



# Girls Who Code At Home

क्या मैं आपकी मदद कर सकता हूँ?  
पाइथन में चैटबॉट

## गतिविधि अवलोकन

इस गतिविधि में आप पाइथन में कंप्यूटर विज्ञान की बुनियादी अवधारणाओं की खोज करेंगे, जबकि चैटबॉट या स्वचालित सहायता प्रणाली बनाकर अपने समुदाय की मदद करना सीखेंगे! आपने शायद पहले चैटबॉट में बातचीत की है, शायद इसका एहसास किए बिना भी। कुछ सामान्य चैटबॉट में शामिल हैं: एप्पल की सिरी, अमेज़ॅन की एलेक्सा, गूगल एसिसटेंट, लिफ्ट (Lyft), और हाल ही में विश्व स्वास्थ्य संगठन ने [कोविड-19 तथ्यों](#) के लिए इसे बनाया है। इससे पहले कि आप अपने चैटबॉट को डिजाइन करना और कोड करना शुरू करें, हमारे टेक स्पोर्टलाइट में विशेष रूप से प्रदर्शित महिला, एरिका कोच्ची को देखें। यूनिसेफ के इनोवेशन यूनिट के सह-संस्थापक के रूप में, एरिका ने नाइजीरिया और रवांडा में लोगों को देखभाल और सहायता प्रदान करने के लिए एक चैटबॉट बनाया।

## सामग्री

- [ट्रिन्केट एडिटर](#)
- [स्टार्टर ट्रिन्केट कोड](#)
- [उदाहरण चैटबॉट प्रॉजेक्ट](#)
- वैकल्पिक: योजना गाइड
- वैकल्पिक: पेन/पेंसिल/मार्कर

## टेक स्पॉटलाइट में महिला: एरिका कोच्चि



एरिका कोच्चि ने [यूनिसेफ](#) के इनोवेशन यूनिट की सह-स्थापना की और सह-नेतृत्व किया, जिसका उद्देश्य तकनीकी नवाचारों का उपयोग करके वैश्विक स्वास्थ्य में सुधार करना है। [RapidSMS](#) नामक उनकी परियोजना ने स्वास्थ्य और शिक्षा सेवाओं के लिए डेटा संग्रह की सुविधा में मदद के लिए मोबाइल फोन और एसएमएस संदेशों का उपयोग किया।

सहयोगियों के साथ कोच्चि के काम ने

खुले स्रोत प्रौद्योगिकियों को विकसित करने में मदद की जिसने नाइजीरिया में 7 मिलियन से अधिक जन्मों को पंजीकृत किया और रवांडा में हजारों गर्भवती महिलाओं को देखभाल प्रदान की। 2013 में, उन्हें TIME 100 के "विश्व के सबसे प्रभावशाली लोगों" में नामित किया गया था!

एरिका कोच्चि और [यूनिसेफ](#) में किए गए उसके कुछ कामों के बारे में जानने के लिए यह [वीडियो](#) देखें।

### सोचें

कोडिंग में केवल महान होने की तुलना में कंप्यूटर वैज्ञानिक होना कहीं अधिक है। इस बात पर विचार करने के लिए कुछ समय लें कि एरिका और उसका काम उन शक्तियों से कैसे संबंधित है जो महान कंप्यूटर वैज्ञानिक निर्माण पर ध्यान केंद्रित करते हैं - जैसे कि बहादुरी, लचीलापन, रचनात्मकता और उद्देश्य।



बहादुरी

एरिका कोच्चि जो कुछ करती है उसका एक बड़ा हिस्सा दूसरों के दुखों में मदद करने की कोशिश करना है। करियर में यह कैसे चुनौतीपूर्ण हो सकता है?

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

## कदम 1: अन्वेषण करें (5 मिनट)

इस ट्यूटोरियल में आप पाइथन में एक कस्टम चैटबॉट बनाएंगे! **चैटबॉट** एक कंप्यूटर प्रोग्राम है जो एक वास्तविक व्यक्ति के साथ बातचीत को अनुरूप बनाता है। यह आर्टिफिशियल इंटेलिजेंस या एआई का एक बहुत ही सरल रूप है। कुछ चैटबॉट्स जिनका आप पहले से ही उपयोग कर रहे हो सकते हैं, उनमें शामिल हैं: [स्टारबक्स बरिस्ता \(Starbucks Barista\)](#), [ऐपल की सिरी \(Apple's Siri\)](#), [अमेज़न की एलेक्सा \(Amazon's Alexa\)](#), और [गूगल असिस्टेंट \(Google Assistant\)](#)।

प्रौद्योगिकी में महिलाओं के बारे में हमारे द्वारा बनाई गई **इस चैटबॉट** की कुछ विशेषताओं का पता लगाने के लिए 5 मिनट का समय लें। हमारे उदाहरण के लिए हमने निम्नलिखित विषय, दर्शकों और लक्ष्य का चयन किया:

- **प्रॉजेक्ट थीम:** महिला/प्रौद्योगिकी
- **दर्शक:** वे लोग जो तकनीक में काम करने वाली महिलाओं के बारे में सीखना चाहते हैं
- **लक्ष्य:** प्रौद्योगिकी आधारित चुटकुले साझा करें और प्रौद्योगिकी में महिलाओं (Women in Technology) के बारे में जानकारी प्रदान करें

जैसा कि आप उदाहरण प्रॉजेक्ट का अन्वेषण करते हैं, निम्नलिखित के बारे में सोचें:

- चैटबॉट का परिचय कैसे दिया जाता है?
- किस प्रकार के प्रश्न पूछे जा रहे हैं? क्या इससे लक्ष्य पूरा होता है?
- उदाहरण में से आप कौन सी विशेषताएं अपने स्वयं के प्रॉजेक्ट में शामिल करना चाहेंगे? आप अपने खुद के प्रॉजेक्ट में अलग तरह से क्या करना चाहेंगे?

