहीं करेंगे, लेकिन जब आप Repl.it के साथ अधिक प्रोजेक्ट बनाना जारी रखेंगे तो यह एक मूल्यवान संसाधन होगा!

इससे पहले कि हम अपने पासवर्ड जनरेटर को कोड करना शुरू करें, चलिए हमारे Repl में संक्षिप्त प्रोजेक्ट विवरण जोड़ते हैं।

- ❑ **प्रोजेक्ट का विवरण खोलें।** अपने स्क्रीन के ऊपरी बायें तरफ अपने प्रोजेक्ट का नाम खोजें। नाम के दायीं तरफ स्थित छोटे डाउन तीर पर क्लिक करें। इससे एक नई विंडो खुलनी चाहिए जहाँ आपके प्रोजेक्ट का नाम और संक्षिप्त विवरण के लिए क्षेत्र होगा।
- ❑ **एक संक्षिप्त विवरण जोड़ें।** आप अपने विवरण में निम्नलिखित जानकारी शामिल करना चाहेंगे:
  - ❑ **सिंहावलोकन:** इसे कैसे काम करना चाहिए?
  - ❑ **निर्देश:** क्या आपके प्रोजेक्ट को चलाने के लिए किन्हीं विशिष्ट निर्देशों की जरूरत है?
  - ❑ **विशेषताएं:** क्या आपको दूसरों से या अतिरिक्त संसाधनों से मदद मिली? सुनिश्चित करें कि आप इन लोगों और संसाधनों को आवाज़ दें!



इस अनुभाग को इस समय फाइनल होने की जरूरत नहीं है,  
आप हमेशा वापस जा सकते हैं और अपने विवरण का बाद में संपादन कर सकते हैं।

## चरण 4: Python का परिचय (3-5 मिनट)



**Python** एक टेक्स्ट-आधारित प्रोग्रामिंग भाषा है, यानि हमें सारे कमांड टाइप करने होंगे! कई प्रोग्रामर्स Python का उपयोग इसलिए चुनते हैं क्योंकि इसे सीखना और समझना आसान है क्योंकि यह अंग्रेजी जैसे आदेशों की नकल करता है। Python एक **मुक्त स्रोत (ओपन सोर्स)** भाषा है, यानि यह सभी के द्वारा उपयोग और आवश्यकतानुसार संशोधन के लिए मुक्त रूप से उपलब्ध है। अपडेट्स कैसे स्वीकारी जाती हैं और भाषा पर लागू की जाती हैं इस बारे में कड़े दिशानिर्देश मौजूद हैं, पर मूल रूप से कहें तो कोई भी इस भाषा के विकास में योगदान दे सकता है!

हम सबसे पहले अपने प्रोजेक्ट में कुछ **कोड टिप्पणियाँ जोड़ेंगे**। कोड टिप्पणियाँ कोड की वे पंक्तियाँ हैं जिन्हें कंप्यूटर द्वारा नहीं पढ़ा गया है। प्रोग्रामर इन टिप्पणियों का उपयोग अक्सर अपने प्रोजेक्ट के भीतर नोट्स छोड़ने के लिए करते हैं जो कोड की कुछ पंक्तियों को समझाते हैं और उनके कोड को अधिक पठनीय बनाते हैं। इससे अन्य लोगों के लिए यह समझना आसान हो जाता है कि प्रोग्रामर अपने प्रोजेक्ट से क्या हासिल करने की कोशिश कर रहा था।

पायथन में एक एकल पंक्ति की टिप्पणी जोड़ने के लिए, आप प्रत्येक पंक्ति को **#** प्रतीक से शुरू करते हैं। इससे कंप्यूटर को पता चलता है कि कोड की पंक्ति एक टिप्पणी है और वह **#** के आगे कुछ भी नहीं पढ़ती है। चलिए एक उदाहरण देखते हैं:

```
# यह एक टिप्पणी है
# यहाँ हम कंप्यूटर के द्वारा हमारे संदेशों को पढ़े बिना कुछ भी लिख सकते हैं
संदेश = "हाय"
```

- ❑ ऊपर दी गई कोड की कतरन की एक पल के लिए समीक्षा करें। कोड की कौन सी पंक्तियाँ टिप्पणियाँ हैं? आप कैसे जानते हैं?



अपने विचारों को पृष्ठ 3 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।

**ध्यान दें:** चूंकि Repl.it विवरणों को केवल 140 वर्णों तक ही सीमित करता है, आप अपने प्रोजेक्ट के शीर्ष पर कोड टिप्पणियों का उपयोग अपने प्रोजेक्ट के लिए अधिक विस्तृत विवरण/निर्देश जोड़ने के लिए भी कर सकते हैं।

## चरण 5: पायथन में सूचियों से मिलें (10-15 मिनट)



**सूची** एक क्रमवार डेटा सेट है जिसका उपयोग विभिन्न प्रकार की जानकारी को स्टोर करने के लिए किया जा सकता है। हम अपने प्रोजेक्ट में अपने रैंडम रूप से जनरेट किए गए पासवर्ड के लिए चुनने के लिए सभी संभव वर्णों को स्टोर करने के लिए **सूचियों** का उपयोग करेंगे। **सूचियाँ** भीतर मौजूद प्रत्येक तत्व को संदर्भित करने के लिए इंडेक्सों, या एक संख्या का उपयोग करती हैं। इसके कारण **सूची** के भीतर जानकारी को पुनर्प्राप्त करना, जोड़ना, डिलीट करना, और संशोधित करना आसान हो जाता है। इंडेक्स 0 पर शुरू होते हैं और 1 के द्वारा बढ़ते हैं, जिसके कारण हमने उपरोक्त उदाहरण में पासवर्ड के लिए हमारे वर्ण पैटर्न में शुरू करने के लिए 0 को चुना! चलिए एक उदाहरण देखते हैं:

PYTHON	विवरण
<pre>प्रोग्रामर = ["Ada", "Grace", "Katherine", "Roya"]</pre>	<ul style="list-style-type: none"><li>→ <b>प्रोग्रामर</b>: उस <b>चर राशि</b> का नाम जिसे हम अपनी <b>सूची</b> में स्टोर करते हैं।</li><li>→ <b>=</b>: हम किसी चर राशि पर किसी वस्तु/मान का आबंटन करने के लिए बराबर चिह्न का उपयोग करते हैं।</li><li>→ <b>[]</b>: स्क्वेयर ब्रैकेट्स का उपयोग किसी <b>सूची</b> में सामग्री के आरंभ और समाप्ति को दिखाने के लिए किया जाता है।</li><li>→ <b>"Ada", "Grace", "Katherine", "Roya"</b>: इस <b>सूची</b> में हम प्रोग्रामरों के नाम जोड़ते हैं। हम डबल या सिंगल कोटेशन मार्कों का उपयोग करके <b>स्ट्रिंग</b>, या शब्द, डेटा प्रकार का संकेत देते हैं।</li><li>→ <b>,</b>: हम <b>सूची</b> में स्टोर किए गए प्रत्येक डेटा/मान को अलग करने के लिए अर्धविराम (commas) का उपयोग करते हैं।</li></ul>

इस **सूची** में 0वें इंडेक्स से शुरू करते हुए प्रत्येक प्रोग्रामर के नाम को इंडेक्स आबंटित किए जाते हैं।

0	1	2	3
"Ada"	"Grace"	"Katherine"	"Roya"

हम देख सकते हैं कि Ada **प्रोग्रामर सूची** में 0वें इंडेक्स पर है, Grace **प्रोग्रामर सूची** में इंडेक्स 1 पर है, इत्यादि।

### सूची इंडेक्स को पहचानने का अभ्यास करें (3-5 मिनट)

चलिए यह सुनिश्चित करने में एक पल बिताएं कि आप **सूचियों** की संरचना को समझते हैं और कि एक **सूची** में विशिष्ट जानकारी स्टोर और ट्रैक करने के लिए इंडेक्स का उपयोग कैसे किया जाता है। इस अभ्यास प्रोजेक्ट में हमने सारा कोड लिख दिया है। आप प्रोजेक्ट को चलाएं और आउटपुट/कंसोल विंडो में प्रश्नों का उत्तर देंगे।

इस प्रोजेक्ट में हमारे पास निम्नलिखित डेटा से युक्त एक **सूची** है:

```
wit = ["Ada", "Grace", "Violet", "Ayanna", "Miral", "ENIAC"]
```

आपसे प्रश्नों की एक शृंखला पूछी जाएगी जैसे कि:

#### इंडेक्स 2 पर WIT क्या है?

उत्तर देने के लिए, बस अपना जवाब टाइप करें और कंसोल में एंटर दबाएं। याद रखें कि पायथन केस सेंसिटिव है, इसलिए सुनिश्चित करें कि आपके सभी जवाब अंग्रेजी के बड़े अक्षर से शुरू हों!

- ❑ **इस सूची अभ्यास Repl.it को खोलने के लिए इस [लिंक](#) पर क्लिक करें।** यह सूचियों और इंडेक्स के बारे में आपके ज्ञान की जाँच करने का एक अभ्यास प्रोजेक्ट है।
- ❑ **शीर्ष पर चलाएं बटन पर क्लिक करें।**
- ❑ **कंसोल विंडो में प्रश्नों का उत्तर दें।** यह प्रोग्राम आपसे अनिश्चित काल के लिए प्रश्न पूछता जाएगा। अगले कार्य पर जाने से पहले इनमें से कम से कम 3-4 प्रश्नों का सही उत्तर दें।

### इंडेक्स का उपयोग करके एक सूची में मान आबंटित करना (2-4 मिनट)

हम अपनी इच्छानुसार डेटा के इंडेक्स को संदर्भित करके सूचियों में स्टोर की गई जानकारी को आसानी से एक्सेस कर सकते हैं। चलिए ऐसा करने के लिए सिंटेक्स पर एक नज़र डालते हैं।

PYTHON	विवरण
<code>listName[index] = newValue</code>	<ul style="list-style-type: none"> <li>→ <b>listName</b>: यह आइटमों की <b>सूची</b> को स्टोर करने वाली चर राशि का नाम है।</li> <li>→ <b>[]</b>: हम अपनी <b>सूची</b> में उन चीजों को पहचानने के लिए स्क्वेयर ब्रैकेटों का उपयोग करते हैं जिनकी जानकारी हम एक्सेस करना चाहते हैं।</li> <li>→ <b>इंडेक्स</b>: यह उस डेटा का इंडेक्स, या नंबर है जिसे आप एक्सेस करना चाहते हैं।</li> <li>→ <b>=</b>: हम किसी नए मान पर आबंटन या पुनः आबंटन दिखाने के लिए बराबर चिह्न का उपयोग करते हैं।</li> <li>→ <b>newValue</b>: यह वह नया डेटा है जिसे आप स्टोर करना चाहते हैं!</li> </ul>

चलिए यह दिखाने के लिए अपने पिछले उदाहरण का उपयोग करते हैं कि हम किसी सूची में किसी विशिष्ट इंडेक्स पर मान कैसे आबंटित करते हैं।

```
प्रोग्रामर = ["Ada", "Grace", "Katherine", "Roya"]
```

- ❑ **अपनी List में "Katherine" को "Violet" में बदलने के लिए हम कौन सी कोड की पंक्ति लिखते हैं?**



अपने विचारों को पृष्ठ 3 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।

## चरण 6: सभी संभव वर्णों की रचना करें (10-15 मिनट)

अब हम आखिरकार अपने पासवर्ड जनरेटर को कोड करना शुरू करने के लिए तैयार हैं, लेकिन हमें कहाँ से शुरू करना चाहिए? चलिए हम अपनी जरूरत की पहली चीज, वर्णों के बारे में सोचते हैं! हमें अपने प्रोग्राम को बताना है कि पासवर्ड जनरेट करते समय कौन से वर्ण वैध विकल्प हैं। ऐसा करने के लिए हम संभव वर्णों, अक्षरों, संख्याओं, और विशेष वर्णों या प्रतीकों के प्रकारों को अलग करने के लिए तीन सूचियाँ बनाएंगे।

