



# Girls Who Code At Home

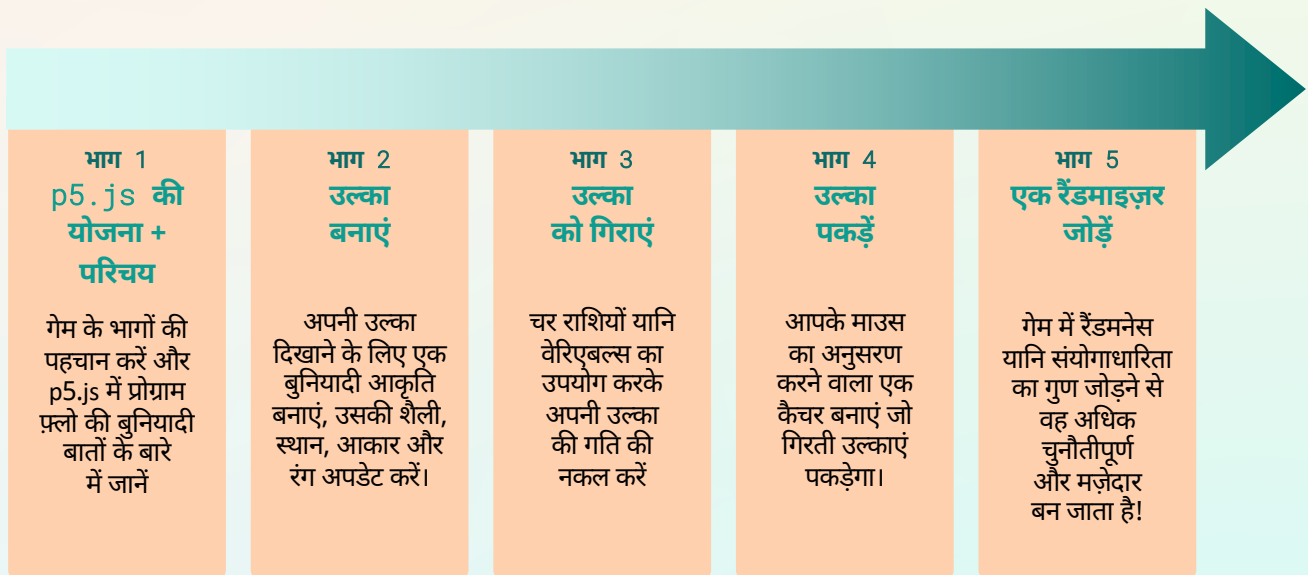
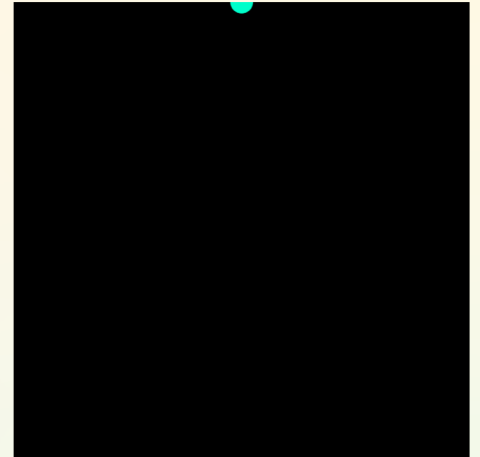
उल्का पकड़ो गेम: भाग 2

उल्का बनाएं

## गतिविधि अवलोकन

पुनः स्वागत है! पिछले सप्ताह आपने p5.js परिवेश को जाना था और अपने उल्का पकड़ो गेम के भागों की योजना बनाना आरंभ किया था। इस भाग में, आप निर्देशांक प्रणाली (कोऑर्डिनेट सिस्टम) का उपयोग करके अपने गेम का पहला घटक बनाएंगे जो है: उल्का! आप p5.js में रंगों का उपयोग करने और अपनी उल्का व रेखा-चित्र के बैकग्राउंड का रंग तय करने के बारे में भी और जानेंगे।

यह गतिविधि आरंभ करने से पहले यह आवश्यक है कि आपने **उल्का पकड़ो गेम सीरीज़** का **भाग 1** पूरा कर लिया हो।



## सीखने के लक्ष्य

इस गतिविधि को पूरा कर लेने पर आप निम्नलिखित कर सकेंगे...