## कदम 2: विशेषताओं का मंथन करें और अपनी परियोजना की योजना बनाएं (10 मिनट)

अब जब आपको नमूना परियोजना का पता लगाने का मौका मिला है, तो सबसे पहले गेम प्लान बनाने के लिए कुछ समय निकालना एक अच्छा विचार है। इस समय का उपयोग यह पता लगाने के लिए करें कि आप अपने प्रॉजेक्ट से क्या करना चाहते हैं और लक्ष्य क्या होगा। आप अपने विचारों को कैप्चर और व्यवस्थित करने में मदद करने के लिए योजनाबद्ध दस्तावेज़ (पृष्ठ 16-18 पर) का उपयोग कर सकते हैं।

### 1. अपने चैटबॉट के लिए एक थीम, दर्शक और लक्ष्य चुनें।

इस बारे में सोचें कि आप अपने चैटबॉट के साथ किस विषय पर ध्यान केंद्रित करना चाहते हैं और इसे क्या पूरा करना चाहिए। आप एक सूचनात्मक, हास्यपूर्ण, सामुदायिक-निर्माण या एक तथ्य/प्रश्नोत्तरी चैटबॉट बना सकते हैं। यदि आप बाधित हो रहे हैं, तो यहां कुछ संभावित परियोजना विचार हैं:

- चैटबॉट जो दूसरों को प्रोत्साहन के शब्द प्रदान करता है यदि वे घर पर ऊब और/या अलग-थलग महसूस कर रहे हैं
- चैटबॉट जो घर पर अधिक मज़ा शामिल करने के बारे में सुझाव देता है
- मजेदार चुटकुले सुनाने वाला एक चैटबॉट
- चैटबॉट जो आपके पसंदीदा शो, कलाकार, संगीतकार या शौक के बारे में तथ्य साझा करता है

अपने चैटबॉट के दर्शक का चयन करते समय सोचें कि ऐसे लक्षित दर्शक कौन हैं जो आपके उत्पाद का उपयोग करने के लिए उत्साहित होंगे। विचार करें कि वे क्या जानने में रुचि रखते हैं और आप उनका ध्यान कैसे आकर्षित करेंगे।



## कदम 2: अपने प्रॉजेक्ट की योजना बनाएँ जारी

2. अपने चैटबॉट के लिए प्रत्येक संभावित उत्तर के लिए प्रतिक्रियाओं को पूछने और ड्राफ्ट करने के लिए तीन "हां या नहीं" प्रश्न चुनें।

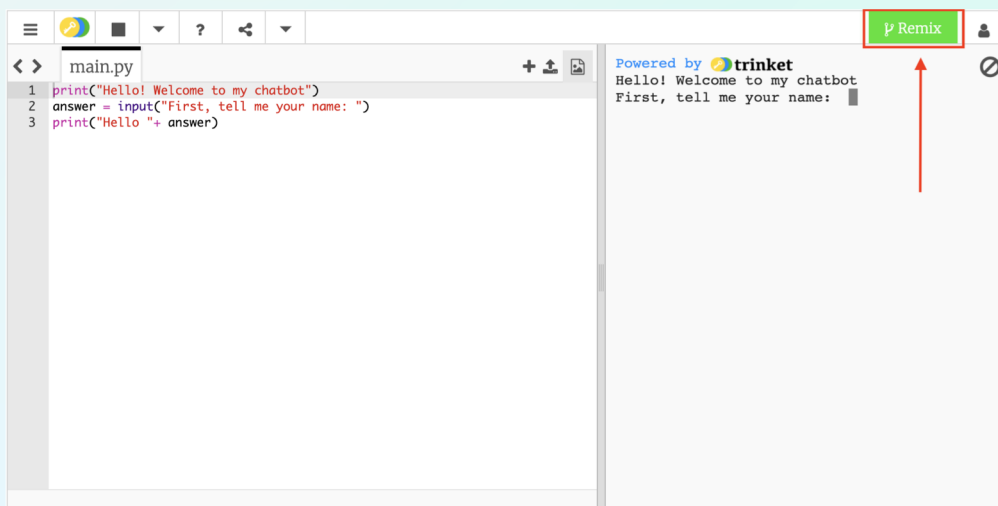
इस बारे में सोचें कि आप कौन से प्रश्न पूछना चाहते हैं और कैसे वे प्रश्न आपके विषय और परियोजना के लक्ष्य से जुड़े हैं। यदि उपयोगकर्ता की प्रतिक्रिया आपके प्रश्न के लिए "हाँ" है तो क्या होगा? क्या होगा अगर प्रतिक्रिया "नहीं" है? यदि उपयोगकर्ता हाँ या नहीं टाइप नहीं करता है तो क्या होगा? आपके चैटबॉट को इन उपयोगकर्ता त्रुटियों का जवाब कैसे देना चाहिए?

## कदम 3: ट्रिंकेट को आरंभ करें (10 मिनट)

**पाइथन** एक टेक्स्ट-आधारित प्रोग्रामिंग भाषा है, जिसका अर्थ है कि सभी कमांड्स को टाइप करने की आवश्यकता होगी! पाइथन में, प्रोग्रामर सूचनाओं को स्टोर करने और कमांड करने में मदद करने के लिए कई वेरिएबल, डेटा संरचना और फ़ंक्शन का उपयोग करते हैं! चूंकि पाइथन टेक्स्ट-आधारित है, इसलिए यह स्क्रैच जैसी अन्य भाषाओं की तुलना में थोड़ा अधिक कठिन है, लेकिन असंभव नहीं है!

हम अपने पाइथन कोड को लिखने और चलाने के लिए [ट्रिंकेट](#) कोडिंग प्लेटफॉर्म का उपयोग करेंगे। कई अलग-अलग कार्यक्रम हैं जिनका उपयोग पाइथन कोड लिखने और चलाने के लिए किया जा सकता है। हमने इसे इसलिए चुना क्योंकि आप इसे सीधे अपने ब्राउज़र में उपयोग कर सकते हैं!