सरलता के लिए हमने समान सूचियों में अपर और लोअरकेस अक्षरों, दोनों को संयोजित करने का निश्चय किया। आप अपने पैटर्न को अधिक पेचीदा बनाने के लिए अक्षरों को दो अलग सूचियों में अलग करने का चुनाव कर सकते हैं, एक लोअरकेस अक्षरों के लिए और दूसरी अपरकेस अक्षरों के लिए।

### सूची चर राशियाँ बनाकर वर्ण के प्रकारों की संभावनाओं को स्टोर करें (5-8 मिनट)

- ❑ अपने पासवर्ड जनरेटर प्रोजेक्ट को Repl.it पर खोलें।
- ❑ इस अनुभाग की पहचान करने के लिए जहाँ आप अपनी सूचियाँ बनाने वाले हैं, शीर्ष पर एक कोड टिप्पणी जोड़ें। आपकी कोड टिप्पणी को "वर्णों के संभव प्रकारों के लिए सूचियाँ" जितनी सरल होना चाहिए। याद रखें कि कोड की टिप्पणियों की पंक्ति के आरंभ में एक # प्रतीक शामिल होना चाहिए!
- ❑ अपर और लोअरकेस अक्षरों के सभी संभव प्रकारों की सूची को स्टोर करने के लिए एक चर राशि बनाएं। वर्णमाला के प्रत्येक अक्षर को अपनी सूची में अपरकेस और लोअरकेस में अर्धविरामों से अलग रखते हुए शामिल करना सुनिश्चित करें। हमने इस चर राशि को **अक्षर** नाम दिया है, पर आप चाहें तो चर राशि को अपना मनचाहा नाम दे सकती हैं। याद रखें कि सूचियाँ स्क्वेयर ब्रैकेटों [] का उपयोग करके अपनी सामग्री के आरंभ और समाप्ति का संकेत देती हैं और अंदर की सारी जानकारी को अर्धविरामों से अलग करना चाहिए! चूंकि ये वर्ण हैं, इसलिए प्रत्येक अक्षर को स्ट्रिंग डेटा प्रकार का होना चाहिए तथा सिंगल या डबल कोटेशन मार्कों से युक्त होना चाहिए।
- ❑ सभी संभव संख्याओं की एक सूची स्टोर करने के लिए एक चर राशि बनाएं। सुनिश्चित करें की आपकी सूची में प्रत्येक संख्या को अर्धविरामों से अलग किया गया है। याद रखें कि हम इन संख्याओं को स्ट्रिंग या शब्दों के रूप में पढ़ेंगे, इसलिए आपको प्रत्येक संख्या के दोनों तरफ सिंगल " या डबल "" कोटेशन मार्क शामिल करने होंगे। हमने इस चर राशि को **संख्याएं** नाम दिया है, पर आप चाहें तो चर राशि को अपना मनचाहा नाम दे सकती हैं।
- ❑ सभी संभव विशेष वर्णों या प्रतीकों की एक सूची स्टोर करने के लिए एक चर राशि बनाएं। चाहें तो प्रत्येक प्रतीक को शामिल करें या केवल कुछ प्रतीकों को चुनें। अपने प्रतीकों में कोटेशन मार्क ( ' या " ) शामिल न करने की सावधानी बरतें क्योंकि पायथन इन्हें स्ट्रिंग निर्दिष्ट करने वाले मुख्य वर्णों के रूप में पढ़ता है। इससे आपके प्रोग्राम में भ्रम पैदा हो सकता है। हमने इसे **sChars** नाम दिया है, पर आप चाहें तो चर राशि को अपना मनचाहा नाम दे सकती हैं।



अपने विचारों को पृष्ठ 4 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।



## print() फंक्शन से मिलें (2 मिनट)

यदि आप **चलाएं** बटन पर क्लिक करते हैं तो आप देखेंगे कि कुछ भी नहीं होता है। ऐसा इसलिए होता है क्योंकि हम चर राशियाँ बनाते हैं लेकिन कंप्यूटर से उस जानकारी के साथ कुछ भी करने के लिए वास्तव में कुछ नहीं कहते हैं। अपने प्रोग्राम में डीबग करने, या त्रुटियों को पहचानने का एक बढ़िया तरीका है, यह जाँचने के लिए कि आपकी चर राशियाँ सही जानकारी से युक्त हैं, अपने कोड में `print()` फंक्शन का उपयोग करना। **फंक्शन** कोड की वे प्रक्रियाएं या ब्लॉक हैं जिन्हें कोई कार्य निष्पादित करने के लिए कहा जा सकता है। प्रोग्रामर अपने कोड को अधिक पठनीय बनाने के लिए अक्सर फंक्शनों का उपयोग करते हैं `print()` फंक्शन एक पूर्वस्थापित पायथन फंक्शन है जो किसी इनपुट को लेता है और उसे कंसोल में प्रदर्शित करता है।



PYTHON	विवरण
<code>print( input )</code>	<ul style="list-style-type: none"><li>→ <code>print()</code>: उस पायथन फंक्शन का नाम जो संदेश को कंसोल पर प्रिंट करता है।</li><li>→ <b>इनपुट</b>: यह इनपुट एक स्ट्रिंग या चर राशि के रूप में हो सकता है।</li></ul>

## अपने कोड को परखें (3-5 मिनट)

चलिए यह जाँच करने के लिए `print()` फंक्शन का उपयोग करते हैं कि हमारी सूचियों को सही ढंग से लागू किया गया है। प्रत्येक प्रिंट स्टेटमेंट के लिए आपको उन्हें अपनी सभी सूची चर राशियों के नीचे जोड़ना चाहिए।