- ❑ p5.js कोऑर्डिनेट सिस्टम और स्क्रीन पर मौजूद पिक्सल्स से उसके संबंध का वर्णन करना।
- ❑ कोऑर्डिनेट प्लेन (निर्देशांक समतल) पर बुनियादी आकृतियां बनाने के लिए बिल्ट-इन फंक्शन्स और कमांड्स का उपयोग करना।

## सामग्रियां

- [p5.js ऑनलाइन एडिटर](#)
- [उल्का पकड़ो गेम सैंपल प्रोजेक्ट](#)
- [उल्का पकड़ो गेम भाग 2 संदर्भ पुस्तिका](#)

## वुमन इन टेक स्पोर्टलाइट: फीनिक्स पेरी



तस्वीर स्रोत: [Hackaday.io](https://hackaday.io)

फीनिक्स के बचपन में उनके माता-पिता उनके लिए एक [Atari](#) होम गेम कंसोल ले आए थे। टेक्नॉलजी की इस नई दुनिया से परिचय होने पर फीनिक्स के मन में टेक्नॉलजी के प्रति रुचि जागी, और उन्होंने [BASIC](#) में प्रोग्रामिंग सीखना शुरू कर दिया। अंडरग्रेजुएट डिग्री पूरी करने के बाद, फीनिक्स ने [एवाइट \(Evite\)](#) के लिए वेब डिज़ाइनर के रूप में काम करना शुरू कर दिया। यहां उन्होंने कई रातें जाग-जाग कर गुजारीं, जो कंपनी में एक सामान्य व्यवहार था, और आराम न मिलने के कारण कलाइयों में होने वाले खिंचाव को दर्दनाशी दवाओं के सहारे कुछ समय संभाला। पर फिर वे [कार्पल टनल](#) सिंड्रोम का शिकार हो गईं; हथेली की हड्डियों और स्नायुओं पर तनाव बढ़ने के कारण यह सिंड्रोम होता है जो प्रोग्रामर्स में आम है।

फीनिक्स ने कला निर्देशक के रूप में कार्य करते हुए अपनी स्थिति के कारण टेक्नॉलजी उद्योग से दूर कई वर्ष गुजारे। उसके बाद वे [NYU टैंडन के इंटीग्रेटेड डिजिटल मीडिया प्रोग्राम](#) के संपर्क में आईं, जहां उन्होंने सहायक प्राध्यापक और शोधकर्ता के रूप में गेम विकास की डिज़ाइन, खेल, और अभिव्यक्ति के बारे में पढ़ाया। अपने अनुभव के माध्यम से वे [कोड लिबरेशन \(Code Liberation\)](#) की सह-संस्थापक बनीं। कोड लिबरेशन, स्टीम (STEAM) में नौकरी पाने के लिए महिलाओं, अन्य लिंग वाले व्यक्तियों, फ्रैम (लेस्बियन जोड़े में स्वयं को महिला मानने वाली स्त्री), और स्वयं को महिला मानने वाले व्यक्तियों को पढ़ाने, तैयार करने और उनकी सहायता करने की आशा रखता है। वहां सभी आयु की, और अपने-अपने करियर के अलग-अलग स्टेज में मौजूद महिलाओं को निःशुल्क कक्षाएं, कार्यशालाएं, गेम जैम्स, हैकेथॉन, और सोशल गेम नाइट्स प्रदान की जाती हैं। फीनिक्स महिलाओं की अगली पीढ़ी को टेक्नॉलजी उद्योग से जुड़ने हेतु आवश्यक कौशल विकसित करने में मदद देने के लिए अपने अनुभव का उपयोग करती हैं। अब उन्होंने यूनाइटेड किंगडम में कोड लिबरेशन (Code Liberation) की शाखा खोलने के लिए [यूनिवर्सिटी ऑफ़ आर्ट्स, लंडन](#) के साथ साझेदारी की है।

फीनिक्स के बारे में, और टेक्नॉलजी में विविधता क्यों ज़रूरी है इस बारे में और जानने के लिए, यह [वीडियो](#) देखें। फीनिक्स के बारे में अधिक जानना चाहती हैं? फीनिक्स की अकादमिक यात्रा के बारे में, और आज वे जहां हैं वहां तक पहुंचने में आई बाधाओं के बारे में और जानने के लिए उनकी [TED टॉक](#) देखें, उनकी [व्यक्तिगत वेबसाइट](#) देखें, और यह [लेख](#) पढ़ें!

## 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 फीनिक्स पेरी 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।



वीरता

कार्पल टनल का शिकार होने के बाद फीनिक्स काम करने और चलने-फिरने में असमर्थ हो गईं। इससे टेक्नॉलजी उद्योग में उनकी वापसी कठिन हो गई। चर्चा करें कि कैसे फीनिक्स ने उद्योग में लौटने में बहादुरी दिखाई, और साथ-ही-साथ कोड लिबरेशन (Code Liberation) में अन्य महिलाओं की मदद भी की और उन्हें तैयार भी किया।

