

پیشرو خلاقیت آزاد مهر ویرا

به نام آفریننده جهان هستی



آموزش محتوای برنامه نویسی مؤسسه پیشرو خلاقیت آزاد
مهر ویرا (شرکت دارای گواهی خلاق معاونت علمی ریاست
جمهوری)

راهنمای نرم افزار PISH_BOT 🚀

پیشرو خلاقیت آزاد مهر ویرا

❓ قدم اول: روشن کردن ربات

ابتدا ربات خود را روشن کنید.

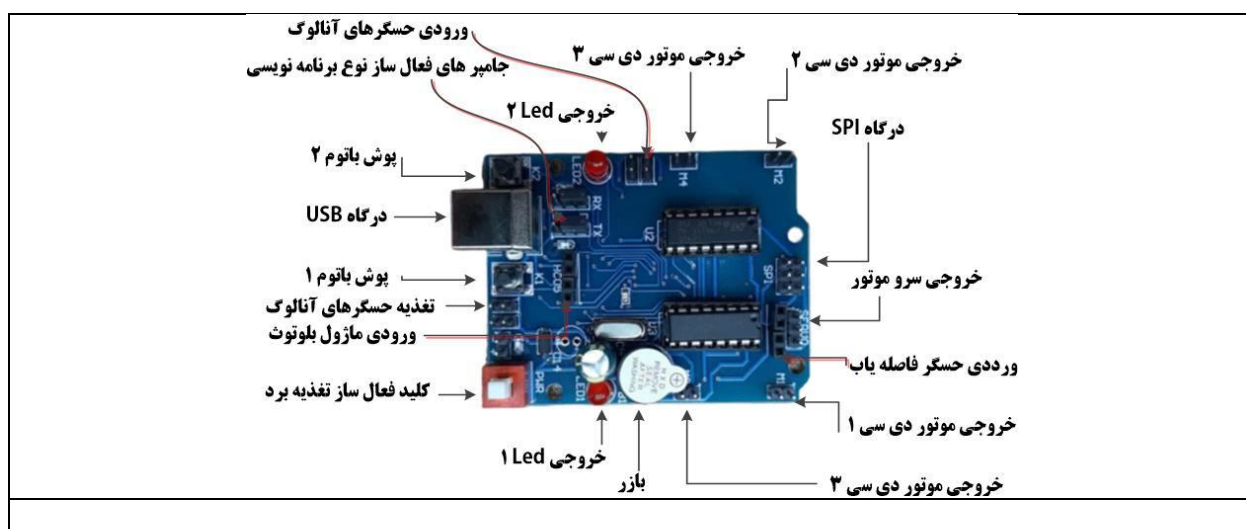
برای این کار باید منبع تغذیه (باتری) را به برد ربات وصل نمایید.

● سیم قرمز مربوط به سر مثبت (+) باتری است.

● سیم مشکی مربوط به سر منفی (-) باتری است.

با دقت به پلاریته (جهت مثبت و منفی) که روی برد مشخص شده، سیم‌ها را مطابق شکل متصل کنید.

✅ در صورت اتصال صحیح، چراغ چشمک‌زن روی برد روشن خواهد شد.



📱 قدم دوم: نحوه اتصال ربات به تلفن همراه

۱. ابتدا بلوتوث گوشی خود را روشن کنید ●

۲. در لیست دستگاه‌ها به دنبال نام ربات باشید. نام ربات‌ها معمولاً شامل عبارت‌هایی مثل

🔍 Vira هستند

۳. روی نام ربات مورد نظر کلیک کنید تا به آن متصل شوید 🍷

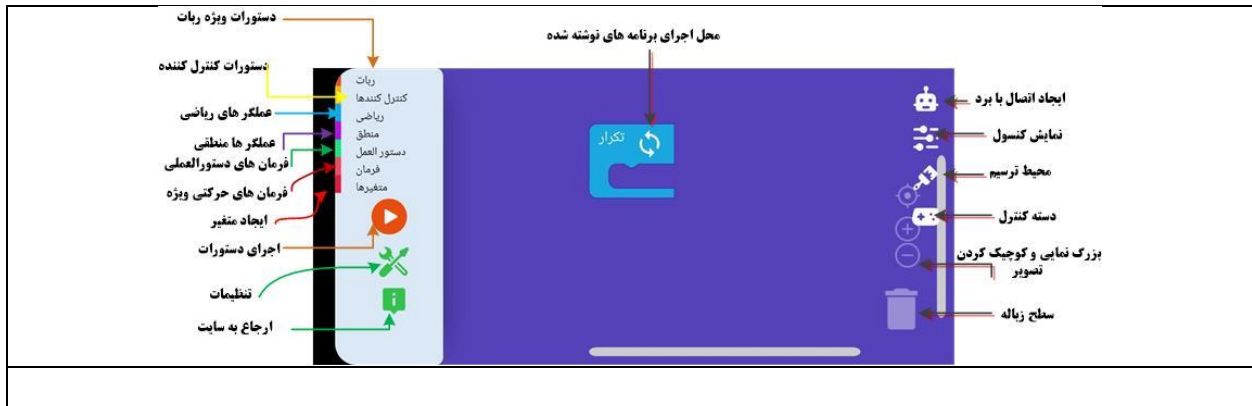
پیشرو خلاقیت آزاد مهر ویرا

۴. اگر گوشی درخواست رمز داشت، عدد جلوی نام ربات را وارد کنید.

مثال: اگر ربات با نام **Vira ۹۹۸۹** نمایش داده شد، رمز را همان **۹۹۸۹** وارد کنید

۵. بعد از اتصال موفق، وارد اپلیکیشن شوید

۶. در اپلیکیشن، در بخش مشخص شده، روی **نماد سر ربات** کلیک کنید تا اتصال کامل



اتصال بلوتوث در اپلیکیشن

در این بخش می‌توانید با انتخاب گزینه **۲. BLUETOOTH** گوشی خود را به ربات متصل کنید

توجه: ⚠

بلوتوث بعضی از گوشی‌ها ممکن است با ورژن متفاوتی باشند و به جای **۲. BLUETOOTH** گزینه **۴. BLUETOOTH** نمایش داده

شود. در این صورت همان گزینه را انتخاب کنید تا اتصال برقرار شود ✓

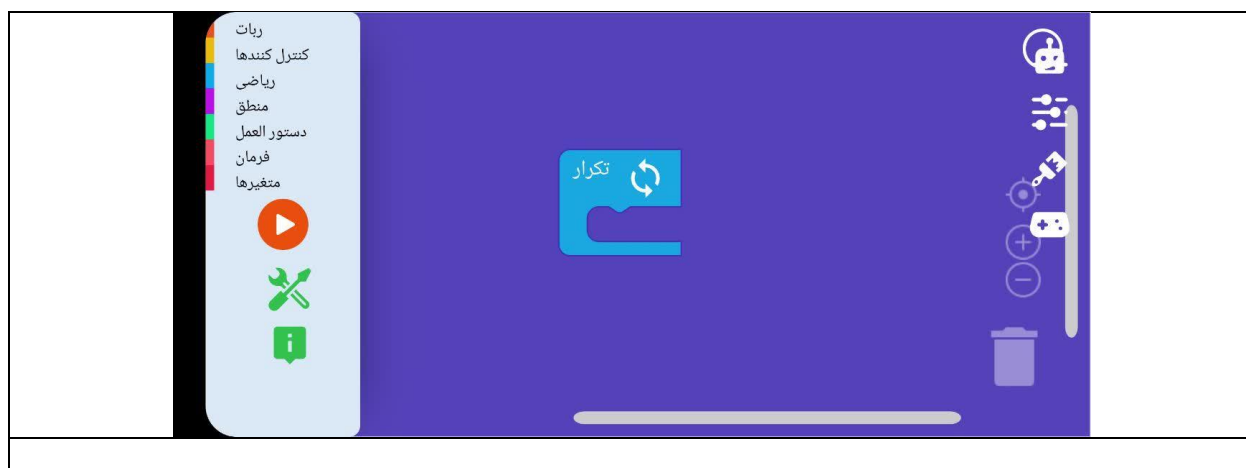


پیشرو خلاقیت آزاد مهر ویرا

⚠ نکته مهم:

اگر این روش‌ها جواب نداد، اتصال بلوتوث گوشی خود را بررسی کنید 🔍
همچنین می‌توانید از فضای برنامه خارج شده و دوباره روی اتصال کلیک کنید 🔄

❓ در صورت اتصال موفق، چراغ چشمک زن روی برد از حالت چشمک زن خارج شده و به حالت روشن ثابت درمی‌آید ✅💡
اکنون ربات شما آماده برنامه‌نویسی و دریافت دستورات است! 🤖🎉



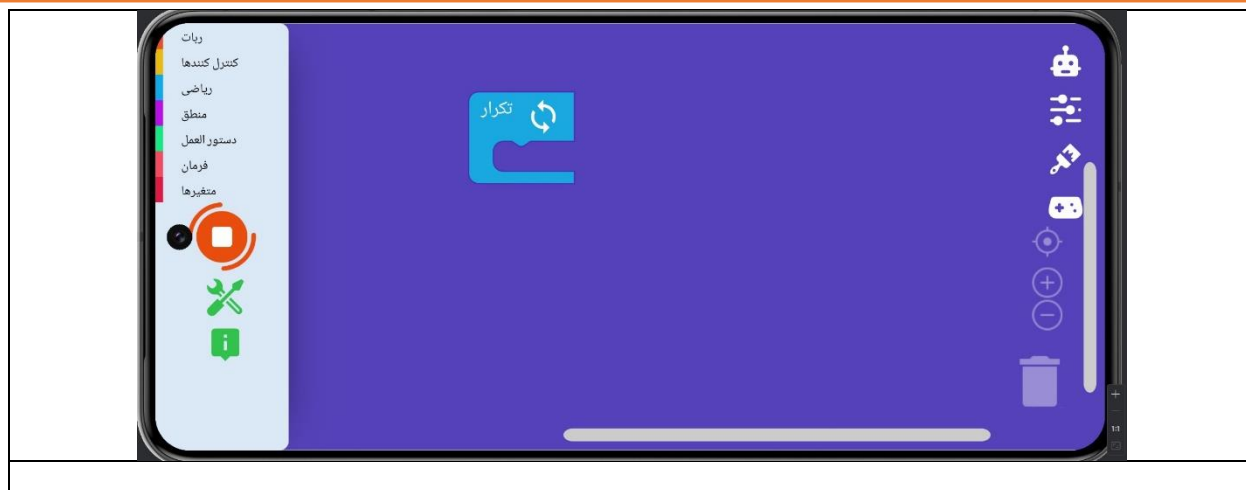
🚀 قدم سوم: دکمه اجرا

● یادتون باشه:

بعد از اینکه برنامه‌تون رو نوشتید، برای اجرای آن **حتما** باید دکمه دایره‌ای قرمز رنگ با مثلث درون آن را فشار دهید ▶
این دکمه در سمت چپ صفحه قرار دارد.