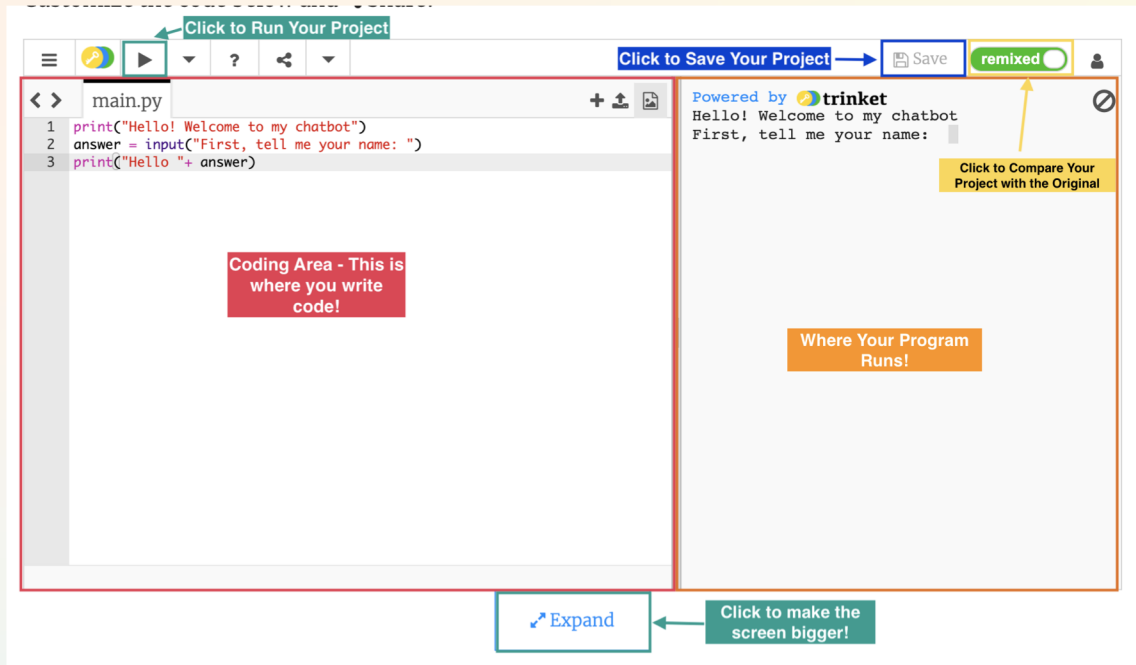
1. [Trinket.io](#) में साइन अप या लॉग इन करें  
यदि आपका पहले से ही अकाउंट नहीं है, तो ट्रिंकेट पर अपने काम को सहेजने के लिए आपको खाता बनाना होगा। खाता बनाने के लिए साइन अप फॉर्म पर निर्देशों का पालन करें। यदि आपकी उम्र 13 वर्ष से कम है, तो आपको साइन अप करने के लिए अपने माता-पिता के ईमेल पते की आवश्यकता होगी।
2. अपने चैटबॉट के लिए इस [स्टार्टर कोड](#) को रीमिक्स करें।  
स्टार्टर कोड में नेविगेट करने के बाद, आपको सबसे पहले प्रॉजेक्ट को **रीमिक्स** करना चाहिए, या इसकी प्रति बनानी चाहिए। ट्रिंकेट विंडो के बाईं ओर **रीमिक्स** बटन पर क्लिक करें।



### कदम 3: ट्रिन्केट को आरंभ करें जारी

#### 3. ट्रिन्केट इंटरफ़ेस का अन्वेषण करें।

यदि आप ट्रिन्केट में नए हैं, तो ट्रिन्केट इंटरफ़ेस का पता लगाने के लिए कुछ मिनट का समय लें। अपने कोड को सहेजने और चलाने के तरीके से परिचित हों।



### कदम 4: स्टार्टर प्रॉजेक्ट का अन्वेषण करें (2 मिनट)

आओ [स्टार्टर फ़ाइल](#) पर एक नज़र डालें। यह कोड क्या करता है यह देखने के लिए रन `_` बटन को दबाएं।

आप देखेंगे कि वहाँ एक छोटा सा परिचय होगा जहाँ कार्यक्रम कहता है :

हैलो! मेरे चैटबॉट में आपका स्वागत है।

इसके बाद एक प्रश्न है:

क्या आप इस कार्यक्रम के साथ संवाद करने के लिए उत्साहित हैं?

“हाँ” टाइप करने की कोशिश करें और प्रतिक्रिया देखें, फिर “नहीं” टाइप करें और फिर “हाँ” टाइप करें।

आप देखेंगे कि इन तीनों उत्तरों की कार्यक्रम में अलग-अलग प्रतिक्रिया है। चूंकि पाइथन एक टेक्स्ट-आधारित भाषा है, इसमें वर्तनी और अक्षरों के केस दोनों मायने रखते हैं! उत्तर “हाँ” और “हॉ” को कंप्यूटर पर अलग-अलग तरीके से पढ़ा जाता है।

चैटबॉट के सबसे महत्वपूर्ण पहलू (1) उपयोगकर्ता से प्रश्न का उत्तर देने के लिए कहने और (2) उपयोगकर्ता से प्रतिक्रियाएँ प्रिंट करके बात करने की योग्यता हैं। हम सबसे पहले सीखेंगे कि कोड के माध्यम से उपयोगकर्ता से कैसे बात करनी है और फिर सवाल पूछने और जवाब देने के बारे में चर्चा करेंगे।

## कदम 5: परिचय शामिल करें (4 मिनट)

कंप्यूटर द्वारा उपयोगकर्ता से “बातें” करने के लिए हम कोड में **प्रिंट** स्टेटमेंट लिखते हैं और फिर यह ट्रिगेट विंडो के बाईं ओर स्टेटमेंट दिखाता है।

आइए **स्टार्टर फ़ाइल** में कोड की पहली पंक्ति पर एक नज़र डालें। यह कोड क्या करता है यह देखने के लिए रन **\_\_** बटन दबाएँ।