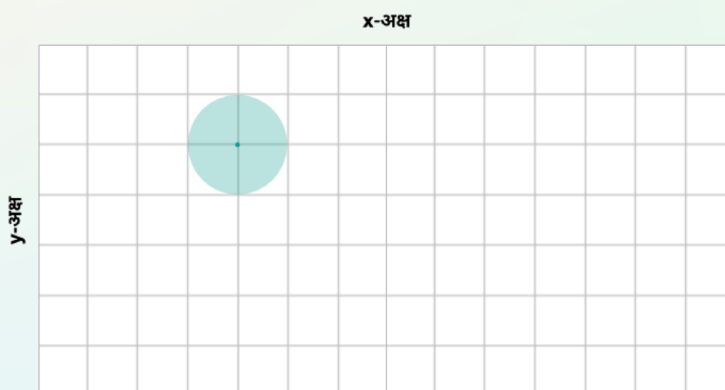
परिवार के किसी सदस्य या मित्र के साथ अपनी प्रतिक्रियाएँ साझा करें। चर्चा में शामिल होने हेतु दूसरों को फीनिक्स के बारे में अधिक पढ़ने के लिए प्रोत्साहित करें!

## चरण 1: गोले को अपनी अंगुलियों पर नचाएं (2-4 मिनट)

मान लें कि आपका कोई दोस्त आपको किसी कागज़ पर एक गोला बनाने को कहता है। आप या तो फटाफट गोला बना सकते हैं, या फिर उससे और जानकारी मांग सकते हैं। यदि आप और जानकारी मांगते हैं, तो हो सकता है कि आप पूछें - कागज़ पर कहां? कितना बड़ा? किस रंग का? बिल्कुल ठीक-ठीक गोल या थोड़ा अंडाकार भी चलेगा? कहां वाले प्रश्न के उत्तर में, आपका दोस्त यह कह सकता है कि, “बायीं ओर से कुल चौड़ाई के एक-तिहाई पर, और ऊपर से नीचे जाते हुए कुल ऊंचाई के तीन-चौथाई पर।” यदि आप किसी मनुष्य से बात कर रहे हैं और आपको विशिष्ट होने की ज़रूरत नहीं है तो इस तरह के निर्देश से काम चल जाएगा। पर कंप्यूटर के मामले में यह निर्देश काम नहीं करेगा! कंप्यूटर के मामले में हमें कोऑर्डिनेट सिस्टम यानि निर्देशांक प्रणाली का उपयोग करके उन घटकों के लिए एक स्थान निर्दिष्ट करना होगा जिन्हें हम हमारे प्रोग्राम में दिखाना चाहते हैं।

कोऑर्डिनेट सिस्टम एक ऐसा सिस्टम है जो एक या अधिक संख्याओं का उपयोग करके दिक्-स्थान में किसी स्थान की पहचान करता है। कोऑर्डिनेट सिस्टम 2D (द्विविमीय) समतल पर या 3D (त्रिविमीय) दिक्-स्थान में हो सकते हैं। कोऑर्डिनेट समतलों (यानि 2D) में एक x-अक्ष होता है जो क्षैतिज होता है, और एक y-अक्ष होता है जो ऊर्ध्वाधर होता है, और ये दोनों अक्ष मिलकर एक ग्रिड बनाते हैं। किसी बिंदु विशेष का स्थान बताने के लिए इसमें एक क्रमबद्ध युग्म यानि जोड़े का उपयोग किया जाता है: (x स्थान, y स्थान)।

p5.js z-अक्ष वाले त्रिविमीय स्थान में भी काम कर सकती है। संभव है कि आपको p5 के दस्तावेज़ों में कुछ फंक्शन्स में z-अक्ष के पैरामीटर का विकल्प मिल जाए, पर आपको उसे शामिल नहीं करना चाहिए।



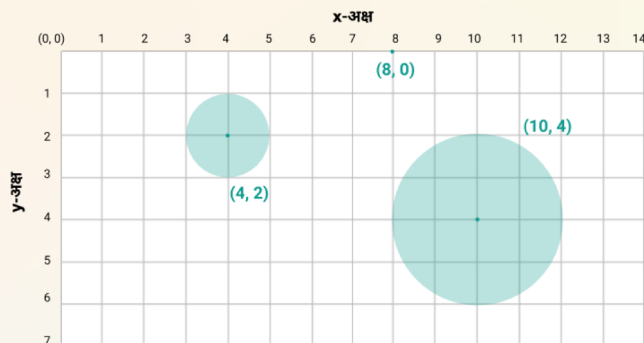
ऊपर वाले कोऑर्डिनेट समतल पर विचार करें। सोचें कि उस स्थान पर गोला बनाने के लिए आप कंप्यूटर को क्या निर्देश देंगे।