⌚ وقتی برنامه در حال اجرا است، شکل داخل دکمه از ▶ (مثلث) به ◼ (دایره) تغییر می‌کند.

پیشرو خلاقیت آزاد مهر ویرا



نماد سطل زباله

در این بخش، هر قسمتی از ۶ گروه سمت چپ که داخل محیط کار خود آوردید و دیگر نمی‌خواهید، کافی است آن را بکشید و داخل سطل زباله بندازید تا حذف شود. 🗑️✂️



در مراحل بعدی با نحوه برنامه‌نویسی ربات خود بیشتر آشنا می‌شوید. ✨🤖



پس لطفاً ناامید نشوید و آماده باشید برای تجربه‌های هیجان‌انگیزی که در انتظار شماست! 🚀🎉

قدم چهارم: آشنایی با بلوک تکرار



وقتی برنامه را باز می‌کنید، صفحه‌ای خالی دارید که فقط یک بلوک آبی به نام **تکرار** در آن دیده می‌شود.



این بلوک، مهم‌ترین جزء برنامه‌نویسی شماست! ✨🌱

پیشرو خلاقیت آزاد مهر ویرا

این بلوک مثل یک زمین خالی است که شما می‌توانید روی آن اختراعات و خلاقیت‌های خود را بسازید.



هر دستوری که می‌خواهید دوباره و دوباره اجرا شود، داخل این بلوک قرار می‌دهید.



یعنی ربات شما همیشه در حال انجام دستورات شما از اول است و مرتب آن‌ها را تکرار می‌کند.

✂️ قدم پنجم: آشنایی با گروه‌های ابزار برنامه‌نویسی

در سمت چپ فضای اپلیکیشن، شش گروه ابزار برای برنامه‌نویسی ربات خود خواهید دید:

۱ ربات

۲ کنترل کننده

۳ ریاضی $\div \times - +$

۴ منطق

۵ دستورالعمل

۶ فرمان

۷ متغیرها

این گروه‌ها به شما کمک می‌کنند تا دستورات و برنامه‌های خود را راحت‌تر بسازید و ربات را با

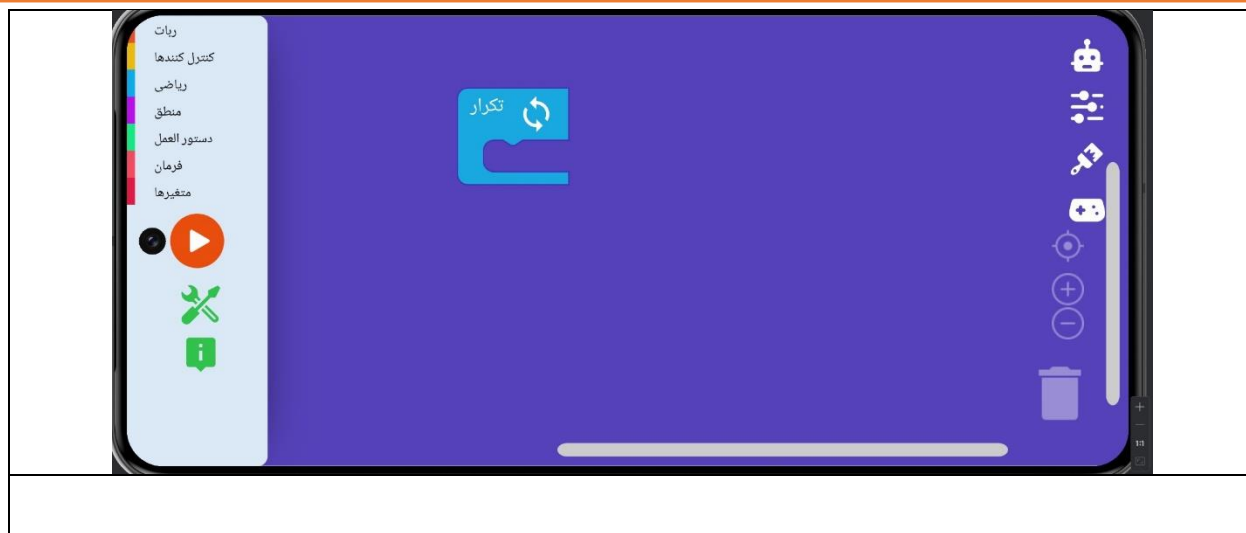
خلاقیت کامل کنترل کنید! ✨🎉

در ادامه هر یک از این بخش‌ها را جداگانه توضیح خواهیم داد و اجزای هر کدام را با مثال معرفی می‌کنیم.

🎯 مرحله به مرحله با ما پیش بروید و آماده شوید تا تبدیل به یک اپراتور و برنامه‌نویس حرفه‌ای

شوید! 🚀👤👤

پیشرو خلاقیت آزاد مهر ویرا



⚠ نکته مهم:

قبل از شروع برنامه‌نویسی، توجه به این نکته خیلی کمک می‌کند تا از سردرگمی نجات پیدا کنید!

در برنامه شما می‌توانید اعداد را بنویسید و مقایسه کنید، این اعداد به معنی خودشان هستند؛ اعدادی برای انجام عملیات ریاضی و مقایسه‌ها.

اما وقتی این اعداد در جلوی چراغ یا موتور نوشته می‌شوند، معنی متفاوتی پیدا می‌کنند!

💡 برای مثال، وقتی عدد ۱۰۰ برای چراغ تنظیم شود، یعنی روشنایی آن زیاد است، و عدد ۱۰ یعنی کم. عدد ۰ یعنی چراغ خاموش است.

در این نسخه فعلی برنامه، فقط روشن یا خاموش بودن موتور و چراغ مد نظر است، پس:

• عدد ۰ یعنی خاموش 🔴

• هر عدد غیر صفر یعنی روشن 🔵

✂ در آپدیت‌های آینده این موضوع با دقت و جزئیات بیشتر پیاده‌سازی خواهد شد.

پس اگر کمی گیج شدید، نگران نباشید! فقط کافی است به راهنمای همان بخش مراجعه کنید.

ما تمام تلاشمان را می‌کنیم تا هوای مخترعین و برنامه‌نویسان فوق‌العاده‌مان را داشته باشیم و این مسیر را برایتان لذت‌بخش کنیم!



ربات 🤖

در این بخش شما می‌توانید به قسمت‌های مختلف ربات خود فرمان دهید.

می‌توانید هر یک از موتورها، سروو موتور، چراغ‌ها 💡، بوق 🔊، حسگرها و سنسور فاصله را انتخاب کرده و

پیشرو خلاقیت آزاد مهر ویرا

کنترل کنید.

همه این‌ها در اختیار شماست تا رباتان را به بهترین شکل هدایت کنید! 



چرخاندن موتور

1  موتور مورد نظر رو انتخاب کن (M^۱ یا M^۲)

2  جهت چرخش رو مشخص کن:

 جلو (Forward)

 عقب (Backward)

3  قدرت موتور:

 خاموش = ۰

 عدد غیر صفر (مثل ۱ یا ۱۰۰) = روشن

تنظیم خروجی

در این قسمت می‌توانید خروجی‌هایی مثل چراغ‌ها و موتورهای شماره ۳ و ۴ را کنترل کنید.

◆ نکات مهم:

• عدد جلوی خروجی یعنی وضعیت روشن یا خاموش بودن

• عدد ۰ = خاموش 

پیشرو خلاقیت آزاد مهر ویرا

- هر عدد غیر از صفر = روشن 

⚠ در آپدیت‌های بعدی ممکن است این عدد برای موتور نشان‌دهنده قدرت هم باشد، ولی در حال حاضر برای همه خروجی‌ها، عدد صفر یعنی خاموش و هر عدد دیگر یعنی روشن.

نکته دوم چراغ 

برای کنترل چراغ در این بخش:

- عدد ۰ به معنای خاموش کردن چراغ 

- هر عدد دیگری به معنای روشن کردن چراغ 

پس هر وقت خواستی چراغ رو روشن کنی، فقط کافیه عددی غیر از صفر وارد کنی، و برای خاموش کردن، عدد صفر رو وارد کن. خیلی ساده! 

نکته سوم موتور 

ممکنه براتون سوال شده باشه چرا موتورهای M۳ و M۴ تو این بخش جدا تعریف شدن و قبلاً استفاده نکردیم؟

- این موتورها یه روش خاص‌تر برای کنترل دارن.
- وقتی یه موتور به پین M۳ یا M۴ وصل می‌کنید، می‌تونید با این دستورها اونا رو کنترل کنید.
- برای حرکت دادن موتور، اول دستور روشن کردن رو برای M۳ یا M۴ بفرستید.
- بعد بلافاصله با یک بلوک دیگه، موتور مقابل رو خاموش کنید تا چرخش اتفاق بیفته.
- با روشن و خاموش کردن برعکس موتورهای M۳ و M۴ می‌تونید جهت چرخش رو تغییر بدید.

خیلی خفن و کاربردی!   

بوق 

(ولی لطفاً باهاش مردم آزاری نکنید! )

- تو این بخش می‌تونید بوق رو خاموش یا روشن کنید.
- تو برنامه‌های بعدی یاد می‌گیریم چطور بوق رو طوری تنظیم کنیم که وقتی ربات به مانع نزدیک شد بوق بزنه  یا مثلاً به صورت منقطع و قطع و وصل صدای بوق شنیده بشه.

یه ابزار خیلی جذاب برای هشدار و اطلاع‌رسانی ربات شما!  

پیشرو خلاقیت آزاد مهر ویرا

حسگرها و سنسور فاصله 📏👁️

این بخش‌ها مثل چشم و گوش ربات شما هستند! 👁️👂🤖

- شما می‌تونید داده‌ها و اطلاعات حسگرها رو دریافت کنید و باهاشون هر برنامه‌ای که می‌خواهید بنویسید.
- چراغ، موتور و بوق خروجی‌های ربات هستند که با دستور شما عمل می‌کنن.
- ولی حسگرها و سنسورها ورودی‌های ربات هستند؛ یعنی ربات با اون‌ها محیط رو می‌فهمه و مثل گوش و چشم به شما اطلاع میده.

• مثلاً می‌تونید بفهمید نور کجا بیشتر یا کمتره ☀️🌑 یا چقدر تا مانع بعدی فاصله دارید 🚧.

• با این اطلاعات می‌تونید برنامه‌هاتون رو هوشمندانه و مطابق شرایط بسازید! 🎯🌟

یه دنیای هیجان‌انگیز برای خلق برنامه‌های خلاقانه! 🚀

املاً درست گفتی! 💡

بخش **ریاضی** 📊 نقش کلیدی در برنامه‌نویسی ربات داره، چون با استفاده از عملیات ریاضی می‌تونیم منطق و کنترل دقیق‌تری روی ربات داشته باشیم.

وقتی بخش ریاضی رو خوب بفهمیم، می‌تونیم مثال‌های کاربردی و جامع‌تر برای بخش **کنترل‌کننده‌ها** 📦 ارائه کنیم که فهمشون خیلی ساده‌تر و جذاب‌تر بشه.