ध्यान दें कि कोड की पहली पंक्ति यह **प्रिंट** (“हेलो! मेरे चैटबॉट में आपका स्वागत है”) पढ़ती है और **परिणाम** स्क्रीन की पहली पंक्ति **हेलो है! मेरे चैटबॉट में आपका स्वागत है।**

आओ **प्रिंट** स्टेटमेंट का उपयोग करने के लिए मुख्य प्रतीकों और शब्दों को देखें:

**प्रिंट** (“नमूना टेक्स्ट”)

- **प्रिंट**: यह कीवर्ड कंप्यूटर को परिणाम स्क्रीन पर कुछ प्रिंट करने की सुविधा देता है
- **()**: कोष्ठक, कंप्यूटर को यह जानने की अनुमति देते हैं कि कोष्ठक के अंदर जो कुछ भी है उसे प्रिंट करना है।
- **“ ”**: दोहरे उद्धरण चिह्नों से कंप्यूटर को पता चल जाता है कि अंदर जो कुछ भी है वे शब्द हैं।
- **नमूना टेक्स्ट**: हम उद्धरण चिह्नों के अंदर किसी भी शब्द को शामिल कर सकते हैं और इसे ठीक उसी तरह प्रिंट किया जाएगा जैसा कि **परिणाम** स्क्रीन पर होता है।

### इसे स्वयं आजमाएं!

अपने चैटबॉट का संक्षिप्त परिचय **प्रिंट** करने के लिए कोड की एक पंक्ति जोड़ें। आप इस विवरण में विषय और लक्ष्य को शामिल कर सकते हैं।

रन **\_\_** बटन दबाकर यह देखें कि क्या आपका कंप्यूटर आपके परिचय को प्रिंट करता है।

### डीबगिंग करने के लिए सुझाव:

- वाक्यविन्यास त्रुटि: जाँच करें कि आपके प्रश्न के आरंभ और अंत दोनों में आपके प्रश्न के आसपास दोहरे-उद्धरण चिह्न हैं।
- वाक्यविन्यास त्रुटि: जाँच करें कि आपके प्रश्न के आसपास कोष्ठक हैं और कोड की पंक्ति एक समापन कोष्ठक के साथ **समाप्त** होती है।

## कदम 6: अपना पहला प्रश्न पूछें (5 मिनट)

अपना पहला प्रश्न पूछने के लिए, दो महत्वपूर्ण कंप्यूटर विज्ञान शब्दों पर चर्चा करें: **इनपुट** और **आउटपुट**।

**इनपुट** वह है जब कंप्यूटर को किसी प्रकार की जानकारी दी जाती है। कभी-कभी यह एक कुंजी को नीचे की तरफ दबाना हो सकता है, किसी डिवाइस में प्लगिंग करना जैसे कि माउस में, या हमारे मामले में, प्रतिक्रिया को टाइप करना हो सकता है। **आउटपुट** वह है जब कंप्यूटर से किसी अन्य प्रोसेस में जानकारी दी जाती है। उपयोगकर्ता की प्रतिक्रिया ("हां" या "नहीं") **इनपुट** है, जबकि कंप्यूटर की प्रतिक्रिया सवाल पूछने की प्रक्रिया का **आउटपुट** है।



पाइथन में प्रश्न की इनपुट प्राप्त करने के लिए हम कमांड `इनपुट()` का इस्तेमाल करते हैं।

ध्यान दें कि दूसरी कोड की पंक्ति `उत्तर = इनपुट("क्या आप इस कार्यक्रम के साथ संवाद करने के लिए उत्साहित हैं?")` परिणाम स्क्रीन की पहली पंक्ति `क्या आप इस कार्यक्रम के साथ संवाद करने के लिए उत्साहित हैं?` पढ़ती है।

आइए **इनपुट** स्टेटमेंट का उपयोग करने के लिए प्रमुख प्रतीकों और शब्दों को देखें:

`उत्तर = इनपुट("नमूना प्रश्न ")`

- **उत्तर**: यह एक **वेरिएबल** है जो उपयोगकर्ता की प्रतिक्रिया को संग्रहीत करता है। विशेष रूप से, इस वेरिएबल का नाम "उत्तर" है।
- **=**: यह बराबर चिह्न **वेरिएबल** उत्तर को मान की असाइनमेंट या रीअसाइनमेंट को दर्शाता है।
- **इनपुट**: यह कीवर्ड कंप्यूटर को उपयोगकर्ता से सवाल पूछने की सुविधा देता है
- **()**: कोष्ठक कंप्यूटर को कोष्ठक के अंदर वाले टेक्स्ट को प्रिंट करने के बारे में परिचित करवाते हैं।
- **" "**: दोहरे-उद्धरण चिह्नों से कंप्यूटर को पता चलता है कि अंदर जो भी कुछ है वे शब्द हैं और इनपुट संदेश का हिस्सा हैं।
- **नमूना प्रश्न**: हम उद्धरण चिह्नों के अंदर किसी भी शब्द को शामिल कर सकते हैं और यह प्रश्न ठीक उसी तरह प्रिंट किया जाएगा जैसा कि **परिणाम** स्क्रीन पर होता है।

**वेरिएबल** एक कंप्यूटर विज्ञान का शब्द है जिसका उपयोग कंप्यूटर प्रोग्राम में जानकारी (डेटा) को स्टोर करने के लिए किया जाता है। वेरिएबल को नाम दिया जाता है ताकि उन्हें आसानी से संदर्भित किया जा सके और एक कार्यक्रम में बदला जा सके।

## कदम 6: अपना पहला प्रश्न पूछें जारी

### इसे स्वयं आजमाएं!

अपने योजनाबंदी कार्यपत्रक को देखें और उपयोगकर्ता से अपना पहला प्रश्न पूछें। एक नई कोड की पंक्ति शामिल करें जो नीचे दिए गए प्रारूप के समान हो।