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

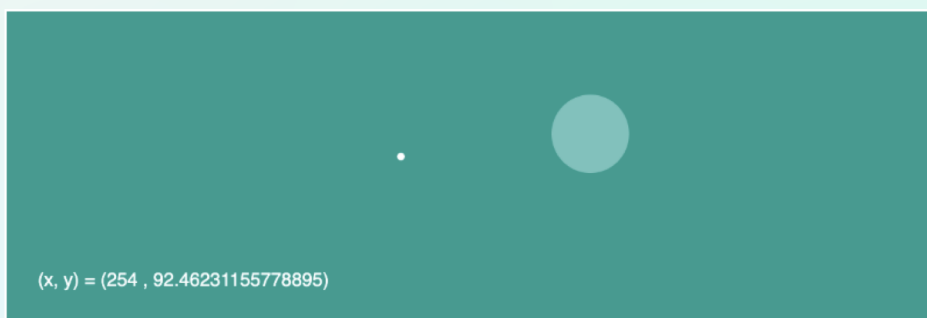
## चरण 2: कोऑर्डिनेट सिस्टम का उपयोग करके अभ्यास करें (5-8 मिनट)

आपकी स्क्रीन पर मौजूद हर पिक्सल का कोऑर्डिनेट सिस्टम में एक अद्वितीय पता होता है। स्क्रीन पर पिक्सल बनाने के लिए हमें हमारे प्रोग्राम को पिक्सल का  $x$  कोऑर्डिनेट (यानि  $x$ -अक्ष पर स्थान) और  $y$  कोऑर्डिनेट (यानि  $y$ -अक्ष पर स्थान) बताना होगा।



स्क्रीन के कोऑर्डिनेट सिस्टम में मूल बिंदु यानि  $(0, 0)$ , ऊपरी बायें कोने पर स्थित होता है। स्क्रीन पर दायीं ओर बढ़ने पर  $x$ -कोऑर्डिनेट की कीमत बढ़ती है। स्क्रीन पर नीचे की ओर जाने पर  $y$ -कोऑर्डिनेट की कीमत बढ़ती है। गणित की कक्षाओं में आपने जो कोऑर्डिनेट सिस्टम का लेआउट (निर्देशांक प्रणाली का खाका) देखा होगा उससे यह लेआउट थोड़ा अलग है, क्योंकि गणित वाले लेआउट में मूल बिंदु केंद्र में या निचले बायें कोने पर होता है।

इस [रेखा-चित्र](#) को देखें और अपने माउस को इधर-उधर फिरा कर गोले के केंद्र, चौड़ाई और ऊंचाई का अनुमान लगाने की कोशिश करें।



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

## चरण 3: अपना प्रॉजेक्ट का रेखा-चित्र बनाएं (5-10 मिनट)

बाकी के अनुभागों में, हम आपके गेम के लिए कोड लिखना शुरू करेंगे। सबसे पहले हमें आपका मुख्य प्रोजेक्ट का रेखा-चित्र बनाना होगा।

- ❑ **p5.js ऑनलाइन एडिटर में लॉग इन करें।** एडिटर अपने-आप आपको एक खाली रेखा-चित्र और शुरुआती कोड दे देता है। *वैकल्पिक रूप से, आप File > New में जाकर नया रेखा-चित्र बना सकते हैं।*
- ❑ **पेंसिल आइकन पर क्लिक करके रेखा-चित्र को कोई ऐसा नाम दें जिसे आप आसानी से पहचान सकते हों, जैसे Meteor Catcher Game v1 (उल्का पकड़ो गेम, संस्करण 1)। ध्यान दें: यह टूलबार के नीचे मौजूद रेखा-चित्र की जानकारी वाले स्थान में होता है।**
- ❑ **अब, File में जाकर Save पर क्लिक करें।** आप Command S (Mac) या Control S (Windows) कुंजीपटल (कीबोर्ड) शॉर्टकट का उपयोग करके भी अपना रेखा-चित्र सहेज सकते हैं। इन शॉर्टकट्स का उपयोग करने से पहले सुनिश्चित कर लें कि आप एडिटर के अंदर नेविगेट कर रहे हों (सिस्टम फ़ोकस एडिटर के अंदर हो)।
- ❑ **अपने रेखा-चित्र में सबसे ऊपर, निम्नलिखित जानकारी वाला एक मल्टीलाइन कमेंट बनाएं:**
  - ❑ **कार्यक्रम का शीर्षक:** रेखा-चित्र का जो नाम है, वही यह शीर्षक भी होना चाहिए।
  - ❑ **प्रोग्राम का संस्करण:** यह पहला संस्करण है, या दूसरा? यदि आप बड़े बदलाव करते हैं, तो नया संस्करण बनाना अच्छा रहता है।
  - ❑ **लेखक:** रचयिता (आपका पहला नाम और अंतिम नाम का प्रथमाक्षर)।
  - ❑ **विवरण:** इस बारे में एक-दो वाक्य कि यह क्या करता है।