- ❑ एक `print()` स्टेटमेंट जोड़कर अपनी अक्षरों की सूची की सामग्री को प्रिंट करें।
- ❑ एक `print()` स्टेटमेंट जोड़कर अपनी संख्याओं की सूची की सामग्री को प्रिंट करें।
- ❑ एक `print()` स्टेटमेंट जोड़कर अपनी विशेष वर्णों की सूची की सामग्री को प्रिंट करें।
- ❑ अपनी सूचियों की सामग्री को देखने के लिए चलाएं पर क्लिक करें।

आपके सामने होना चाहिए:

- सभी संभव अक्षरों की एक प्रिंट की हुई सूची।
- सभी संभव संख्याओं की एक प्रिंट की हुई सूची।
- सभी विशेष वर्णों की एक प्रिंट की हुई सूची।

```
['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']
['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
['!', '@', '#', '$', '%', '^', '&', '*', '(', ')', '-', '+']
```

जैसा चाहा था वैसे कार्य नहीं कर रहा? **ये डीबगिंग के सुझाव आजमाएं:**

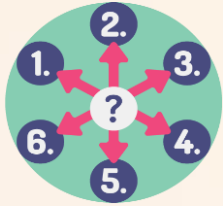
- क्या आपने फंक्शन के नाम की सही स्पेलिंग लिखी?
- क्या आपने अपनी सूची में प्रत्येक वर्ण को अर्धविरामों से अलग किया?
- क्या आपने प्रत्येक वर्ण के दोनों ओर सिंगल या डबल कोटेशन मार्क लगाए?
- क्या आपने प्रत्येक सूची की सामग्री को दोनों ओर स्क्वेयर ब्रैकेट लगाए?
- क्या आपने अतिरिक्त कोष्ठकों या ब्रैकेटों का उपयोग किया है? आप देखेंगे कि जब आप ( सिम्बल टाइप करते हैं तो एडिटर अपने-आप क्लोजिंग ब्रैकेट ) जोड़ देता है। इसलिए हो सकता है कि आप गलती से अतिरिक्त ब्रैकेट टाइप कर जाएं।



अपने विचारों को पृष्ठ 4 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।



## चरण 7: सूचियों में रैंडमनेस को समझें (5 मिनट)



अब समय है अपना रैंडम रूप से जनरेट किया गया पासवर्ड बनाने का। अपने फाइनल पासवर्ड को स्टोर करने के लिए हम एक पासवर्ड **सूची** बनाएंगे। फिर हम अपने रैंडम वर्णों को प्रत्येक इंडेक्स आबंटित करेंगे। यह काफी सारा काम लग सकते हैं लेकिन वास्तव में थोड़ी सी कोड की पंक्तियों से पूरा किया जा सकता है!

### रैंडम लाइब्रेरी से मिलें (2 मिनट)

**लाइब्रेरी** फंक्शनों और चर राशियों का एक संकलन है। प्रोग्रामर लाइब्रेरियों का उपयोग करते हैं क्योंकि कई फंक्शनों को आपके उपयोग के लिए पहले से लिखा जा चुका है! यह पूर्व-स्थापित पायथन लाइब्रेरी हमारी **सूचियों** से वर्णों को रैंडम ढंग से चुनने में हमारी मदद करेगी। इससे पहले कि हम लाइब्रेरी में उपलब्ध किसी फंक्शन का उपयोग कर पाएं, हमें पहले कंप्यूटर से लाइब्रेरी को **इम्पोर्ट**, या लोड करने के लिए कहना होगा।

PYTHON	विवरण
<code>import random</code>	<ul style="list-style-type: none"><li>→ <b>import</b>: यह कीवर्ड कंप्यूटर को एक पायथन लाइब्रेरी से जानकारी लोड करना जानने देता है।</li><li>→ <b>random</b>: इस पायथन लाइब्रेरी का उपयोग संख्याओं को रैंडम तरीके से जनरेट करने के लिए किया जाता है।</li></ul>

**random** लाइब्रेरी में कई फंक्शन हैं लेकिन हम अपने प्रोजेक्ट में **choice()** फंक्शन का उपयोग करेंगे। यह फंक्शन हमें एक **सूची** या अनुक्रम से एक आइटम को रैंडम रूप से चुनने की अनुमति देता है। चलिए इस फंक्शन का सिंटेक्स देखते हैं।

PYTHON	विवरण
<code>random.choice(listName)</code>	<ul style="list-style-type: none"><li>→ <b>random</b>: यह कंप्यूटर को यह जानने देता है कि हम <b>random</b> लाइब्रेरी से एक फंक्शन का उपयोग करेंगे।</li><li>→ <b>.</b>: इस प्रतीक का उपयोग अक्सर किसी लाइब्रेरी या किसी और फाइल से किसी फंक्शन को कॉल करने के लिए किया जाता है।</li><li>→ <b>choice()</b>: यह उस फंक्शन का नाम जिसका उपयोग हम करना चाहते हैं।</li><li>→ <b>listName</b>: उस <b>सूची</b> का नाम जिसमें से हम चुनना चाहते हैं।</li></ul>

## चरण 8: रैंडम वर्ण संख्या 1 जनरेट करें (5-10 मिनट)



हम अब अपने रैंडम रूप से जनरेट किए गए पासवर्डों को कोड करने के लिए पूरी तरह से सुसज्जित हैं! आप चाहें तो इस चरण के लिए पासवर्ड (पेज 7 पर) के लिए अपने नियोजित पैटर्न को प्रिंट करना या आसान एक्सेस योग्य बना सकते हैं।

पिछले अनुभाग में हमने अपने कोड को डीबग करने और यहाँ जाँचने के लिए कि हमारी सूचियों को ठीक से लागू किया गया है, `print()` स्टेटमेंट जोड़े थे। आप इन कोड की पंक्तियों को *डिलीट करने* या उन पर *टिप्पणी करने* का चुनाव कर सकते हैं क्योंकि वे इस प्रोजेक्ट के समग्र लक्ष्य में योगदान नहीं करती हैं।