उत्तर = इनपुट("प्रश्न#1")

प्रश्न#1 को अपने वास्तविक प्रश्न के स्थान पर लिखें।

रन    बटन दबाकर यह देखें कि क्या कंप्यूटर आपका पहला प्रश्न पूछता है।

### डीबगिंग करने के लिए सुझाव:

- वाक्यविन्यास त्रुटि: जाँच करें कि आपके प्रश्न के आरंभ और अंत दोनों में आपके प्रश्न के आसपास दोहरे-उद्धरण चिह्न लगे हैं।
- वाक्यविन्यास त्रुटि: जाँच करें कि आपने प्रश्न के आसपास कोष्ठक लगाए हैं और कोड की पंक्ति समापन कोष्ठक के साथ समाप्त होती है।
- वाक्यविन्यास त्रुटि: जाँचें कि वेरिएबल नाम का उत्तर लिखने के बाद आपने बराबर का चिह्न लगाया है।

## कदम 7: किसी उत्तर पर प्रतिक्रिया दें (10 मिनट)

आपने देखा होगा कि कंप्यूटर प्रश्न पूछता है, लेकिन एक बार आपके द्वारा जवाब टाइप करने के बाद प्रतिक्रिया के लिए कुछ और नहीं करता है। ऐसा इसलिए है क्योंकि हमने कंप्यूटर को उस डेटा के साथ कुछ नहीं करने के लिए कहा है!

कोड में टाइपिंग करते हुए: उत्तर = इनपुट("प्रश्न#1") उपयोक्ता की प्रतिक्रिया को वेरिएबल उत्तर में स्टोर किया गया है। यदि उपयोगकर्ता ने "हां" या "नहीं" में उत्तर दिया तो हम इसे ऐसे निर्धारित करेंगे।

सबसे पहले, हमें संभावित विकल्पों ("हाँ" या "नहीं") की तुलना करने के लिए सशर्त स्टेटमेंट का उपयोग करना चाहिए। सशर्त स्टेटमेंट यह देखता है कि क्या नियमों (या स्टेटमेंट) के सेट को पूरा किया गया है और फिर यह तय करता है कि यदि नियम या स्टेटमेंट सही या गलत है तो कौन से काय को निष्पादित किया जाता है। हमारे प्रश्न के लिए तीन संभावित विकल्प हैं "हाँ", "नहीं", या कोई अन्य प्रतिक्रिया।

जब हम पाइथन में शर्त का उपयोग करते हैं तो हमें इन कीवर्ड्स का उपयोग करना चाहिए यदि, और अगर, और/या नहीं तो। आइए इस पर एक नज़र डालें कि प्रत्येक संभावित उत्तर के लिए उपयुक्त प्रतिक्रिया निर्धारित करने के लिए चैटबॉट में सशत का उपयोग कैसे किया जाता है।

## कदम 7: उत्तर पर प्रतिक्रिया दें जारी

उत्तर = इनपुट("क्या आप इस कार्यक्रम में संवाद करने के लिए उत्साहित हैं?")

यदि (उत्तर == "हाँ"):

प्रिंट ("मैं भी बहुत उत्साहित हूँ!")

और अगर (उत्तर == "नहीं"):

प्रिंट ("आहा, खैर मुझे आशा है कि मैं आपका मन बदल सकता हूँ!")

नहीं तो:

प्रिंट ("हम्म.. मुझे लगता है कि मैं आपके जवाब को समझ नहीं पा रही/रही।")

- **यदि:** यदि वाले स्टेटमेंट को इंगित करने के लिए शब्द
- **और अगर:** और-अगर वाले स्टेटमेंट को इंगित करने के लिए कीवर्ड यह एक वैकल्पिक स्टेटमेंट है, लेकिन इसे यदि वाले स्टेटमेंट के बाद आना चाहिए।
- **नहीं तो:** नहीं तो वाले स्टेटमेंट को इंगित करने के लिए कीवर्ड यह एक वैकल्पिक स्टेटमेंट है, लेकिन यह यदि और और-अगर वाले स्टेटमेंट के बाद आना चाहिए।
- **( ):** कोष्ठक वैकल्पिक हैं। वे हमारे कोड को व्यवस्थित रखने और पढ़ने में आसान बनाने में मदद करने के लिए शर्त के चारों ओर जोड़े जाते हैं।
- **उत्तर:** यह एक **वेरिएबल** है जो उपयोगकर्ता की प्रतिक्रिया को स्टोर करती है। विशेष रूप से, यह वेरिएबल नामित उत्तर है।
- **==:** बराबर का चिन्ह एक तुलनचिह्न है। इसका उपयोग किसी अन्य मान के चर उत्तर की तुलना करने के लिए किया जाता है, हमारे मामले में "हाँ" या "नहीं"
- **"हाँ"/"नहीं":** दोहरे-उद्धरण चिह्नों का उपयोग कंप्यूटर को यह बताने के लिए किया जाता है कि वह टेक्स्ट के रूप में अंदर का मान पढ़े। चूंकि हम इन शब्द प्रतिक्रियाओं के उत्तर की तुलना करना चाहते हैं इसलिए हम हाँ और नहीं के आसपास दोहरे उद्धरण चिह्नों का उपयोग करते हैं।
- **:** कोलन प्रोग्राम को सशर्त स्टेटमेंट के अंत की जानकारी देता है।
- **इंडेंट:** सभी कोड की पंक्तियाँ जिन्हें शर्त के पूरा होने पर कार्यान्वित किया जाना चाहिए, यदि वाले स्टेटमेंट के बाद इंडेंट किया जाना चाहिए। इससे कंप्यूटर को पता चल जाता है कि कौन सी कोड की पंक्तियाँ चलानी चाहिए। ऊपर का उदाहरण देखें जहाँ प्रिंट कमांड इंडेंट है।

याद रखें कि हम अपने प्रश्न के उपयोगकर्ता के उत्तर को **संग्रहित** करने के लिए **उत्तर** वेरिएबल का उपयोग करते हैं।