अपने रेखा-चित्र में सबसे ऊपर आप एक कमेंट लिखेंगे जो रेखा-चित्र की बुनियादी जानकारी देगा। **कोड कमेंट्स** का उपयोग इसलिए करें: खुद को याद दिलाना कि कोई चीज़ कैसे काम करती है, किसी निर्णय के पीछे का कारण, या कोई फ़ॉलो-अप कार्य।

- सिंगल लाइन कमेंट्स के लिए डबल फ़ॉर्वर्ड स्लैश, `//` का उपयोग किया जाता है।
- मल्टीलाइन कमेंट्स के मामले में, कमेंट शुरू करने के लिए एक फ़ॉर्वर्ड स्लैश और एक एस्तेरिस्क, `/*`, लिखें, और उसे बंद करने के लिए एक एस्तेरिस्क और एक फ़ॉर्वर्ड स्लैश, `*/`, लिखें।

```
// यह एकल-पंक्ति टिप्पणी है
```

```
/*  
यह बहुपंक्ति टिप्पणी है  
*/
```

## चरण 4: उल्का बनाएं (3-5 मिनट)

अब जबकि हमने हमारा प्रोजेक्ट रेखा-चित्र सहेज लिया है, चलिए हमारे गेम का पहला घटक बनाते हैं, यानि उल्का! सबसे पहले हम यह जानेंगे कि किसी स्थान विशेष पर आकृति कैसे चित्रित करें, और फिर उसका रंग-रूप बदलने के लिए हम उसमें एक रंग भरेंगे।

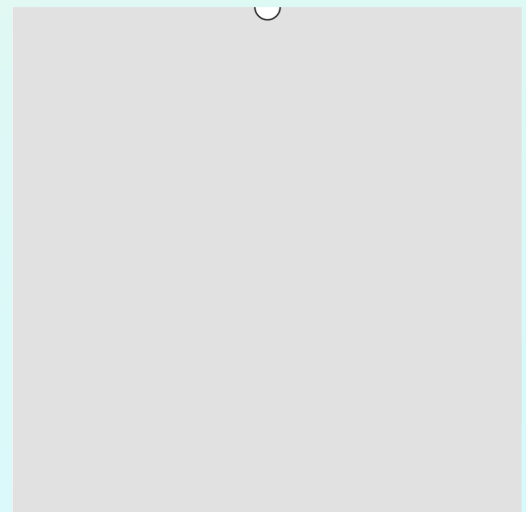
### आकृति और स्थान

फिर से वह पिछला चरण याद करें जहां हमने कोऑर्डिनेट सिस्टम के बारे में जाना था। अब हम जो जानते हैं उसके आधार पर, यदि हम कोई आकृति बनाना चाहते हैं तो हमें कंप्यूटर को अलग-अलग हर पिक्सल बनाने को कहना होगा। साफ है कि यह बहुत थकाऊ काम है, इसलिए p5 ने पहले से फंक्शन बना दिए जो हमारे लिए आकृतियों का चित्रण करते हैं। हमें बस इतना करना है कि वह स्थान बताना है जहां हमें आकृति चाहिए, और फिर उसकी चौड़ाई व ऊंचाई बतानी है। आइए गोले के मामले में फंक्शन के वाक्य-विन्यास को समझें:

| जावास्क्रिप्ट                            | विवरण   |
|--|---|
| <pre>ellipse(x, y, width, height);</pre> | <ul style="list-style-type: none"><li>→ <b>ellipse</b>: फंक्शन का नाम। दीर्घवृत्त (Ellipse), अंडाकार (oval) का ही दूसरा नाम है। <a href="#">p5.js संदर्भ</a></li><li>→ <b>()</b>: हम कोष्ठकों द्वारा हमारे प्रोग्राम को यह बताते हैं कि उसे इस फंक्शन को कॉल करना है। कभी-कभी हम हमारे कोष्ठकों के अंदर फंक्शन के पैरामीटर या इनपुट शामिल कर देते हैं।</li><li>→ <b>x</b>: दीर्घवृत्त के केंद्र का x-कोऑर्डिनेट।</li><li>→ <b>y</b>: दीर्घवृत्त के केंद्र का y-कोऑर्डिनेट।</li><li>→ <b>width</b>: दीर्घवृत्त की चौड़ाई तय करता है (पिक्सल में)।</li><li>→ <b>height</b>: दीर्घवृत्त की ऊंचाई तय करता है (पिक्सल में)।</li><li>→ <b>,</b>: हम फंक्शन्स में विभिन्न पैरामीटर्स या इनपुट्स को अलग करने के लिए कॉमा का उपयोग करते हैं।</li><li>→ <b>::</b> p5.js की सभी कोड की पंक्तियों का अंत सेमीकोलन से होना आवश्यक होता है।</li></ul> |

स्क्रीन पर उल्का बनाने के लिए `ellipse()` फंक्शन का उपयोग करें:

- ❑ `ellipse()` फंक्शन को अपने रेखा-चित्र में `draw()` फंक्शन के अंदर जोड़ें।
- ❑ इसे कैनवास के केंद्र (आपके x-अक्षीय स्थान) में सबसे ऊपर (आपके y-अक्षीय स्थान) रखें। यदि आपको रिफ्रेश की ज़रूरत हो तो ऊपर वाले ग्राफ की मदद लें।
- ❑ गोले या दीर्घवृत्त का आकार 20 पिक्सल चौड़ा और 20 पिक्सल ऊंचा पर सेट करें।
- ❑ खुद को यह याद दिलाने के लिए एक कमेंट जोड़ें कि यह उल्का है।
- ❑ प्ले बटन दबा कर अपना कोड चलाएं और उसे परखें।



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

## चरण 5: उल्का में रंग भरें (5-8 मिनट)

आकृतियों में रंग भरने के लिए हम `fill()` फंक्शन का उपयोग कर सकते हैं। यह, कमांड के बाद चित्रित हुई किसी भी आकृति में निर्धारित रंग भर देगा। नीचे वाले उदाहरण में, दोनों गोले नीले हैं क्योंकि वे `fill()` फंक्शन के बाद आते हैं:

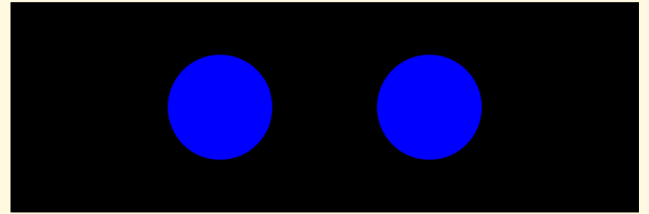
### जावास्क्रिप्ट

```
function setup() {
  createCanvas(600, 200);
}

function draw() {
  background(0);

  // दोनों घेरों को नीला कर दें
  fill(0, 0, 255);
  ellipse(width / 3, height / 2, 100, 100);
  ellipse(width / 3*2, height / 2, 100, 100);
}
```

### परिणाम



पर यदि हमें दूसरी आकृति में नीले की बजाय नील-हरित (teal) रंग भरना हो तो? हमें उस आकृति से पहले एक और `fill()` फंक्शन जोड़ना होगा:

### जावास्क्रिप्ट

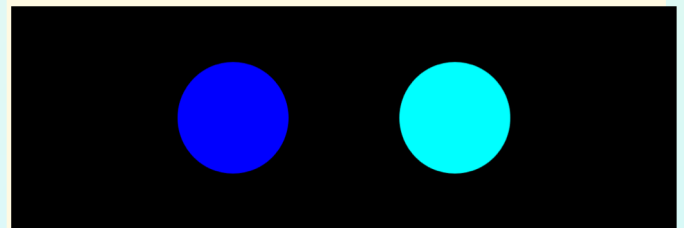
```
function setup() {
  createCanvas(600, 200);
}

function draw() {
  background(0);

  // नीला घेरा
  fill(0, 0, 255);
  ellipse(width / 3, height / 2, 100, 100);
  // चैती घेरा
  fill(0, 255, 255);
  ellipse(width / 3*2, height / 2, 100, 100);
}
```

### परिणाम

इस बदलाव से हमें एक नीला गोला और एक नील-हरित गोला मिलेगा



`fill()` मेथड RGB कलर मोड का उपयोग करता है; इस मोड में रेड, ग्रीन और ब्लू लाइट के संयोजन के उपयोग से तमाम रंग बनाए जाते हैं।

- `(0,0,255)` से नीला रंग दिखेगा
- `(0,255,255)` से नील-हरित (teal) रंग दिखेगा।



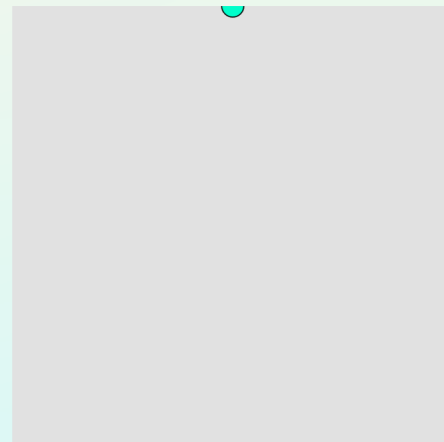
## चरण 5: उल्का में रंग भरें (जारी)