- ❑ अपने प्रोग्राम के शीर्ष पर, `random` लाइब्रेरी इम्पोर्ट करने के लिए, अपनी चर राशियों के ऊपर एक नई कोड की पंक्ति जोड़ें। आप खुद को याद दिलाने के लिए कि यह कोड की पंक्ति क्या करती है एक टिप्पणी भी जोड़ सकते हैं। आम तौर पर प्रोग्रामर इम्पोर्ट करने वाली लाइब्रेरियों जैसे किसी भी संसाधन को अपने प्रोग्राम में पहली कुछ कोड की पंक्तियों के रूप में जोड़ते हैं।
- ❑ एक नई पासवर्ड चर राशि बनाएं और अपने कोड के तल में 10 खाली मानों की एक सूची आबंटित करें। हमने इस चर राशि को `pw` नाम दिया है, पर आप चाहें तो चर राशि को अपना मनचाहा नाम दे सकती हैं। आप चाहें तो 0s या खाली स्ट्रिंग्स `""` से युक्त एक सूची बना सकते हैं। हम अगले चरणों में इनमें से प्रत्येक मान को प्रतिस्थापित करेंगे।
- ❑ अपनी पासवर्ड सूची के 0वें इंडेक्स को अपने वर्णों की सूची में से एक में किसी रैंडम आइटम पर आबंटित करें जिसके लिए `random.choice()` फंक्शन का उपयोग करें। यह निर्धारित करने के लिए कि प्रत्येक इंडेक्स से आपको कौन सी सूची चुननी है अपने पासवर्ड पैटर्न का उपयोग करें। उदाहरण के लिए, यदि आप अपने पासवर्ड में किसी अक्षर को पहले दिखाना चाहते हैं, तो अक्षरों की सूची को `random.choice()` फंक्शन में एक इनपुट के रूप में इंगित करना सुनिश्चित करें।
- ❑ अपनी पासवर्ड सूची को प्रिंट करने के लिए एक कोड की पंक्ति जोड़ें। आगे बढ़ने से पहले, हम प्रिंट करना चाहेंगे कि हमारी सूची कैसी दिखती है ताकि हम सुनिश्चित कर सकें कि हमने अपनी सूची में एक वर्ण को सफलतापूर्वक पुनः प्राप्त और 0वें इंडेक्स पर मान को पुनः आबंटित कर दिया है।



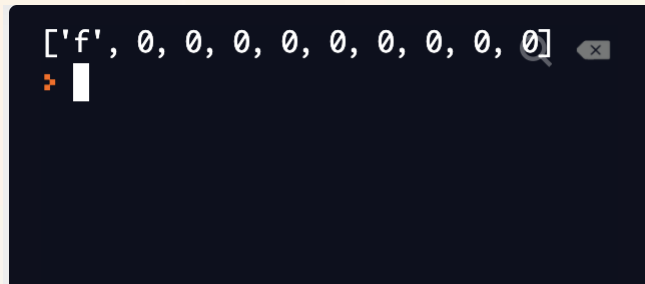
अपने विचारों को पृष्ठ 5 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।

### अपने कोड को परखें (3-5 मिनट)

चलिए अब तक हमने जो लिखा है उसे परख कर यह सुनिश्चित करते हैं कि हमारा प्रोग्राम उसी तरह चल रहा हो जैसे हम चाहते हैं।

**चलाएं** बटन पर क्लिक करके अपना रेखा-चित्र चलाएं। आपके सामने होना चाहिए:

- एक छपी हुई पासवर्ड **सूची** जहाँ पहला वर्ण एक रैंडम अक्षर, संख्या, या विशेष वर्ण है और शेष वर्ण या तो 0s हैं या खाली स्ट्रिंग "" हैं।



जैसा चाहा था वैसे कार्य नहीं कर रहा? **ये डीबगिंग के सुझाव आजमाएं:**

- क्या आपने फंक्शन और लाइब्रेरी के नाम की सही स्पेलिंग लिखीं?
- क्या आपने अपनी **सूची** में प्रत्येक वर्ण को अर्धविरामों से अलग किया?
- क्या आपने प्रत्येक वर्ण के दोनों ओर सिंगल या डबल कोटेशन मार्क लगाए?
- क्या आप तब स्क्वेयर ब्रैकेटों [ ] का उपयोग करते हैं जब आप किसी **सूची** को कॉल करते हैं या उस पर मान आबंटित करते हैं?
- क्या आपने अतिरिक्त कोष्ठकों या ब्रैकेटों का उपयोग किया है? आप देखेंगे कि जब आप ( सिम्बल टाइप करते हैं तो एडिटर अपने-आप क्लोजिंग ब्रैकेट ) जोड़ देता है। इसलिए हो सकता है कि आप ग़लती से अतिरिक्त ब्रैकेट टाइप कर जाएं।

## चरण 9: अपना रैंडम पासवर्ड बनाएं (5-10 मिनट)



अब जब हमने समझ लिया है कि पहला रैंडम वर्ण कैसे जनरेट करते हैं, चलिए हमारे पासवर्ड के शेष पैटर्न को कोड करना जारी रखते हैं! अपनी पासवर्ड सूची के शेष इंडेक्स आबंटित करते समय आपको यह काम अपने पासवर्ड के लिए प्रिंट स्टेटमेंट के ऊपर करना चाहिए।

- ❑ अपनी पासवर्ड सूची के शेष इंडेक्सों पर अपने वर्णों की सूची में से एक में किसी रैंडम आइटम को आबंटित करें जिसके लिए `random.choice()` फंक्शन का उपयोग करें। आपको यह चरण 9 बार दोहराना पड़ेगा, जिसमें आपको वर्ण प्रकार सूची में एक रैंडम वर्ण पर 1 से 9 तक प्रत्येक शेष इंडेक्स आबंटित करना होगा।