पहला **यदि** वाला स्टेटमेंट तुलना करता है यदि वेरिएबल उत्तर **"हाँ"** है। यदि जवाब **"हाँ"** है, तो **यदि** वाले स्टेटमेंट के नीचे की इंडेंटिड कोड की पंक्ति प्रिंट की जाती है। यदि **उत्तर "हाँ"** नहीं है, तो हम अगले सशर्त स्टेटमेंट **और-अगर** पर जाएंगे।

फिर यह **"नहीं"** के प्रति जवाब की तुलना करता है। इसी तरह, यदि **उत्तर "नहीं"** है, तो **और-अगर** वाले स्टेटमेंट के नीचे इंडेंटिड कोड की पंक्ति को प्रिंट किया जाता है। अंतिम **नहीं तो** वाला स्टेटमेंट उन सभी अन्य प्रकार की प्रतिक्रियाओं के लिए कैच ऑल है जो **उत्तर** हो सकता है देते हैं। चूंकि हम हाँ या नहीं के अलावा किसी अन्य प्रतिक्रिया की उम्मीद नहीं कर रहे हैं, फिर कंप्यूटर उपयोगकर्ता को यह बताने के लिए इंडेंट किए गए स्टेटमेंट को प्रिंट करता है कि यह एक अमान्य उत्तर था।

## कदम 7: जारी

### इसे स्वयं आजमाएं!

अपने योजनाबद्ध कार्यपत्रक को देखें और अपनी प्रतिक्रियाओं की जाँच करें कि उपयोगकर्ता ने "हां", "नहीं" में उत्तर दिया है, या अमान्य प्रतिक्रिया दी है।

कोड जो आपसे पहला प्रश्न पूछता है, उसके बाद उसी फॉर्मेट का अनुसरण करती हुई इन कोड की पंक्तियों को शामिल करें। प्रिंट स्टेटमेंटों को **इंडेंट** करना सुनिश्चित करें, लेकिन **यदि**, **और-अगर**, और **नहीं तो** वाले स्टेटमेंटों के बाद।

```
उत्तर = इनपुट("प्रश्न#1 ")
```

```
यदि (उत्तर == "हाँ"):
```

```
    प्रिंट ("हाँ प्रतिक्रिया")
```

```
और-अगर (उत्तर == "नहीं"):
```

```
    प्रिंट ("कोई प्रतिक्रिया नहीं")
```

```
नहीं तो:
```

```
    प्रिंट ("अन्य प्रतिक्रिया")
```

अपने योजनाबद्ध दस्तावेज़ में लिखी गई प्रतिक्रियाओं के साथ प्रत्येक विकल्प के लिए प्रिंट स्टेटमेंटों में प्रतिक्रियाओं को अपडेट करें

रन    बटन को दबाकर यह पता लगाएँ कि क्या कंप्यूटर आपके उत्तरों के अनुसार प्रतिक्रिया करता है।



## कदम 8: डेटा को एकत्र करने के लिए अधिक प्रश्न पूछना (10-15 मिनट)

अब जब आपने एक प्रश्न लिख लिया है, तो अपने शेष प्रश्नों को पूछने के लिए कदम 5 और 6 में दिए गए निर्देशों का पालन करें। आपके प्रोग्राम को नीचे जैसे समान प्रारूप का पालन करना चाहिए:

```
उत्तर = इनपुट("प्रश्न#1")
```

```
यदि (उत्तर == "हाँ"):
```

```
    प्रिंट ("हाँ प्रतिक्रिया")
```

```
और-अगर (उत्तर == "नहीं"):
```

```
    प्रिंट ("नहीं प्रतिक्रिया")
```

```
नहीं तो:
```

```
    प्रिंट ("अन्य प्रतिक्रिया")
```

```
उत्तर = इनपुट("प्रश्न#2")
```

```
यदि (उत्तर == "हाँ"):
```

```
    प्रिंट ("हाँ प्रतिक्रिया")
```

```
और-अगर (उत्तर == "नहीं"):
```

```
    प्रिंट ("कोई प्रतिक्रिया नहीं")
```

```
नहीं तो:
```

```
    प्रिंट ("अन्य प्रतिक्रिया")
```

```
उत्तर = इनपुट("प्रश्न#3")
```

```
यदि (उत्तर == "हाँ"):
```

```
    प्रिंट ("हाँ प्रतिक्रिया")
```

```
और-अगर (उत्तर == "नहीं"):
```

```
    प्रिंट ("नहीं प्रतिक्रिया")
```

```
नहीं तो:
```

```
    प्रिंट ("अन्य प्रतिक्रिया")
```

एक प्रश्न की लिखाई समाप्त करने के बाद आप यह सुनिश्चित करने के लिए अपने प्रोग्राम की जाँच करना न भूलें कि यह सही ढंग से चलता है।

### डीबगिंग करने के लिए सुझाव:

- वाक्यविन्यास त्रुटि: जाँच करें कि आपके प्रश्न के आरंभ और अंत दोनों में आपके प्रश्न के आसपास दोहरे-उद्धरण चिह्न हैं।
- वाक्यविन्यास त्रुटि: जाँच करें कि आपने प्रश्न के आसपास कोष्ठक लगाए हैं और कोड की पंक्ति एक समापन कोष्ठक के साथ समाप्त होती है।
- वाक्यविन्यास त्रुटि: जाँचें कि वेरिएबल नाम का उत्तर लिखने के बाद आपने बराबर का चिह्न लगाया है।
- वाक्यविन्यास त्रुटि: जाँच करें कि यदि, और-अगर, और नहीं तो वाले स्टेटमेंट के बाद प्रिंट स्टेटमेंट्स इंडेंट किए गए हैं।



## कदम 9: आपके चैटबॉट के लिए विस्तार (15-30 मिनट)

अपने चैटबॉट प्रॉजेक्ट को अगले स्तर तक ले जाने के लिए बहुत सारे तरीके हैं!