منتظر توضیح‌های بخش ریاضی هستیم تا با هم بریم سراغ کنترل‌کننده‌ها و برنامه‌های خفن بسازیم! 🚀🌟

🌟!عالیه

بخش **ریاضی** مثل جعبه ابزار جادویی شماست 🧙🌟 که باهاش می‌تونید

- $+$ $-$ $\times$ $\div$ عملیات پایه‌ای مثل جمع، تفریق، ضرب و تقسیم انجام بدید
- اعداد تصادفی 🎲 تولید کنید (که برای بازی‌ها یا کارهای خلاقانه خیلی کاربردی)
- محاسبات پیچیده‌تری مثل سینوس 📐 و توابع دیگه انجام بدید
- مقایسه عددها برای تصمیم‌گیری منطقی بکنید (مثلاً آیا عدد بزرگ‌تر است یا کوچک‌تر؟)

این بخش، پایه‌ی ساختن برنامه‌های هوشمند و منعطف برای ربات شماست 💡🤖

😊🚀 خیلی منتظر نمونه‌های جذاب و کاربردی در بخش کنترل‌کننده‌ها هستیم تا بیشتر باهاش آشنا بشیم

پیشرو خلاقیت آزاد مهر ویرا



در بخش **چهار عمل اصلی**، شما می‌تونید به راحتی عملیات ریاضی پایه رو روی دو عدد انجام بدید:

جمع $+$

منها $-$

ضرب $\times$

تقسیم $\div$

کافیه روی فلش کنار بخش کلیک کنید تا گزینه‌ها باز بشه و عمل مورد نظر رو انتخاب کنید.
مثلاً:

$$8 = 3 + 5$$

$$5 = 2 \div 10$$

این قابلیت خیلی برای مقایسه‌ها، محاسبات و شرط‌گذاری توی برنامه ربات مهمه!   آماده‌اید که با این ابزار جادویی برنامه‌های خلاقانه بسازیم؟ 

بسیار خوب!  

در بخش **مقایسه** می‌تونید بین دو مقدار — مثل اعداد، متغیرها یا حتی خروجی سنسورها — عملیات مقایسه انجام بدید.
با کلیک روی فلش کنار بخش، این چهار حالت رو خواهید دید:

مساوی $(=)$ 

نامساوی $(\neq)$ 

بزرگ‌تر $(>)$ 

کوچک‌تر $(<)$ 

پیشرو خلاقیت آزاد مهر ویرا

مثلاً می‌تونید بررسی کنید که آیا عدد سنسور برابر با مقدار خاصی هست یا نه، یا ببینید یک متغیر از متغیر دیگه بزرگ‌تر هست یا کوچکترا! این قابلیت پایه و اساس تصمیم‌گیری‌های هوشمندانه توی برنامه‌هاست. ⚡🤖

حاضرید با این ابزار به رباتتون فرمان‌های هوشمند بدید؟ 🚀🤖

عدد دلخواه

با گزینه **عدد دلخواه** می‌تونید هر عددی که می‌خواید رو وارد کنید و در برنامه‌هاتون استفاده کنید.

کافیه عدد رو انتخاب کنید، صفحه کلید عددی براتون باز میشه، عدد رو وارد کنید و OK رو بزنید.

اینطوری هر جا که نیاز داشتید می‌تونید اعداد دقیق و خاص خودتون رو به برنامه اضافه کنید! ✨🤖

مثلاً می‌تونید سرعت موتورها رو با عدد خاصی تنظیم کنید یا مقدار حساسیت سنسورها رو تعیین کنید. ⚙️🚗

بهم بگو دوست داری چطوری از این قابلیت استفاده کنی؟ 💡🤖

اعداد تصادفی

اگر می‌خواید برنامه‌تون کمی هیجان داشته باشه و ربات رفتارهای غیرقابل پیش‌بینی انجام بده، می‌تونید از بخش **اعداد تصادفی** (رندوم) استفاده کنید.

مثلاً ربات می‌تونه هر بار یک عدد تصادفی بین ۱ تا ۱۰۰ انتخاب کنه و بر اساس اون سرعت حرکتش یا روشنایی چراغ‌هاش رو تغییر بده.

اینطوری برنامه‌تون پویا و جذاب‌تر میشه! ✨🚀

آیا دوست داری یک مثال برنامه رندوم برات آماده کنم؟ 🎉🤖

سینوس

بخش **سینوس (sin)** برای زمانی کاربرد داره که بخوایم حرکت یا تغییرات ربات رو به صورت نوسانی یا دوره‌ای داشته باشیم.

مثلاً می‌تونید سرعت موتور یا شدت نور LED رو به صورت نوسانی تغییر بدید تا ربات رفتار طبیعی‌تر و جالب‌تری داشته باشه.



مثال ساده:

اگر بخواید مقدار سینوس زاویه‌ای را محاسبه کنید و بر اساس اون چراغی روشن و خاموش شود یا موتور کمی با نوسان بچرخد، از این تابع استفاده می‌کنید.

پیشرو خلاقیت آزاد مهر ویرا

میخواهی یک مثال برنامه با سینوس برات آماده کنم؟ 🤖😊

کنترل کننده ها

بخش کنترل کننده ها واقعاً قلب تپنده‌ی برنامه نویسی رباته! 🤖❤️

اگر بلوک تکرار مثل زمین بازی باشه، کنترل کننده ها اسکلت و ساختار این زمین هستن که دستورات رو زمان بندی و مدیریت می کنن. ⚙️🔴

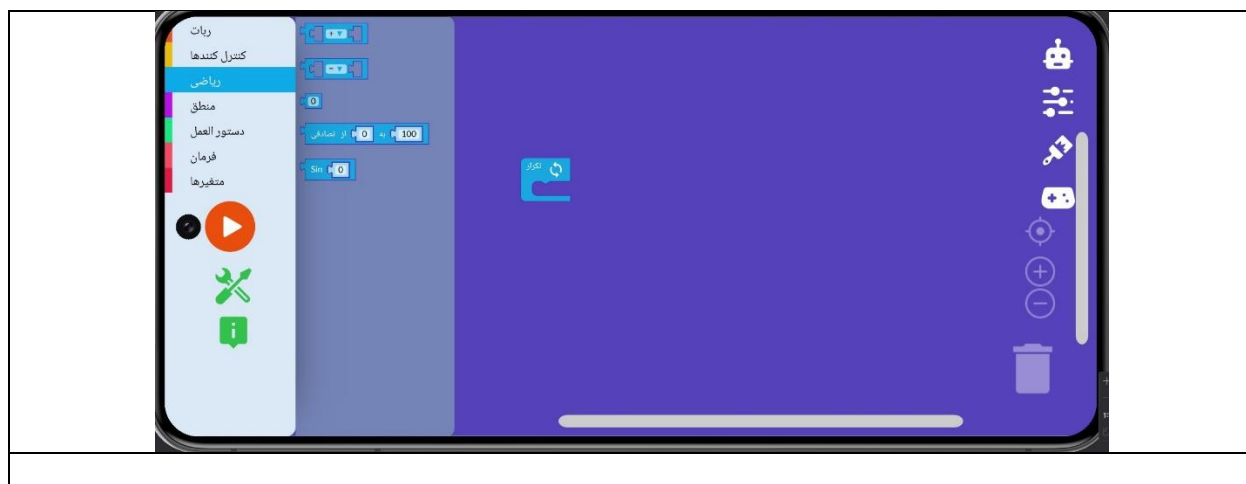
این کنترل کننده ها تعیین می کنن:

- کی دستور اجرا بشه
- چطور شرایط بررسی بشن
- و چگونه تصمیم گیری ها صورت بگیره

با ترکیب این کنترل ها می تونید رفتارهای پیچیده و هوشمندانه ای برای رباتتون بسازید. ✨

منتظرم برات مثال های کاربردی و جذابی آماده کنم تا کامل باهاشون آشنا بشی! 🤖😊

چی دوست داری اول توضیح بدم؟ شرط ها، حلقه ها یا شاید شرط های چندگانه؟ 🤖😐



انتظار

بخش انتظار (Delay) خیلی مهم و کاربردی! 🕒⌚

پیشرو خلاقیت آزاد مهر ویرا

وقتی از این دستور استفاده می کنید، می تونید فاصله یا تاخیر زمانی بین اجرای دستورات رو مشخص کنید. عدد جلوی این بلوک، مقدار تاخیر رو بر حسب میلی ثانیه تعیین می کنه. مثلا:

۱۰۰۰ یعنی ۱ ثانیه

۲۰۰۰ یعنی ۲ ثانیه

این دستور باعث میشه ربات حرکت ها و تغییراتش رو با فاصله زمانی منطقی و قابل فهم انجام بده. 💡🤖

مثال جالبی که گفتم هم خیلی خوبه:

بدون تاخیر، چراغ روشن و خاموش میشه و ما فقط یه چشمک خیلی سریع و نامشخص می بینیم ⚡

ولی با گذاشتن انتظار ۲ ثانیه ای، چراغ به مدت ۲ ثانیه روشن می مونه، بعد خاموش میشه و دوباره بعد ۲ ثانیه روشن میشه و این چراغ چشمک زن خیلی واضح و زیبا میشه! 🌟🔦

پس همیشه موقعی که می خواید حرکت ها یا تغییرات قابل تشخیصی داشته باشید، از این بلوک استفاده کنید. 🎯

اگر دوست داری می تونم برات برنامه ی چراغ چشمک زن ساده با این بلوک بنویسم! 😊🔥

اگر
if

دستور اگر (if) یکی از پایه های اصلی برنامه نویسیه و در ربات سازی هم فوق العاده کاربردییه! 🤖⚡

با این دستور می تونید به ربات بگید:

«اگر یه شرط خاص درست بود، یه کاری انجام بده؛ وگرنه انجام نده یا کار دیگه ای انجام بده.»