RGB कलर मोड का उपयोग करने वाले `fill()` फंक्शन का वाक्य-विन्यास नीचे देखें। RGB कलर मोड में रेड, ग्रीन और ब्लू लाइट के संयोजन के उपयोग से तमाम रंग बनाए जाते हैं। आप हर रंग को 0 से 255 के बीच की एक कीमत निर्धारित कर सकते हैं। उदाहरण के लिए, (255, 0, 0) लाल होगा, (0, 0, 0) काला होगा, और (255, 136, 0) नारंगी होगा। आप [कलर पिकर्स \(color pickers\)](#) जैसे टूल्स या [कूलर्स \(Coolers\)](#) जैसे पैलेट टूल्स का उपयोग करके विभिन्न कीमतें और रंग आजमा कर देख सकते हैं।

| जावास्क्रिप्ट                                       | विवरण  |
|---|--|
| <code>fill(redValue, greenValue, blueValue);</code> | <ul style="list-style-type: none"><li>→ <code>fill</code>: फंक्शन का नाम।</li><li>→ <code>()</code>: हम कोष्ठकों द्वारा हमारे प्रोग्राम को यह बताते हैं कि उसे इस फंक्शन को कॉल करना है। कभी-कभी हम हमारे कोष्ठकों के अंदर फंक्शन के पैरामीटर या इनपुट शामिल कर देते हैं।</li><li>→ <code>redValue</code>: 0 से 255 के बीच की लाल की कीमत।</li><li>→ <code>blueValue</code>: 0 से 255 के बीच की ब्लू की कीमत।</li><li>→ <code>greenValue</code>: 0 से 255 के बीच की ग्रीन की कीमत।</li><li>→ <code>;</code>: p5.js की सभी कोड की पंक्तियों का अंत सेमीकोलन से होना आवश्यक होता है।</li></ul> |

`fill()` फंक्शन का उपयोग करके अपनी उल्का में रंग भरें:

- ☐ अपनी उल्का के लिए एक रंग चुनें और RGB कीमतें नोट कर लें।
- ☐ `fill()` फंक्शन को अपनी उल्का की RGB कीमत के साथ कॉल करें (याद रखें कि स्थान महत्वपूर्ण है!)
- ☐ जब आप कार्य पूरा कर लें तो प्ले बटन दबा कर अपना प्रोग्राम चलाएं। आपकी उल्का का रंग बदल कर वह हो जाना चाहिए जो आपने निर्धारित किया था।



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

## चरण 6: रूपरेखा बदलें (3-5 मिनट)

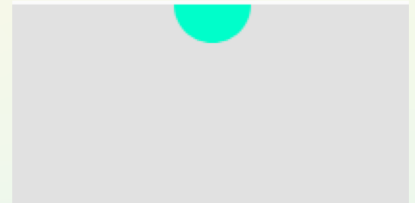
शायद आपका ध्यान इस बात पर गया होगा कि आपकी उल्का के चारों ओर एक काली रूपरेखा है। हम `noStroke()` फंक्शन का उपयोग करके यह रूपरेखा हटा सकते हैं। इस फंक्शन को कॉल करने का यह अर्थ है कि आपके रेखा-चित्र में किसी भी आकृति की रूपरेखा नहीं होगी। यदि आप रूपरेखाएं सक्षम करना चाहते हैं, तो आपको उस आकृति के ऊपर (यानी उस आकृति से पहले) `stroke()` फंक्शन को कॉल करना होगा।

आप `stroke()` और `strokeWeight()` जैसे अन्य फंक्शन्स का उपयोग करके आकृतियों की सीमाओं के रंग और उनकी मोटाइयों को नियंत्रित कर सकते हैं।

`noStroke()` फंक्शन का उपयोग करके रूपरेखा हटाएं:

- ❑ `noStroke()` फंक्शन को `background()` फंक्शन के नीचे जोड़ें। चूंकि हम हमारी किसी भी आकृति पर रूपरेखा नहीं चाहते हैं, अतः हम इसे सबसे ऊपर रखेंगे।
- ❑ कार्य पूरा कर लेने पर प्रोग्राम को चलाएं।

अब आपकी उल्का पर रूपरेखा नहीं दिखेगी। यह नज़दीकी स्क्रीनशॉट देखें, आपके रेखा-चित्र में सबसे ऊपर उल्का को ऐसा दिखना चाहिए।



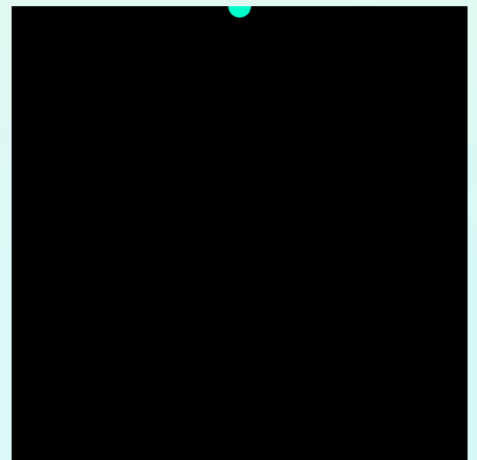
अपने कोड को पृष्ठ 4 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।