- "हां" या "नहीं" के अलावा अन्य प्रतिक्रियाओं वाले प्रश्न को शामिल करें। (5-8 मिनट)

हम किसी भी चीज़ की जाँच के लिए सशर्त विवरण लिख सकते हैं! इस समय हमारे सशर्त विवरण केवल "हां" या "नहीं" के उपयोगकर्ता के उत्तर की तुलना करते हैं लेकिन इसे आसानी से बदला जा सकता है। अगर हम हमारे संभावित उत्तरों को "सही" या "गलत" में बदलना चाहते हैं, तो हम केवल उत्तर == "True" पढ़ने के लिए कोष्ठकों के अंदर की शर्त को अपडेट करते हैं।

पिछला कोड	अद्यतित कोड
उत्तर = इनपुट("प्रश्न") यदि (उत्तर == "हाँ"): प्रिंट ("हाँ प्रतिक्रिया") और-अगर (उत्तर == "नहीं"): प्रिंट ("नहीं प्रतिक्रिया") नहीं तो: प्रिंट ("अन्य प्रतिक्रिया")	उत्तर = इनपुट("प्रश्न") यदि (उत्तर == "सही"): प्रिंट ("सही प्रतिक्रिया") और-अगर (उत्तर == "गलत"): प्रिंट ("गलत प्रतिक्रिया") नहीं तो: प्रिंट ("अन्य प्रतिक्रिया")

योजना बनाएं और एक और प्रश्न शामिल करें जिसमें "हां" या "नहीं" के अलावा अलग-अलग उत्तर हों।

- दो से अधिक संभावित प्रतिक्रियाओं वाले प्रश्न शामिल करें। (8-10 मिनट)

हमने केवल दो प्रतिक्रियाओं (हां या नहीं) वाले प्रश्नों पर विचार किया है लेकिन क्या होगा यदि हम एक सवाल पूछना चाहते हैं जिसमें बहुत सी प्रतिक्रियाएं हो सकती हैं? यदि हम एक और प्रतिक्रिया को कैप्चर करना चाहते हैं तो हमें एक और सशर्त स्टेटमेंट को शामिल करने की आवश्यकता है। याद रखें कि सशर्त स्टेटमेंटों का क्रम यदि, और-अगर, नहीं तो था। यदि हम एक और संभावित प्रतिक्रिया शामिल करना चाहते हैं तो हम दूसरे स्टेटमेंट से पहले एक अतिरिक्त और-अगर स्टेटमेंट को शामिल करते हैं।

चूंकि 'नहीं तो' सभी संभावित उत्तरों के लिए एक कैच है, इसलिए हम नहीं तो वाले स्टेटमेंट तक पहुंचने से पहले हमारे तीसरे विकल्प को कैप्चर करना सुनिश्चित करना चाहते हैं। उदाहरण के लिए, यदि हम प्रतिक्रिया "शायद" शामिल करना चाहते हैं तो हम नीचे दिए गए उदाहरण के अनुसार कोड को अपडेट करेंगे।

पिछला कोड	अद्यतित कोड
उत्तर = इनपुट("प्रश्न") यदि (उत्तर == "हाँ"): प्रिंट ("हाँ प्रतिक्रिया") और-अगर (उत्तर == "नहीं"): प्रिंट ("नहीं प्रतिक्रिया")  नहीं तो: प्रिंट ("अन्य प्रतिक्रिया")	उत्तर = इनपुट("प्रश्न") यदि (उत्तर == "हाँ"): प्रिंट ("हाँ प्रतिक्रिया") और-अगर (उत्तर == "नहीं"): प्रिंट ("नहीं प्रतिक्रिया") और-अगर (उत्तर == "शायद"): प्रिंट ("शायद प्रतिक्रिया") नहीं तो: प्रिंट ("अन्य प्रतिक्रिया")

## कदम 9: विस्तार जारी

- प्रतिक्रियाओं के लिए केस संवेदनशीलता को हटाएँ। (5-8 मिनट)

यह सुनिश्चित करके थक गए हैं कि आपने सभी उत्तर लोअरकेस में टाइप किए हैं? इसे ठीक करने का एक आसान तरीका है! इसके लिए हम कमांड लोअर() का प्रयोग करेंगे जो एक शब्द को सभी लोअरकेस में परिवर्तित करता है। यदि कोई शब्द पहले से ही लोअरकेस में है तो यह कमांड कुछ नहीं करती और शब्द को वैसा ही रखती है। यदि शब्द में मिश्रित केस हैं तो सभी अक्षरों को लोअरकेस में बदल दिया जाएगा (उदाहरण GWC → gwc)

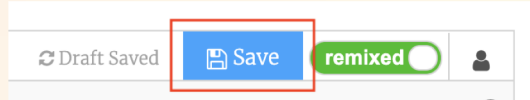
कमांड लोअर() का उपयोग करने के लिए, हम नई कोड की पंक्ति शामिल कर सकते हैं जो लोअरकेस संस्करण के लिए वेरिएबल उत्तर को निर्दिष्ट करता है जैसे:

पिछला कोड	अद्यतित कोड
<pre>उत्तर = इनपुट("प्रश्न")  यदि (उत्तर == "हाँ"):     प्रिंट ("हाँ प्रतिक्रिया") और-अगर (उत्तर == "नहीं"):     प्रिंट ("नहीं प्रतिक्रिया") नहीं तो:     प्रिंट ("अन्य प्रतिक्रिया")</pre>	<pre>उत्तर = इनपुट("प्रश्न") उत्तर = उत्तर.lower() यदि (उत्तर == "हाँ"):     प्रिंट ("हाँ प्रतिक्रिया") और-अगर (उत्तर == "नहीं"):     प्रिंट ("कोई प्रतिक्रिया नहीं") नहीं तो:     प्रिंट ("अन्य प्रतिक्रिया")</pre>