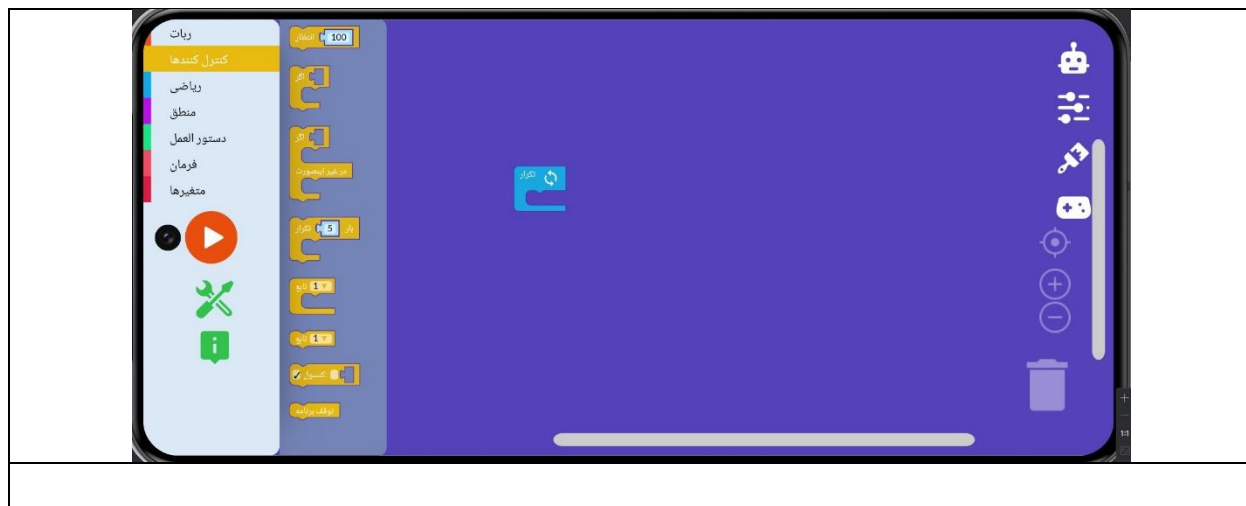
پیشرو خلاقیت آزاد مهر ویرا

مثلاً:

- اگر جمع دو عدد بزرگ‌تر از ۱۰ بود، چراغ روشن شود 💡
 - اگر کلید فشار داده شد، ربات حرکت کند 🚀
 - اگر مانعی جلو ربات بود، ربات مسیرش را تغییر دهد ➡️ ⬅️
- اینطوری ربات شما هوشمند میشه و تصمیماتش بر اساس شرایط محیطی گرفته میشه. 🧠 🎯

💡 نمونه‌هایی که می‌تونیم بنویسیم:

- اگر فاصله حسگر کمتر از ۱۰ سانتی‌متر بود، بوق زده شود
- اگر دکمه فشار داده شد، موتور روشن شود
- اگر نور کم بود، چراغ روشن شود



بلوک اگر در غیر این صورت
else if

بلوک اگر ... در غیر این صورت (if ... else) یکی از کلیدی‌ترین ابزارها برای تصمیم‌گیری در برنامه‌نویسی رباته! 🤖💡

به کمک این بلوک می‌تونید شرایط مختلف رو به ربات بگید و در هر حالت واکنش مناسب رو انجام بده.

مثلاً:

- اگر عدد بزرگ‌تر از ۵ بود، چراغ روشن شود 💡
- در غیر این صورت، بوق بزند 🔔

پیشرو خلاقیت آزاد مهر ویرا

یا:

- اگر فاصله ربات با مانع کمتر یا مساوی ۳۰ سانتی متر بود، تغییر مسیر بده 

- در غیر این صورت، به حرکت ادامه بده 

این یعنی شما برای همه حالت‌ها برنامه‌ریزی می‌کنید و رباتتون هوشمندانه عمل می‌کنه!  

بلوک تکرار به دفعات دلخواه

بلوک تکرار به دفعات دلخواه  برای مواقعی که می‌خواه یه دستور یا مجموعه دستورات رو دقیقاً به تعداد مشخصی تکرار کنی، مثل روشن و خاموش کردن چراغ به تعداد ۵ بار!  

نکته مهم:

- این بلوک فرق داره با بلوک تکرار اصلی آبی‌رنگ که کل برنامه رو پشت سر هم و همیشه تکرار می‌کنه.
- توی بلوک تکرار به دفعات دلخواه، فقط اون بخشی که داخلش قرار میدی، همون تعداد بار مشخص تکرار می‌شه و بعد برنامه ادامه پیدا می‌کنه یا تموم می‌شه.

تابع

تابع (Function) تو برنامه‌نویسی یکی از بهترین ابزارها برای مرتب کردن و ساده‌سازی کدهاتونه.  

◆ چی هست؟

تابع مثل یه بسته یا مجموعه‌ای از دستورات هست که می‌سازی و بهش یه اسم میدی. دیگه لازم نیست هر دفعه اون دستورات رو تکرار کنی، فقط کافیه تابع رو صدا بزنی تا اون مجموعه اجرا بشه.

◆ چرا مهمه؟

- باعث میشه کدهات تمیزتر و قابل فهم‌تر باشن
- مدیریت و تغییر توی کد راحت‌تر بشه
- از تکرار بی‌مورد جلوگیری می‌کنه

◆ مثال واقعی:

مثل مثال خودت با مداد تراشیدن، وقتی بخوای دوباره انجامش بدی، فقط میگی «مدادمو تراش کن» بدون اینکه جزئیاتش رو از اول بگی.

پیشرو خلاقیت آزاد مهر ویرا

♦ تو ربات:

فرض کن میخوای چراغ چشمک زن و بوق هشدار رو چندین بار استفاده کنی، بجای نوشتن هر بار همه دستورات، تابعی بساز که این کار رو انجام بده، هر وقت خواستی فقط تابع رو صدا می زنی. 📢🚦

اگر دوست داری، می تونم بهت کمک کنم یک نمونه تابع ساده هم بسازیم! 🎯

۴.۱-mini

کنسول

بخش کنسول (Console) تو اپلیکیشن برنامه نویسی ربات، یه ابزار فوق العاده کاربردی برای مشاهده و بررسی دقیق مقادیر ورودی ها و خروجی های برنامه ت.

چطوری کار می کنه؟

- سمت راست صفحه اپلیکیشن یه قسمت هست که با نماد نقطه و خط مشخص شده.
- وقتی اون رو انتخاب می کنی، یه صفحه آبی با قسمت پایین زرد رنگ برات باز میشه.
- اینجا می تونی ورودی های مختلف مثل حسگرها، متغیرها و... رو ببینی.
- **بخش نظارت بر ورودی داده های حسگر فاصله**، حسگرهای ۱ تا ۴ و موقعیت یابی عمودی و افقی تلفن رو به صورت لغزنده نشون میده.
- پایین صفحه، بخش کنسول هست که نتایج عددی برنامه ت رو نمایش میده.

کاربردهای مهم کنسول:

- مشاهده ی مقدار متغیر یا حاصل عملیات در برنامه
- دیدن عدد دقیق فاصله ی ربات با مانع (از حسگر فاصله)
- نمایش شدت نور دریافتی از حسگر نور
- بررسی عملکرد کد و عیب یابی برنامه در زمان واقعی

نکته های مهم:

- کنسول رو باید داخل بلوک تکرار قرار بدی تا مداوم به روزرسانی بشه و اطلاعات جدید رو نشون بده.

پیشرو خلاقیات آزاد مهر ویرا

- برای نمایش، باید تیک کنسول رو بزنی تا مقدارش دیده بشه، اگر تیک نزنی نمایش داده نمیشه.
- می تونی چند کنسول تعریف کنی و با اختصاص شماره یا اسم، اعداد نمایش داده شده رو به هم مرتبط کنی تا راحت تر متوجه بشی هر عدد مربوط به کدوم بخش هست.

مثال ساده:

فرض کن می خوای نتیجه ی جمع دو عدد رو ببینی.

- داخل بلوک تکرار، بلوک کنسول رو قرار بده
- بهش بگو جمع دو عدد رو نشون بده
- حالا تو صفحه ی کنسول می بینی هر بار عدد جدید چقدر هست

توقف برنامه

توقف برنامه (Stop Program)

این بخش خیلی مهم و کاربردی ست وقتی که بخواهید اجرای برنامه رو کاملاً متوقف کنید.

کاربردها:

- وقتی به شرط خاصی رسیدید و می خواهید برنامه ادامه پیدا نکند
- مواقعی که خطایی پیش آمده و باید اجرای ربات متوقف شود
- برای پایان دادن به یک روند خاص و جلوگیری از ادامه اجرای کد

نکته:

- بعد از اجرای این دستور، برنامه در همان لحظه کاملاً متوقف می شود و هیچ دستوری بعد از آن اجرا نخواهد شد.
- اگر این بلوک داخل حلقه تکرار یا بلوک شرطی باشد، اجرای آن به محض رسیدن به دستور، متوقف خواهد شد.

منطق

منطق

پیشرو خلاقیت آزاد مهر ویرا

در این بخش با عملگرهای منطقی آشنا می‌شوید:

- و (AND) 
- یا (OR) 
- صحیح (True) 
- غلط (False) 

این ابزارها به شما کمک می‌کنند شرایط مختلف را به هم وصل کنید و تصمیمات هوشمندانه‌تری بگیرید! 💡🤖



صحیح و غلط

در منطق دیجیتال فقط دو حالت داریم:

- درست یا غلط
- روشن یا خاموش
- صفر یا یک

هر عبارتی یا درست است یا غلط، هیچ حالت وسطی وجود ندارد! این پایه‌ی تصمیم‌گیری در برنامه‌نویسی ربات است. 🤖🌟

عملگر و (AND)

این عملگر یعنی «و» به این معناست که برای خروجی گرفتن نتیجه درست ، هر دو شرط باید برقرار باشند!

پیشرو خلاقیت آزاد مهر ویرا

اگر حتی یکی از شرایط غلط ✗ باشد، نتیجه کل عبارت غلط می‌شود.

مثال ساده:

- اگر دو دوست به مهمانی دعوت شده‌اند، فقط زمانی اجازه ورود دارند که هر دو حاضر باشند.
 - اگر فقط یک نفر آمد یا هیچکدام نیامدند، هیچکدام وارد نمی‌شوند.
- در برنامه‌نویسی ربات هم به همین شکل:
- مثلاً اگر عدد حسگر فاصله هم کمتر از ۳۰ باشد و هم کمتر از ۵۰ (یعنی بین ۳۰ و ۵۰ باشد)، آنگاه چراغ روشن شود.
 - اگر هر کدام از این شرط‌ها برقرار نباشد، چراغ خاموش می‌ماند.

در کل:

✓ نتیجه «و» فقط وقتی درست است که همه شروط درست باشند!

عملگر یا (OR) 

این عملگر یعنی «یا» به این معناست که اگر حداقل یکی از شرط‌ها درست باشد، نتیجه کلی درست ✓ است!

تنها زمانی نتیجه کلی غلط ✗ می‌شود که هر دو شرط نادرست باشند.

مثال ساده:

- اگر دو دوست به مهمانی دعوت شده‌اند، نگهبانان اجازه ورود می‌دهند حتی اگر یکی از آن‌ها بیاید.
- فقط زمانی که هیچ‌کدام از دوستان نیایند، کسی اجازه ورود ندارد.

در برنامه‌نویسی ربات:

- اگر عدد اول بزرگ‌تر از ۳ باشد یا عدد دوم بزرگ‌تر از ۳ باشد، عبارت کلی درست است.
- یعنی اگر حداقل یکی از این شروط برقرار باشد، کار مورد نظر انجام می‌شود.