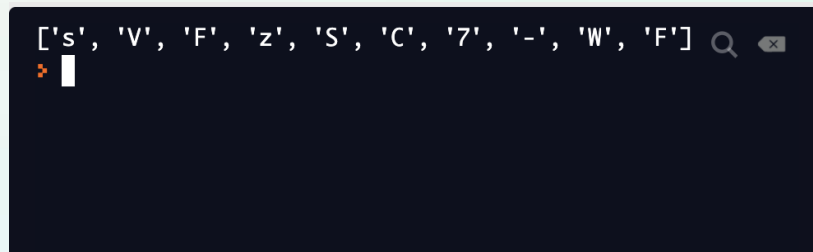


अपने विचारों को पृष्ठ 6 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।

### अपने कोड को परखें (3-8 मिनट)

चलिए अब तक हमने जो लिखा है उसे परख कर यह सुनिश्चित करते हैं कि हमारा प्रोग्राम उसी तरह चल रहा हो जैसे हम चाहते हैं। **चलाएं** बटन पर क्लिक करके अपना रेखा-चित्र चलाएं। आपके सामने होना चाहिए:

- एक छपी हुई पासवर्ड **सूची** जहाँ प्रत्येक वर्ण एक रैंडम अक्षर, संख्या, या विशेष वर्ण है और आपके पासवर्ड के नियोजित पैटर्न का पालन करता है।



जैसा चाहा था वैसे कार्य नहीं कर रहा? **ये डीबगिंग के सुझाव आजमाएं:**

- क्या आपने फंक्शन और लाइब्रेरी के नाम की सही स्पेलिंग लिखीं?
- क्या आपने अपनी **सूची** में प्रत्येक वर्ण को अर्धविरामों से अलग किया?
- क्या आपने प्रत्येक वर्ण के दोनों ओर सिंगल या डबल कोटेशन मार्क लगाए?
- क्या आप तब स्क्वेयर ब्रैकेटों `[ ]` का उपयोग करते हैं जब आप किसी **सूची** को कॉल करते हैं या उस पर मान आबंटित करते हैं?
- क्या आपने अतिरिक्त कोष्ठकों या ब्रैकेटों का उपयोग किया है? आप देखेंगे कि जब आप ( सिम्बल टाइप करते हैं तो एडिटर अपने-आप क्लोजिंग ब्रैकेट ) जोड़ देता है। इसलिए हो सकता है कि आप गलती से अतिरिक्त ब्रैकेट टाइप कर जाएं।
- क्या आपकी कोड की पंक्तियाँ सही स्थान पर और सही क्रम में है? याद रखें कि प्रोग्राम फ़्लो में क्रम मायने रखता है!

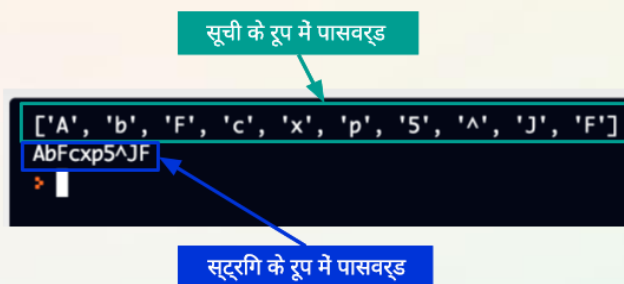
## चरण 10: विस्तार (2-20 मिनट)



अपने पासवर्ड जनरेटर में अधिक फंक्शन और पेचीदगी जोड़ने के लिए इन एक्सटेंशनों का उपयोग करें। इन चुनौतियों को अधिक कठिन होने के लिए बनाया गया है, लेकिन हिम्मत मत हारें! प्रत्येक चुनौती में कवर की गई अवधारणाओं के बारे में अधिक जानने के लिए एक्सटेंशन संसाधनों का उपयोग करें। प्रदान किए गए प्रोजेक्ट को देखना और कोड को देखकर पीछे की ओर से निर्माण करना भी उपयोगी हो सकता है।

### एक्सटेंशन 1: अपने पासवर्ड को एक स्ट्रिंग के रूप में प्रिंट करना (2-5 मिनट)

आप को दिख सकता है कि आपकी पासवर्ड सूची को प्रिंट करने से आपके पासवर्ड में प्रत्येक वर्ण के बीच अर्धविराम दिखाई देते हैं। यदि आप अपने पासवर्ड को स्ट्रिंग के रूप में दिखाना चाहते हैं तो हम `join()` फंक्शन का उपयोग कर सकते हैं।



इस एक्सटेंशन का उदाहरण रेखा-चित्र देखने के लिए [यहां](#) क्लिक करें।

पायथन में `join()` फंक्शन आपको एक दोहराने योग्य वस्तु, या सूचियों जैसी संरचनाओं को लेने की अनुमति देता है जो एकाधिक जानकारी को धारण करती हैं जहाँ प्रत्येक आइटम को आसानी से एक्सेस किया जा सकता है, और प्रत्येक आइटम को एक स्ट्रिंग, या शब्द के साथ संयोजित करती है या जोड़ती है।

PYTHON	विवरण
<code>stringName.join(itrObject)</code>	<ul style="list-style-type: none"><li>→ <code>stringName</code>: यह एक चर राशि है जो एक स्ट्रिंग, या शब्द, प्रकार को धारण करती है।</li><li>→ <code>.</code>: इस प्रतीक का उपयोग अक्सर किसी फंक्शन को कॉल करने का संकेत देने के लिए किया जाता है।</li><li>→ <code>join()</code>: उस फंक्शन का नाम जो दोहराने योग्य वस्तु और स्ट्रिंग को एक स्ट्रिंग के रूप में जोड़ता है।</li><li>→ <code>itrObject</code>: यह एक चर राशि है जो एक दोहराने योग्य वस्तु, जैसे कि एक सूची को धारण करती है।</li></ul>