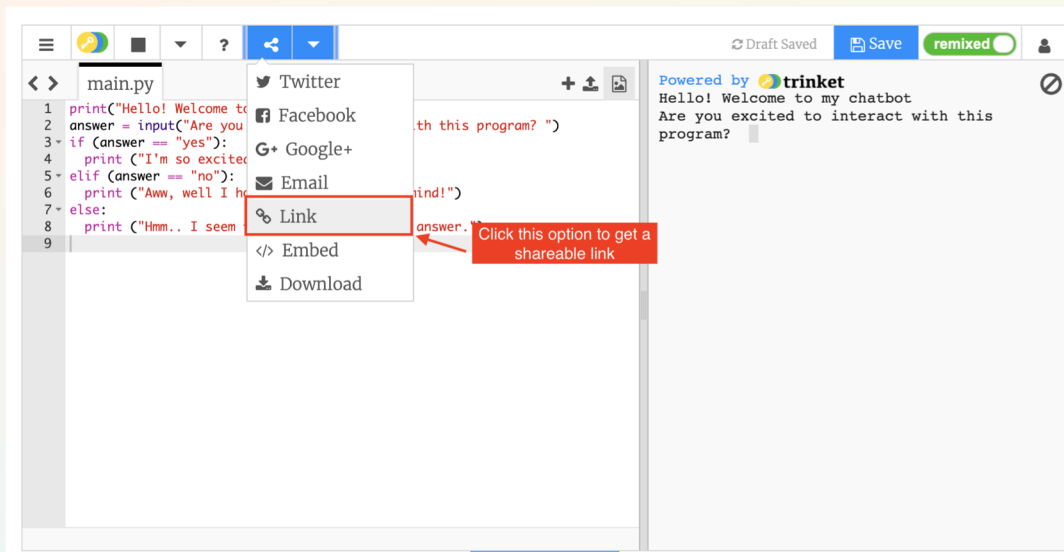
हमें हर प्रश्न के पूछे जाने के बाद इस नई पंक्ति को शामिल करना चाहिए।

## कदम 10: अपने गर्ल्ज हू कोड एट होम प्रॉजेक्ट को साझा करें! (5 मिनट)

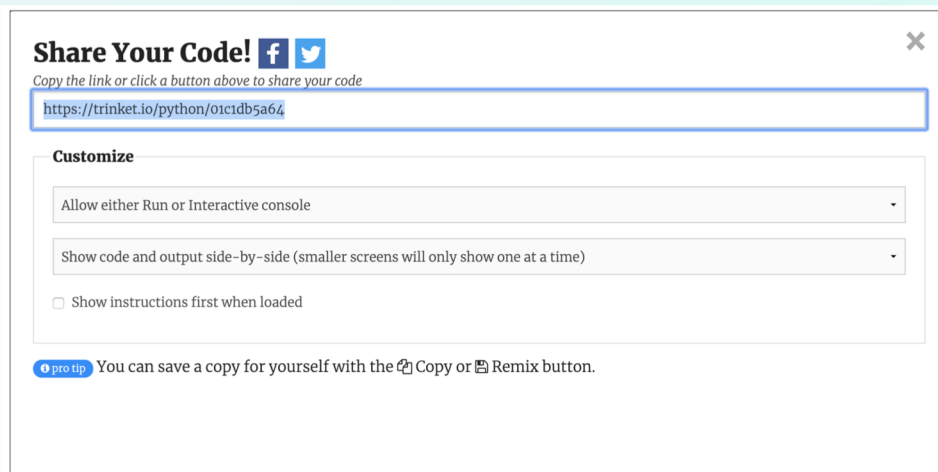
- ट्रिंकेट विंडो के दाईं ओर सेव बटन पर क्लिक करके आप अपने काम को सहेजना न भूलें



- अपने प्रॉजेक्ट को मित्रों और परिवार के साथ साझा करने के लिए बाईं ओर स्थित शेयर आइकन पर क्लिक करें और ड्रॉप डाउन मेनू में से लिंक विकल्प चुनें।



- अपने वर्किंग चैटबॉट की एक तस्वीर या वीडियो साझा करें या लिंक को कॉपी करें और अपने चैटबॉट को सोशल मीडिया पर साझा करें! @Girlswhode को टैग करना न भूलें और हैशटैग #codefromhome का उपयोग करें और हम आपको हमारे खाते में फीचर भी कर सकते हैं!



## क्या मैं आपकी मदद कर सकता हूँ? प्रॉजेक्ट प्लानिंग वर्कशीट

### प्रॉजेक्ट अवलोकन योजनाबंदी

**थीम:** वह कौन सा विषय है जो आपके चैटबॉट से कवर किया जाएगा?

**दर्शक:** यह चैटबॉट किसे सर्व करेगा? लोगों का कौन सा समूह आपके उत्पाद में रुचि रखेगा?

**लक्ष्य:** आप अपने चैटबॉट से क्या पूरा करना चाहते हैं? आपके दर्शकों के लिए यह रुचिकर क्यों है?

## प्रश्न योजना

अपने उपयोगकर्ताओं से पूछने के लिए तीन हां या नहीं वाले कोई प्रश्न चुनें। प्रतिक्रिया अमान्य थी। पहली पंक्ति में प्रश्न लिखें और "हां" या "नहीं" के लिए प्रत्येक पंक्ति में आपकी चैटबॉट की प्रतिक्रिया। पाइथन जवाब स्वीकार करने में बहुत चालाक है, अंतिम विकल्प "अन्य प्रतिक्रियाएं" हां या नहीं के अलावा अन्य जवाब देने के लिए है। आप एक संदेश शामिल कर सकते हैं जिससे उपयोगकर्ता को पता चले कि उनकी प्रतिक्रिया अमान्य थी।

प्रश्न #1:	
हाँ	
नहीं	
अन्य प्रतिक्रिया	

## प्रश्न योजना जारी

प्रश्न #2:	
हाँ	
नहीं	
अन्य प्रतिक्रिया	

प्रश्न #3:	
हाँ	
नहीं	
अन्य प्रतिक्रिया	