خلاصه:

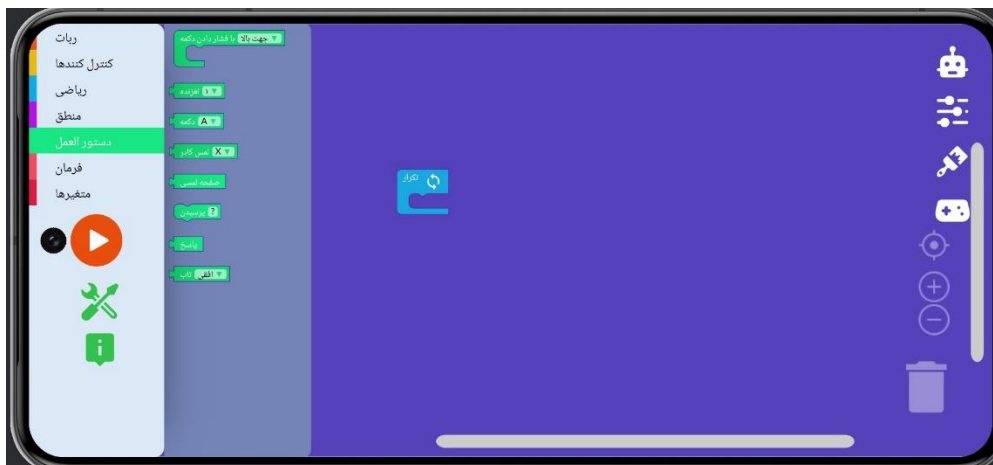
✓ نتیجه «یا» وقتی درست است که یکی یا هر دو شرط برقرار باشند.

✗ فقط وقتی هر دو شرط غلط باشند، نتیجه غلط است!

دستورالعمل 

در این بخش می‌توانید با کنترل دکمه‌ها و لغزنده‌ها که در نرم‌افزار آماده شده‌اند، ربات خود را مثل یک ماشین کنترلی 🚗 هدایت کنید.

پیشرو خلاقیت آزاد مهر ویرا



برای دسترسی:

♦ در سمت راست صفحه، پایین‌ترین آیکون که شبیه دسته بازی 🎮 هست رو بزنید.

♦ صفحه‌ای باز می‌شود که شامل دکمه‌های A، B، و چهار جهت ⬆️ ⬇️ ⬇️ ⬆️ به همراه لغزنده‌ها است.

با این ابزارها می‌تونید ربات رو با فرمان‌های مختلف کنترل کنید و بازی رو مدیریت کنید! 🤖

با فشار دادن دکمه 🎯

فرض کنید می‌خواهید با زدن یک دکمه خاص، یک عملیات ویژه و منحصر به فرد انجام شود.

✅ یکی از روش‌ها این است که داخل بلوک تکرار برنامه‌نویسی کنید و با دستور **اگر (if)** بررسی کنید که آیا دکمه مورد نظر فشار داده شده است یا نه، سپس عملیات را اجرا کنید.

اما روش ساده‌تر و بهتر:

♦ خارج از بلوک تکرار، می‌توانید از این گزینه استفاده کنید.

♦ اینجا می‌توانید مستقلاً برای هر دکمه یک دستور خاص بنویسید که به محض فشار دکمه، همان دستور اجرا شود.

⚠️ نکته مهم:

این دستور از محدود دستوراتی است که نیازی نیست داخل بلوک تکرار قرار بگیرد و مستقل از آن کار می‌کند!

مثال:

• با فشار دادن دکمه A، چراغ شماره ۱ روشن شود 💡

• با فشار دادن دکمه B، همان چراغ خاموش شود 🖐️

این کار باعث می‌شود برنامه شما تمیزتر و واکنش سریع‌تر به دکمه‌ها داشته باشد! 🎉

پیشرو خلاقیت آزاد مهر ویرا

لغزنده 

با انتخاب این پارامتر، می‌توانید دستورات لازم را به لغزنده‌های ۱ و ۲ بدهید.

هر لغزنده مقداری بین ۰ تا ۱۰۰ دارد که بسته به محل قرارگیری آن تغییر می‌کند. 

مثال:

اگر عدد لغزنده اول کمتر از ۵۰ باشد، چراغ روشن شود 

برای درک بهتر، می‌توانید از بخش کنسول استفاده کنید و لغزنده ۱ یا ۲ را جایگذاری کنید تا تغییرات عددی بین ۰ تا ۱۰۰ را در

کنسول ببینید 

دکمه 

با انتخاب این پارامتر، می‌توانید تمام دکمه‌های دسته بازی را کنترل کنید و هر دستوری که می‌خواهید با فشردن کلید منتخب اجرا کنید.

مثال‌ها:

• اگر کلید A فشار داده شود، هر دو موتور به سمت جلو روشن شوند 

• اگر کلید B فشار داده شود، بوق روشن شود 

حتی اگر هیچ دکمه‌ای فشار داده نشود، می‌توانید با گزینه «هیچ» برنامه بنویسید و واکنش‌های خاصی تعریف کنید! 

صفحه لمسی 

به سمت راست، کنار نماد قلم‌مو، این بخش قرار دارد.

این قسمت بیشتر برای برنامه‌ریزی و مشاهده مسیر ربات تعقیب خط طراحی شده، اما کاربرد مهم دیگرش استفاده از صفحه لمسی (Touch) است.

وقتی این صفحه را باز کنید، با ضربه یا لمس، خروجی‌ای به صورت درست یا غلط (۰ و ۱) دریافت می‌کنید. شما می‌توانید این خروجی صفر و یک صفحه لمسی را در برنامه‌نویسی ربات خود به کار ببرید و بر اساس لمس

صفحه، عملکردهای مختلفی تعریف کنید. 

تاب 

تا حالا فکر کردید چطور با چرخوندن گوشی، جهت صفحه تغییر می‌کند؟

در واقع گوشی شما مجهز به تکنولوژی «ژیروسکوپ» است که جهت و چرخش دستگاه رو تشخیص می‌ده.

ما در این اپلیکیشن مستقیماً به ژیروسکوپ گوشی شما وصل شدیم! 

پیشرو خلاقیت آزاد مهر ویرا

اگر بخواید برنامه‌ای بسازید که با تاب دادن گوشی به صورت افقی یا عمودی، ربات دستوری بگیره، از این بخش استفاده کنید.

مثلاً می‌تونید طوری برنامه‌ریزی کنید که فقط با حرکت گوشی توی دست، ربات مثل ماشین کنترلی حرکت کنه! 🎮🚗

برای دیدن نتایج هم حتی نیاز به اتصال به ربات نیست.

کافیه از سمت راست اپلیکیشن، نماد دسته بازی 🎮 رو انتخاب کنید و «کنترل پل» رو باز کنید. اونجا دو ردیف پایین با نام‌های «افقی» و «عمودی» وجود داره که جهت‌گیری گوشی رو نمایش می‌ده. وقتی گوشی رو حرکت می‌دید، مقدار این اعداد تغییر می‌کنه. همچنین می‌تونید در بخش کنسول، تاب افقی یا عمودی رو انتخاب کنید و تغییرات عددی این جهت‌گیری‌ها رو ببینید. 📊👁️

متغیرها ✨📊

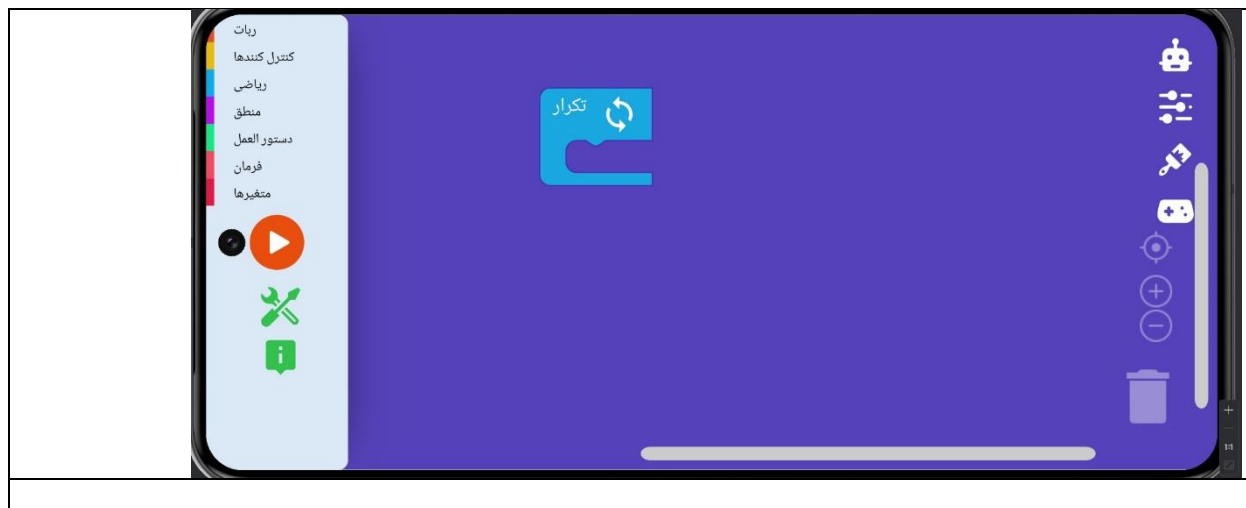
در این بخش می‌تونید به تعداد دلخواه متغیر بسازید و برنامه‌هاتون رو حرفه‌ای‌تر کنید! بعد از ساخت متغیر، می‌تونید بهش مقدار اولیه بدید و هر زمان خواستید مقدارش رو تغییر بدید. 🔄

این امکان باعث میشه برنامه شما انعطاف‌پذیرتر و هوشمندتر بشه و راحت‌تر بتونید اطلاعات رو ذخیره و مدیریت کنید. 🧠💡



پیشرو خلاقیت آزاد مهر ویرا

قدم ششم  
فضای کلی اپلیکیشن



پارامترهای سمت چپ و بالای صفحه که ابزارهای برنامه‌نویسی شما هستند را قبلاً توضیح دادیم.
حالا در ادامه، به نحوه ذخیره اطلاعات و بررسی سایر بخش‌های اپلیکیشن و نکات نهایی می‌پردازیم.  
با این توضیحات، مسیر کامل‌تری برای استفاده بهتر و کارآمدتر از اپلیکیشن خواهید داشت! 