- ❑ किसी खाली स्ट्रिंग `""` को पासवर्ड सूची के साथ संयोजित करने के लिए `join()` फंक्शन का उपयोग करें और इसे एक चर राशि में स्टोर करें। हमने इस चर राशि को `pwS` नाम दिया है, पर आप चाहें तो चर राशि को अपना मनचाहा नाम दे सकती हैं। आप खुद को याद दिलाने के लिए कि यह कोड की पंक्ति क्या करती है एक टिप्पणी भी जोड़ सकते हैं।
- ❑ अपना पासवर्ड स्ट्रिंग प्रिंट करें।

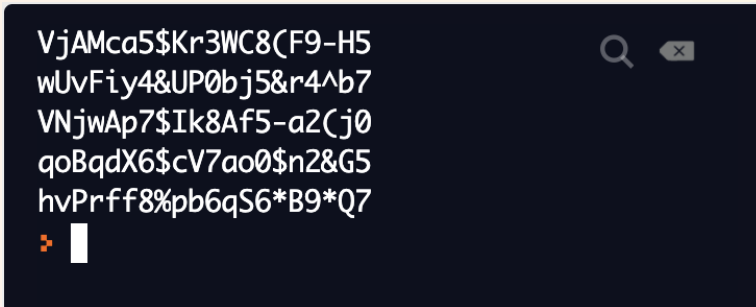
#### एक्सटेंशन संसाधन

नीचे कुछ सहायक संसाधन दिए जा रहे हैं जिनका उपयोग करके हमने यह एक्सटेंशन बनाया था। इनसे आपको शुरुआत करने में मदद मिलेगी, पर याद रखें कि और भी बहुत से संसाधन मौजूद हैं जो आपसे बस एक सर्च इंजन की दूरी पर हैं!

- [join\(\)](#)
- [दोहराने योग्य वस्तुएं](#)

## एक्सटेंशन 2: अपने पासवर्ड की लंबाई बढ़ाना (10-15 मिनट)

इस समय हमारा पासवर्ड 10 वर्ण लंबा है, लेकिन यदि हम अपने पासवर्ड को और भी अधिक सुरक्षित बनाना चाहते हों तो क्या? अधिकांश पासवर्ड मैनेजर आपको कम से कम 15 वर्णों का या यहाँ तक कि 20 वर्णों का पासवर्ड बनाने का सुझाव देंगे!



इस एक्सटेंशन का उदाहरण रेखा-चित्र देखने के लिए [यहां](#) क्लिक करें।

अपनी पासवर्ड सूची में अतिरिक्त वर्ण जोड़कर और शेष वर्णों को रैंडम मानों पर आबंटित करके अपने पासवर्ड को अधिक पेचीदा बनाएं।

- ❑ अपने पासवर्ड के शेष पैटर्न की योजना बनाएं।
- ❑ अपनी पासवर्ड सूची चर राशि में, अपनी सूची को वांछित लंबाई का बनाने के लिए अतिरिक्त आइटम जोड़ें। अपने पासवर्ड को 10 से बढ़ाकर 20 वर्णों का बनाने के लिए, हमने अपनी सूची में स्क्वेयर ब्रैकेटों `[]` के बीच 10 और `0` जोड़ें।
- ❑ अपनी पासवर्ड सूची के शेष इंडेक्सों पर अपने वर्णों की सूची में से एक में किसी रैंडम आइटम को आबंटित करें जिसके लिए `random.choice()` फंक्शन का उपयोग करें।

### एक्सटेंशन 3: रैंडम वर्ण पैटर्न वाले पासवर्ड जनरेट करें (10-15 मिनट)

हमारे प्रोग्राम में आपने अपने पासवर्ड के पैटर्न को चुना, लेकिन यदि हम अपने पासवर्डों के लिए पैटर्न को यादृच्छीकृत करना चाहते हैं तो क्या होगा? हम अपनी किसी भी वर्ण सूची से रैंडम तरीके से चुनने के लिए `random` लाइब्रेरी का उपयोग कर सकते हैं। इस एक्सटेंशन के लिए, हमारा पासवर्ड जनरेट करने के लिए आपको एक **लूप** और **कंडीशनल्स** का उपयोग करने की जरूरत पड़ेगी।



इस एक्सटेंशन का उदाहरण रेखा-चित्र देखने के लिए [यहां](#) क्लिक करें।

इस एक्सटेंशन में, अपने पासवर्ड में प्रत्येक वर्ण को एक विशिष्ट पैटर्न आबंटित करने की बजाय रैंडम प्रकार का वर्ण रैंडम तरीके से आबंटित करने के लिए एक लूप का उपयोग करने की कोशिश करें। इस एक्सटेंशन को पूरा करने के लिए आपको इन बुनियादी चरणों की ज़रूरत होगी:

- ❑ एक नई पासवर्ड चर राशि बनाएं और अपने कोड के तल में 10 खाली मानों की एक सूची आबंटित करें।
- ❑ एक `for` लूप बनाएं जिसमें `range()` फंक्शन हो जो 10 बार दोहराया जाता है। हम चर राशि `i` का उपयोग करके `randint` फंक्शन के आइटरेशन मान को स्टोर करते हैं लेकिन आप चर राशि को अपना खुद का नाम दे सकते हैं।
- ❑ अपने `for` लूप के भीतर, 1 से 3 के बीच एक संख्या को रैंडम तरीके से चुनने के लिए `random.randint()` फंक्शन का उपयोग करें और उसके परिणाम को एक चर राशि में स्टोर करें। हम यह चुनने के लिए कि किस प्रकार के वर्ण को चुनना चाहिए हम एक रैंडम संख्या को चुनेंगे। चूंकि हमारे पास 3 प्रकार (अक्षर, संख्याएं, और विशेष वर्ण) हैं, हम अपनी रेंज को केवल 1, 2, और 3 संख्याओं तक ही सीमित करेंगे। हमने इस चर राशि को `charType` नाम दिया है, पर आप चाहें तो चर राशि को अपना मनचाहा नाम दे सकती हैं।
- ❑ अपने `for` लूप के भीतर, कंडीशनल स्टेटमेंट जोड़ें जो आपकी पासवर्ड सूची में `randint()` फंक्शन द्वारा रैंडम तरीके से चुने गए वर्ण प्रकार के आधार पर वर्ण आबंटित करते हैं। हमारे उदाहरण में हमने कहा कि यदि `charType` 1 हो, तो हम एक अक्षर को रैंडम तरीके से चुनेंगे। यदि `charType` 2 हो, तो हम रैंडम तरीके से एक संख्या चुनेंगे। अंत में यदि `charType` 3 हो, तो हम एक विशेष संख्या को रैंडम तरीके से चुनेंगे। निस्संकोच होकर अपने खुद के मानों को वर्ण के प्रकारों के एक अलग क्रम पर आबंटित करें।