## चरण 7: पृष्ठभूमि के रंग को जोड़ें (3-5 मिनट)

अपने रंग ज्ञान को परखें! `background()` फंक्शन का उपयोग करके बैकग्राउंड को भरें। इस समय, इसके पास केवल एक कीमत है, पर `fill()` फंक्शन की भांति, यह RGB आदि अन्य कलर मोड के लिए पैरामीटर्स स्वीकार सकता है।

अपने गेम का बैकग्राउंड बदलें:

- ❑ एक बैकग्राउंड रंग चुनें और उसकी RGB वैल्यूज़ नोट कर लें। (ये रहे **कलर पिकर्स (color pickers)** और **कूलर्स (Coolers)** जैसे पैलेट टूल्स, जिनके बारे में हमने पहले बताया था।)
- ❑ `draw()` के अंदर `background()` फंक्शन में अपनी नई कलर वैल्यूज़ अपडेट करें।

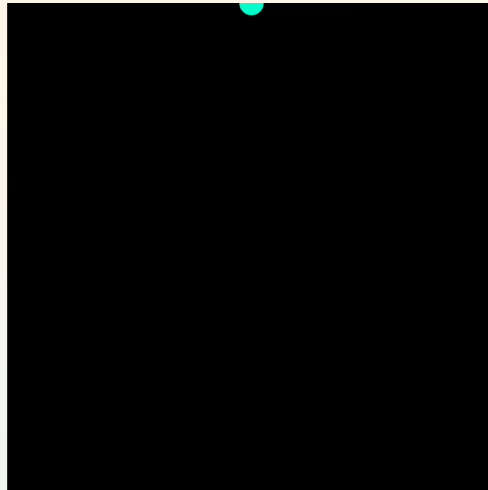


अपने कोड को पृष्ठ 4 पर संदर्भ मार्गदर्शिका से जांचना न भूलें।

## चरण 8: 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। प्ले बटन पर क्लिक करके अपना रेखा-चित्र चलाएं। आपके सामने होना चाहिए:

- रंगीन बैकग्राउंड वाला एक 400 गुणा 400 का कैनवास।
- कैनवास के ऊपरी मध्यबिंदु पर नए रंग और बिना किसी सीमा वाली एक गोल उल्का।



जैसा चाहा था वैसे कार्य नहीं कर रहा? यदि आपका सामना किसी त्रुटि से हो जो आपके कोड को कंपाइल न होने दे और चलने न दे, तो p5.js कन्सोल में एक त्रुटि संदेश दिखाएगी। जब कोई चीज़ ठीक से कार्य न करे, तो समस्या का पता लगाने के लिए वहीं से शुरुआत करें।

या फिर, ये **डीबगिंग के सुझाव आजमाएं**:

- क्या आपका कोड सही मझले कोष्ठकों (कर्ली ब्रैकेट्स) के अंदर है?
- क्या आपने कोड की हर लाइन के अंत में सेमीकोलन टाइप किए हैं?
- क्या आपने चर राशीयों और फंक्शन के नामों की सही स्पेलिंग लिखी? याद रखें कि जावास्क्रिप्ट भी केस सेंसिटिव है यानि इसमें अपरकेस और लोअरकेस अक्षरों को समान नहीं माना जाता है!
- क्या आपके फंक्शन्स सही स्थान पर और सही क्रम में हैं? याद रखें कि प्रोग्राम फ़्लो में क्रम मायने रखता है!
- क्या आपके पैरामीटर्स की कीमतें, उनके फंक्शन की सही रेंज के अंदर हैं? उदाहरण के लिए, क्या x की कीमत 500 है, जबकि कैनवास की चौड़ाई 400 पिक्सल है? क्या आपकी RGB वैल्यूज़ 0 और 255 के बीच हैं?

यदि आपको डीबगिंग की सर्वोत्तम कार्यप्रथाओं के बारे में रीफ़्रेशर की ज़रूरत हो, तो p5.js समुदाय की यह [कमाल की पोस्ट](#) देखें।

## चरण 9: समझ की जाँच करें

नीचे कोड में आकृति के स्थान, आकार, और रंग का वर्णन करें:

```
function setup() {  
  createCanvas(100, 100);  
}  
  
function draw() {  
  fill(0, 0, 255);  
  ellipse(50, 50, 5, 5);  
}
```



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

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

हम आपके काम को देखना पसंद करेंगे और हम जानते हैं कि दूसरे भी ऐसा करेंगे। अपना स्यूडोकोड (छद्म कोड) हमसे साझा करें!

[@girlswhocode](#) [#codefromhome](#) को टैग करना मत भूलें, और हो सकता है कि हम आपको हमारे खाते में प्रदर्शित कर देंगे!

और Girls Who Code at Home परियोजनाओं के लिए बनी रहें!