mini-۴.۱

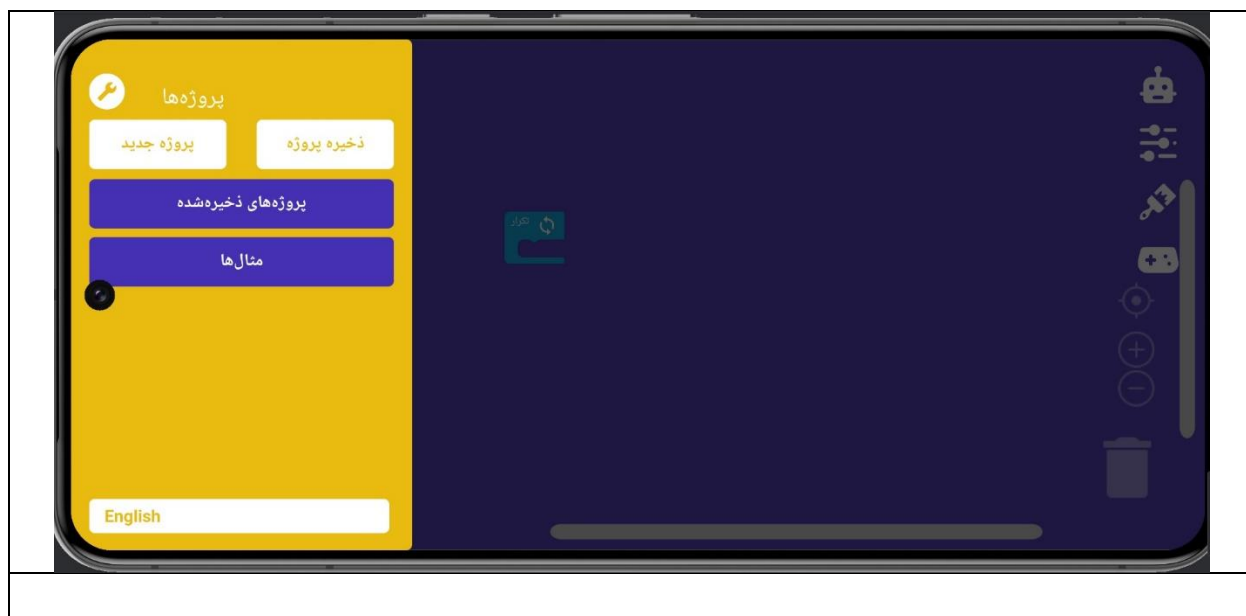
دکمه استارت  پس از نوشتن هر کد، برای اجرای آن کافیست این دکمه را فشار دهید تا برنامه اجرا شود.
همچنین برای توقف برنامه در حال اجرا، می‌توانید دوباره همین دکمه را بزنید. 
یک دکمه ساده ولی بسیار کاربردی برای کنترل اجرای برنامه شما! 

نماد ابزار و آچار  در سمت چپ برنامه این نماد را می‌بینید که مدیریت پروژه‌هایتان را انجام می‌دهد.
در این بخش می‌توانید:

- پروژه‌های خود را ذخیره کنید 
- به هر پروژه نام دلخواه بدهید 
- پروژه جدید بسازید 

پیشرو خلاقیت آزاد مهر ویرا

- پروژه‌های ذخیره شده قبلی را باز کنید 📁
 - مثال‌های آماده شده توسط ما را مشاهده و استفاده کنید 📖
- این بخش کمک می‌کند تا همیشه پروژه‌هایتان مرتب و قابل دسترس باشند! 😊



راهنما !
با انتخاب این علامت تعجب، به سایت رسمی آکادمی پیشرو هدایت می‌شوید! ✨🌐
در آنجا می‌توانید آموزش‌ها، نکات بیشتر و پشتیبانی برای برنامه‌نویسی و کار با ربات‌ها را دریافت کنید. 📖🤖
همیشه یک راهنمای خوب کنار دستتان باشد! 😊

توضیح نمادهای مهم و کلیدی سمت راست اپلیکیشن 🎮🔧
در بخش‌های قبل با ابزارهای برنامه‌نویسی و نوشتن کد آشنا شدید،
اما حالا وقت آشنایی با نمادها و ابزارهای کمکی در سمت راست صفحه است که نقش «عصای دست» شما را دارند! ? ✨
این نمادها به شما کمک می‌کنند:

- داده‌های ورودی ربات را مشاهده کنید 📺
- دستورات کنترل را سریع‌تر اجرا کنید ⚙️

پیشرو خلاقیت آزاد مهر ویرا



- و تعامل بهتری با ربات‌تان داشته باشید

پس هر وقت خواستید دستورات را مدیریت یا ورودی‌ها را ببینید، به سمت راست صفحه نگاه



کنید! اگر بخواهید، توضیح هر نماد را هم جداگانه می‌توانم برایتان بگویم!



نماد ربات

این نماد مخصوص اتصال اپلیکیشن موبایل به ربات شماست.

با استفاده از این بخش، می‌توانید ارتباط بین دستگاه و ربات را برقرار کنید تا دستورات نوشته شده روی گوشی به ربات ارسال شود و کنترل آن را به دست بگیرید.

این قسمت در قدم دوم آموزش به طور کامل توضیح داده شده است و پایه اصلی تعامل شما با ربات به شمار می‌رود.

پس هر وقت خواستید ربات را به گوشی وصل کنید، کافی است روی این نماد کلیک کنید و مراحل اتصال را



دنبال کنید!



نماد کنترل پنل

(نقطه و خط)

این بخش برای نظارت و مشاهده ورودی‌ها طراحی شده است.

- کادر آبی رنگ نشان‌دهنده داده‌های ورودی از ربات و گوشی شماست؛ مثل حسگرها و سنسورهای مختلف. اینجا

می‌توانید مقدار حسگرها مثل فاصله، نور و موقعیت را به صورت لحظه‌ای ببینید.

- کادر زرد رنگ مربوط به کنسول است که وقتی در برنامه‌نویسی خود از بخش کنسول استفاده می‌کنید، می‌توانید

مقادیر و داده‌های خاصی را به صورت عددی و دقیق در این قسمت مشاهده کنید.



با کمک این نماد می‌توانید عملکرد ربات و داده‌های ورودی را به خوبی کنترل و تحلیل کنید!



نظارت بر ورودی

پیشرو خلاقیت آزاد مهر ویرا

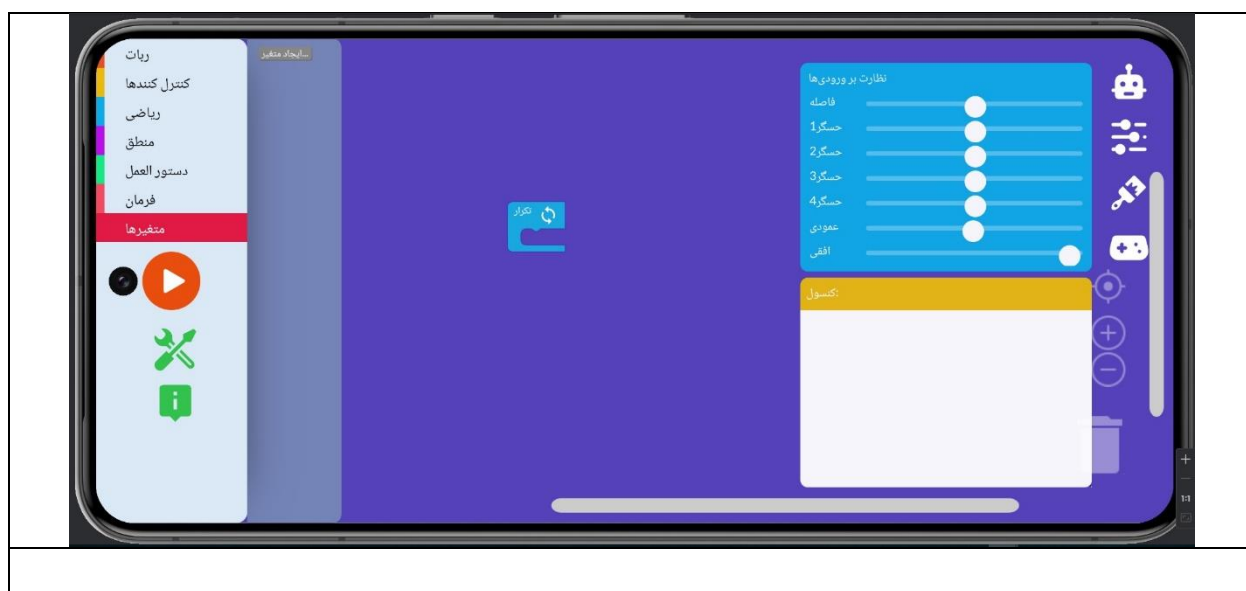
- سمت چپ خط یعنی مینیمم یا کمترین مقدار ممکن

- سمت راست خط یعنی ماکزیمم یا بیشترین مقدار ممکن

حرکت نقطه روی خط برای هر حسگر نشان‌دهنده مقدار عددی تقریبی آن حسگر است.

هر چه نقطه به سمت راست برود، مقدار ورودی افزایش یافته و هر چه به سمت چپ برگردد، مقدار ورودی کاهش یافته است.

این نمایش به شما کمک می‌کند تا به صورت تصویری و سریع وضعیت حسگرها و ورودی‌های ربات را بفهمید!



 حسگرها

 حسگر فاصله

این حسگر معمولاً از نوع سنسور التراسونیک است که در ربات‌های ما تعبیه شده است. نحوه کار آن به این صورت است:

- حسگر یک موج صوتی (امواج فراصوت) ارسال می‌کند.
- سپس زمان رفت و برگشت این موج را محاسبه می‌کند.
- با توجه به این زمان، فاصله جسم مقابل ربات را اندازه‌گیری و به صورت یک عدد نمایش می‌دهد.

اگر ربات شما این حسگر را دارد و اپلیکیشن را به ربات متصل کرده‌اید:

پیشرو خلاقیت آزاد مهر ویرا

- می‌توانید با حرکت دست خود جلو و عقب مقابل حسگر، تغییر مکان نقطه روی خط در بخش نظارت بر ورودی را ببینید.

- هر چقدر دستتان به حسگر نزدیک‌تر باشد، نقطه به سمت چپ (مقدار کمتر) و هر چقدر دورتر باشد، به سمت راست (مقدار بیشتر) حرکت می‌کند.

همچنین:

- با انتخاب بخش **کنسول** و جایگذاری **ورودی حسگر فاصله** در برنامه، می‌توانید مقدار دقیق عددی و تغییرات آن را در کنسول به صورت زنده مشاهده کنید.

- این تغییرات در بخش نظارت بر ورودی هم به صورت تقریبی قابل مشاهده هستند.

👉 دست‌تان را نزدیک و دور کنید تا تفاوت‌ها را ببینید و برنامه‌نویسی بهتر و دقیق‌تری داشته باشید!

حسگر ۱ و ۲ ✨

این دو حسگر معمولاً برای حسگرهای حساس به نور (LDR) و حسگر خط در ربات استفاده می‌شوند.

نکات مهم درباره حسگرهای نور و خط:

- حسگرهای حساس به نور (LDR):

- با تغییر شدت نور محیط واکنش نشان می‌دهند.
- مقدار خروجی این حسگرها در بخش **نظارت بر ورودی** قابل مشاهده است.
- با نوشتن برنامه و استفاده از **کنسول** می‌توانید مقادیر دقیق عددی این حسگرها را ببینید.
- وقتی نور محیط زیاد است، مقدار عددی حسگر کاهش می‌یابد و وقتی محیط تاریک است، مقدار عددی افزایش می‌یابد.
- برای تست، می‌توانید از **فلش گوشی** برای روشنایی استفاده کنید یا با دست مسیر نور را مسدود کنید تا محیط تاریک شود.