हमें पासवर्ड सूची में प्रत्येक वर्ण को आबंटित करने के लिए अपने `for` लूप से `i` चर राशि का उपयोग करने की जरूरत पड़ेगी। `i` चर राशि प्रत्येक `for` लूप आइटरेशन के बाद अपडेट होगी और 0 से 9 तक के मानों को धारण करेगी।

- ❑ अपने `for` लूप के बाहर, अपना रैंडम तरीके से जनरेट किया गया पासवर्ड प्रिंट करें!



**एक्सटेंशन संसाधन**

नीचे कुछ सहायक संसाधन दिए जा रहे हैं जिनका उपयोग करके हमने यह एक्सटेंशन बनाया था। इनसे आपको शुरुआत करने में मदद मिलेगी, पर याद रखें कि और भी बहुत से संसाधन मौजूद हैं जो आपसे बस एक सर्च इंजन की दूरी पर हैं!

- [लूप्स के लिए](#). आप इस उदाहरण के लिए एक [while लूप](#) का उपयोग भी कर सकते हैं लेकिन हम for लूप का सुझाव देते हैं क्योंकि हम दोहरावों की विशिष्ट संख्या जानते हैं।
- [range\(\) फंक्शन for लूपों के साथ](#)।
- [random.randint\(\)](#)
- [कंडीशनल स्टेटमेंट](#)

## चरण 11: अपने Girls Who Code at Home परियोजना को साझा करें! (5-10 मिनट)

हम आपके काम को देखना पसंद करेंगे और हम जानते हैं कि दूसरे भी ऐसा करेंगे। अपनी अंतिम परियोजना को हमारे साथ साझा करें। [@girlswhocode](#) [#codefromhome](#) को टैग करना मत भूलें, और हो सकता है कि हम आपको हमारे खाते में प्रदर्शित कर देंगे!

**केवल देखने का एक्सेस (1 मिनट)**

Repl.it पर आपके काम को साझा करना आसान है! बस अपने Repl प्रोजेक्ट के URL पते को शीर्ष पर वेब एड्रेस बार में कॉपी और पेस्ट करें! इससे अन्य लोग आपके प्रोजेक्ट को चलाने, आपका कोड देखने, और आपके प्रोजेक्ट को “फोर्क” करने या डुप्लीकेट करने, तथा अपने खुद के प्रोजेक्ट के साथ रीमिक्स करने में सक्षम होंगे।

इस लिंक को अपने सोशल मीडिया खातों पर साझा करें तथा [@girlswhocode](#) [#codefromhome](#) को टैग करना मत भूलें और हो सकता है कि हम आपको अपने खाते में शामिल कर लें!

Repl.it - GWC\_passwordGenerator

https://repl.it/@samanthahegwc/GWCpasswordGenerator#...

Run Upgrade Share

main.py

```

1 # Import the random library
2 import random
3
4 # Lists for possible types of characters
5 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',

```

इस URL पते को अपनी परियोजना में कॉपी और साझा करें!

अपने प्रोजेक्ट में सहयोगियों को जोड़ने के लिए शेयर बटन पर क्लिक करें!

## चरण 11: अपने Girls Who Code at Home परियोजना को साझा करें! (जारी)

### सहयोगी जोड़ना (2 मिनट)

यदि आप किसी प्रोजेक्ट पर मित्रों के समूह के साथ काम करना चाहते हैं, तो आप विंडो के ऊपरी-दाएं भाग में स्थित **साझा करें** बटन का उपयोग करके उन्हें सहयोग करने के लिए आसानी से आमंत्रित कर सकते हैं। इससे एक नई विंडो खुलेगी जिसमें आपके प्रोजेक्ट पर सहयोग करने के लिए अन्य लोगों को आमंत्रित करने के दो विकल्प होंगे।

- **ईमेल या Repl.it यूजरनेम के द्वारा आमंत्रित करें।** यह विकल्प आपको विशिष्ट लोगों के साथ अपना प्रोजेक्ट साझा करने के लिए उनका ईमेल पता या यदि उनके पास पहले से ही Repl.it के साथ खाता है तो Repl.it यूजरनेम टाइप करके आमंत्रित करने देता है। हम यह सुनिश्चित करने के लिए कि आप अपने प्रोजेक्ट को सही लोगों के साथ साझा कर रहे हैं, इस विकल्प की अनुशंसा करते हैं!
- **आमंत्रण लिंक साझा करें।** विंडो के तल पर एक अद्वितीय आमंत्रण लिंक है। आप इस लिंक को मित्रों को कॉपी और पेस्ट कर सकते हैं ताकि वे आपके प्रोजेक्ट को एक्सेस कर सकें।

**सहयोगियों के बारे में टिप्पणी:** याद रखें कि सहयोगी जोड़ने से अन्य लोगों को आपके प्रोजेक्ट को संपादित करने की सुविधा मिलती है। इससे वे आपके कोड, नाम, और विवरण को बदलने में सक्षम होते हैं। **अपने आमंत्रण लिंक को सोशल मीडिया पर साझा मत करें!** इन संपादन अधिकारों को आप किनके साथ साझा करेंगे इस बारे में चयनशील रहें।

**मिसिंग कोड को डीबग करें भाग 2 में अधिक जानकारी की प्रतीक्षा करें!**