- حسگر تشخیص نور مادون قرمز (IR):

- می‌توانید از آن هم استفاده کنید و تغییرات آن را در بخش نظارت بر ورودی یا کنسول ببینید.
-

پیشرو خلاقیت آزاد مهر ویرا

نکته مهم ⚠️

اگر از یکی از این حسگرها استفاده می کنید، حتماً دقت کنید که:

- کدام حسگر روی برد به کار گرفته شده است (مثلاً حسگر شماره ۱ یا حسگر شماره ۲).
- برنامه نویسی شما باید مطابق با همان حسگر باشد.
- اگر حسگر شماره ۱ به برد متصل است ولی برنامه برای حسگر ۲ نوشته شده باشد، ربات پاسخ مناسبی نخواهد داد.

حسگر ۳ و ۴

این حسگرها در واقع **کلیدهای روی برد ربات** هستند.

کارشون خیلی ساده ست: فقط دو حالت دارند،

✓   (صفر) یعنی خاموش

✓   ۱۰۰ (صد) یعنی روشن یا فشار داده شده

وقتی کلیدها رو فشار میدی، می تونی تغییر وضعیتشون رو تو **کنترل پنل** ببینی و برنامه ت رو بر اساس اون ها تنظیم کنی!



حسگرهای عمودی و افقی

این دو حسگر مستقیماً به گوشی شما متصل هستند و نیازی به اتصال ربات ندارید!

آن ها به **ژیروسکوپ** و **شتاب سنج** گوشی وصل شده اند و میزان **زاویه و جهت گیری** گوشی را در حالت افقی و عمودی اندازه گیری می کنند.

برای تست:

◆ گوشی تان را صاف بگیرید

◆ سپس به آرامی آن را بچرخانید یا برعکس کنید

◆ ببینید چطور مقدارهای افقی و عمودی تغییر می کنند!

با این قابلیت می توانید برنامه بنویسید که:

🚗 ربات با حرکت و چرخش گوشی در دست شما حرکت کند و رانندگی کند!

اگر می خواهید مقدار عددی این حسگرها را ببینید، کافی ست از **بخش کنسول** و دستورالعمل های **تاب افقی** یا **تاب عمودی**

استفاده کنید تا اعداد مربوط به موقعیت گوشی را مشاهده کنید. 🎮 ⚡

کنسول 

پیشرو خلاقیت آزاد مهر ویرا

در این بخش می‌توانید با نوشتن برنامه و انتخاب دستور **کنسول**، همه چیز را مشاهده و نظارت کنید! هر داده‌ای که بخواهید—مثل خروجی‌های برنامه یا ورودی‌های ربات—در کنسول نمایش داده می‌شود.

برای استفاده:

✓ دستور کنسول را انتخاب کنید

✓ سپس مقدار یا پارامتری که می‌خواهید ببینید را جایگذاری کنید

به این صورت می‌توانید به راحتی تغییرات و مقادیر ورودی و خروجی را دنبال کنید و برنامه‌تان را بهتر کنترل کنید!  

نماد قلم‌مو 

صفحه‌ی ترسیم

در این بخش می‌توانید مسیر حرکت رباتی که برنامه داده‌اید را مشاهده کنید، مخصوصاً اگر ربات روی مسیرهای مشخص مثل دایره، مستطیل یا خط حرکت کند.

همچنین، با انتخاب بخش **صفحه‌ی لمسی**، می‌توانید از آن به عنوان یک کلید لمسی استفاده کنید! اگر می‌خواهید با لمس این صفحه دستوری اجرا شود، می‌توانید برنامه آن را بنویسید و از این قسمت به عنوان صفحه‌ی لمسی بهره ببرید.



خروجی لمس صفحه از نوع **درست یا غلط (True/False)** است، یعنی هر بار که صفحه را لمس کنید، مقدار آن تغییر می‌کند و می‌توانید این مقدار را در برنامه خود استفاده کنید.  

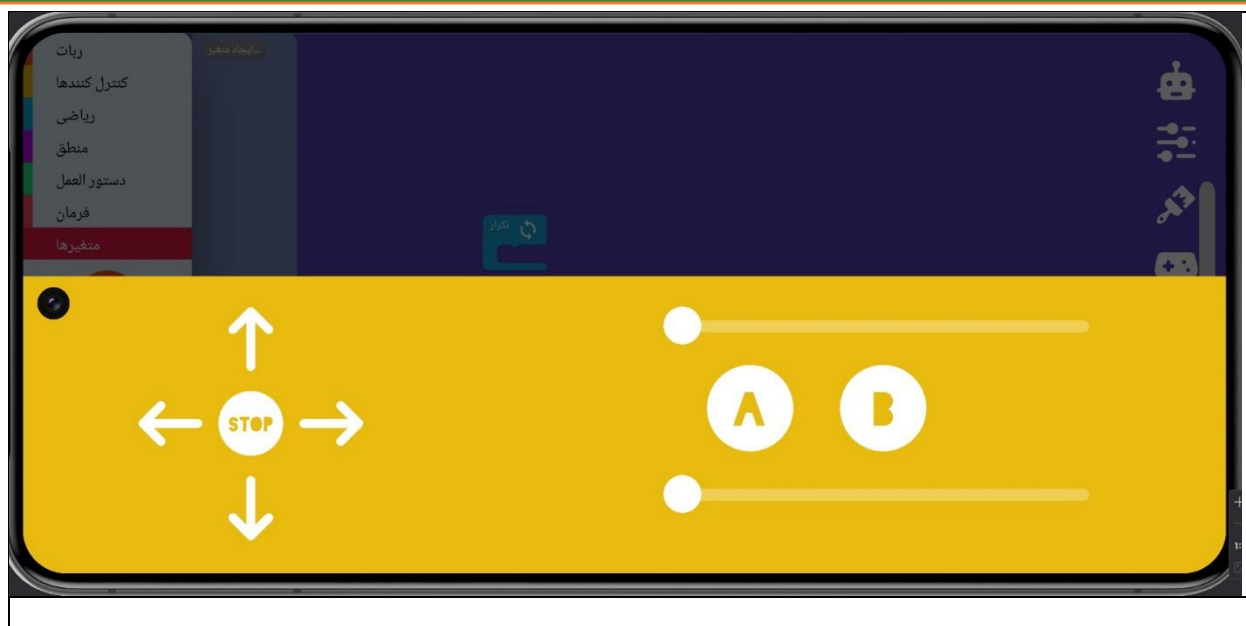
نماد دسته کنترل 

رابط دسته کنترل

این بخش هیجان‌انگیز، اتاق فرمان ربات شماست!

در این رابط، چهار جهت اصلی (بالا، پایین، چپ، راست)، دکمه‌های A و B، دکمه Stop و دو لغزنده (لغزنده ۱ و لغزنده ۲) در اختیار شما قرار دارند.

پیشرو خلاقیت آزاد مهر ویرا



شما می‌توانید برای هر یک از این کلیدها برنامه‌ای بنویسید که با فشردن آن‌ها دستورات خاصی اجرا شود. مثلاً:

- با دکمه‌های جهت، ربات شما مثل یک ماشین کنترلی حرکت کند.
- با دکمه‌های A و B، رقص نورهای مختلفی را اجرا کنید.
- با لغزنده‌ها تعیین کنید که کی بوق بزند یا چراغ روشن شود.

یک نکته مهم:

شما حتی می‌توانید برای حالت «هیچ» یا «هیچ دکمه‌ای فشرده نشده» برنامه بنویسید. یعنی وقتی هیچ کلیدی فشرده نیست، چه اتفاقی بیفتد.

مثلاً:

- وقتی هیچ دکمه‌ای زده نشده، موتورهای ربات خاموش شوند.
- یا همیشه ربات حرکت کند و فقط وقتی دکمه Stop زده شد، موتورها خاموش یا برنامه متوقف شود.

این انعطاف باعث می‌شود کنترل کامل و حرفه‌ای روی ربات خود داشته باشید! 🎵💡🚗



نماد سطل آشغال

حذف پارامترهای دلخواه

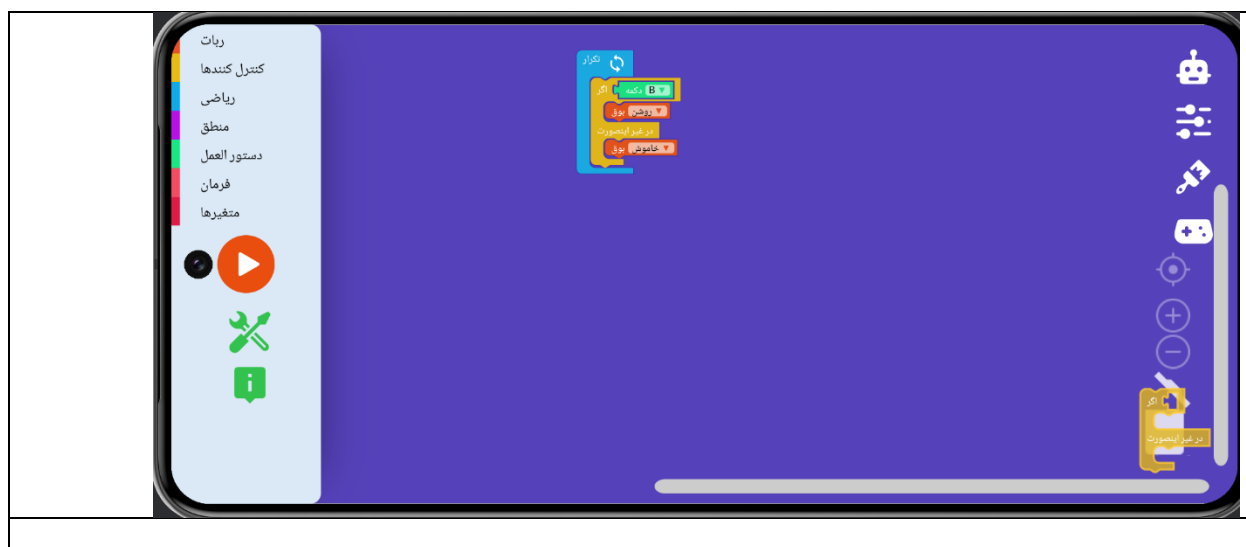
پیشرو خلاقیت آزاد مهر ویرا

در این بخش شما می‌توانید هر بلوک یا دستوری را که از گروه‌های سمت چپ به فضای کاری خود آورده‌اید، به راحتی حذف کنید.

فقط کافی است:

- بلوک یا دستوری که می‌خواهید پاک کنید را انتخاب کنید.
- سپس آن را به سمت نماد سطل آشغال بکشید و رها کنید.

به این صورت بخش‌های غیرضروری یا اشتباه به سرعت از برنامه شما حذف خواهند شد و فضای کاری شما مرتب باقی می‌ماند.



بزرگ‌نمایی، کوچک‌نمایی و تنظیم فوکوس در محیط برنامه‌نویسی 🔍

در هنگام کار با برنامه‌نویسی، گاهی اوقات لازم است صفحه را بزرگ‌تر یا کوچک‌تر کنید تا بهتر بتوانید کدها یا بخش‌های مختلف را ببینید 😊.

برای این کار، سه نماد مهم وجود دارد:

- **+** نماد به‌علاوه: با کلیک روی این نماد، صفحه شما بزرگ‌تر یا زوم بیشتر می‌شود.
- **-** نماد منهای: با استفاده از این گزینه، صفحه کوچک‌تر یا زوم کمتر خواهد شد.

پیشرو خلاقیت آزاد مهر ویرا

-  **نماد نقطه یا بازگشت به حالت اصلی:** اگر خواستید صفحه به حالت استاندارد و اولیه خودش برگردد، این نماد کمک‌تان می‌کند.

 **نکته مهم:**

در هر کدام از این حالت‌ها، اگر صفحه از اندازه مانیتور بزرگ‌تر شد، می‌توانید با کشیدن صفحه به سمت بالا، پایین، چپ یا راست، باقی قسمت‌های آن را ببینید  .

 **امکانات اضافه در فضای اپلیکیشن برای راحتی بیشتر کاربران**

حالا که با همه بخش‌های اصلی برنامه آشنا شدیم، وقتشه که یک قدم جلوتر بریم و با امکانات اضافی آشنا بشیم که مخصوص راحت‌تر کردن کار شما طراحی شدن 😊 !

این امکانات برای تمام دستورات در برنامه در دسترس هستن و استفاده ازشون خیلی ساده‌ست:

 فقط کافیه یکی از دستورات رو انتخاب کنید و انگشت تون رو چند لحظه روی اون نگه دارید...

 حالا می‌بینید که یکسری گزینه‌های جدید و کاربردی براتون ظاهر میشه! این امکانات کمک می‌کنن تا دستورات رو راحت‌تر و سریع‌تر مدیریت کنید.

مثلاً ممکنه بتونید:

-  دستور رو کپی یا ویرایش کنید
-  حذفش کنید
-  یا اون رو به جای دیگه‌ای منتقل کنید

پیشرو خلاقیت آزاد مهر ویرا

🎯 این ویژگی‌ها باعث میشن کنترل بیشتری روی برنامه داشته باشید و همه چیز براتون ساده‌تر و جذاب‌تر بشه!

🔑 گزینه‌ی تکرار (کپی کردن دستورات)

گاهی وقت‌ها در برنامه‌نویسی لازم دارید یک دستور یا حتی یک بخش کامل از برنامه رو چند بار استفاده کنید.

اما آیا هر بار باید دوباره اون رو از اول بنویسید؟ 😞
نه! اصلاً لازم نیست وقتتون رو تلف کنید.

💡 شما می‌تونید خیلی راحت، با استفاده از گزینه‌ی تکرار، دستور یا بلوکی که نوشتید رو کپی کنید و ازش در بخش‌های دیگه استفاده کنید.

مثال کاربردی:

فرض کنید دارید یک برنامه برای ماشین کنترلی 🚗 یا چراغ‌های رقصان 🌟 می‌نویسید.
لازم نیست هر بار از اول برید و دستور "موتور" یا "چراغ" رو از بخش ربات بیارید.
فقط کافیه همون بلوک رو انتخاب و تکرار کنید 🔄!

یا مثلاً اگه یک بلوک شرطی مثل {اگر {یا f ساخته‌اید و می‌خواهید با ورودی‌های مختلف دوباره از اون استفاده کنید،
خیلی راحت می‌تونید کل اون بلوک رو کپی کنید و فقط ورودی‌هاش رو تغییر بدید ✂️🧠.

📝 اضافه کردن نظر (کامنت) روی دستورات

گاهی وقتاً موقع برنامه‌نویسی ممکنه بخوای یک یادداشت یا توضیح درباره یک دستور خاص بنویسی تا بعداً یادت بیاد اون دستور دقیقاً برای چی بوده 😊.

اینجا گزینه‌ی "اضافه کردن نظر" به کمکت میاد 💡👉!

چطور نظر اضافه کنیم؟

۱. اول دستور مورد نظر رو انتخاب کن ✅

۲. بعد، گزینه‌ی "اضافه کردن نظر" رو بزن 🗨️

۳. حالا یک دایره آبی رنگ با علامت سوال ؟ روی اون دستور ظاهر میشه

۴. روی اون دایره کلیک کن و حالا می‌تونی هر توضیح یا نظری که داری بنویسی 📝👉!

مثلاً می‌تونی بنویسی:

"این قسمت کنترل چرخ‌هاست" 💠

پیشرو خلاقیت آزاد مهر ویرا

"♦ اگر ورودی برابر با ۳ بود، چراغ روشن بشه"
یا هر چیزی که بهت کمک کنه بعداً راحت‌تر کدت رو بفهمی.

نکته مهم:

هر وقت دوباره روی اون نماد ؟ کلیک کنی، می‌تونی کامنت رو ببینی یا پنهان کنی.

🗑 و اگه خواستی کامنت رو پاک کنی، دوباره دستور رو انتخاب کن و این بار روی گزینه‌ی "Remove comment" بزنی.

جمع کردن بلوک‌ها

وقتی برنامه‌تون بزرگ میشه و تعداد دستورات زیاد میشه، صفحه شلوغ میشه و پیدا کردن بخش‌های مختلف سخت میشه 😓!
برای همین بهتره از گزینه‌ی جمع کردن بلوک استفاده کنید تا برنامه‌تون جمع‌وجور و مرتب بشه 🧹.

کافیه روی فلش کوچیکی که کنار بعضی از بلوک‌ها ظاهر میشه کلیک کنید ▼
تا اون بلوک جمع بشه و فقط عنوانش دیده بشه. اینطوری همه چیز منظم‌تر دیده میشه 📁!

غیرفعال کردن بلوک‌ها (و دوباره فعال کردن اون‌ها)

در برنامه‌نویسی ممکنه برنامه‌تون درست کار نکنه یا اصطلاحاً دچار باگ بشه 🐛.
در این مواقع، یکی از روش‌های مهم برای پیدا کردن مشکل، اینه که بخش‌به‌بخش برنامه رو تست کنید.

💡 اما شاید نخواهید اون بخش رو حذف کنید... فقط می‌خواهید فعلاً اجرا نشه.

🔧 اینجاست که گزینه‌ی "غیرفعال کردن بلوک" به دردتون می‌خوره:

۱. بلوک مورد نظر رو انتخاب کن ☒

۲. از گزینه‌ها، روی "غیرفعال کردن بلوک" بزنی

۳. حالا می‌بینی که اون بلوک کمرنگ‌تر میشه، یعنی فعلاً توی برنامه اجرا نمیشه 🙄

اگر خواستی دوباره اون بلوک رو فعال کنی، همونجا می‌تونی روی گزینه‌ی "فعال‌سازی بلوک" بزنی تا دوباره روشن بشه ⚡!

🔒 نکته مهم:

- اگه بلوک فعال باشه، گزینه‌ی چهارم میشه غیرفعال‌سازی
- اگه بلوک غیرفعال باشه، گزینه‌ی چهارم تبدیل میشه به فعال‌سازی

پیشرو خلاقیت آزاد مهر ویرا

گزینه‌ی حذف بلوک 

گاهی وقت‌ها ممکنه یک بلوک رو اشتباه اضافه کرده باشید یا دیگه بهش نیاز نداشته باشید.
در این مواقع، به جای شلوغ شدن برنامه، بهتره اون بلوک رو پاک کنید تا برنامه‌تون مرتب بمونه  .

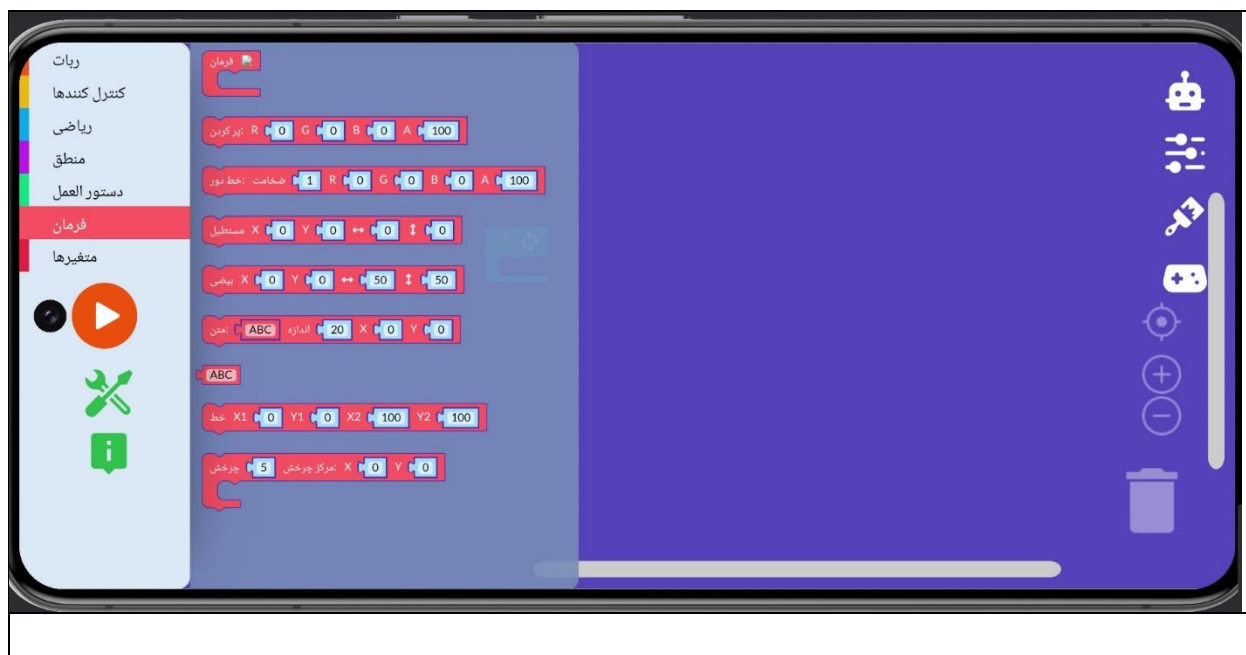
برای این کار:

۱. بلوک مورد نظر رو انتخاب کنید ☒
۲. از بین گزینه‌ها، روی  "حذف بلوک" (Delete Block) کلیک کنید
۳. تمومه! اون بلوک از برنامه حذف میشه و دیگه اجرا نمیشه 

⚠ **نکته:** قبل از حذف بلوک مطمئن بشید که دیگه به اون بخش نیاز ندارید، چون بعد از حذف، باید دوباره دستی بسازیدش) مگر اینکه Undo بزنید. 

 معرفی کامل بخش دستورات ترسیم (Drawing Commands) در PISH BOT 

در این بخش شما می‌تونید به ربات PISH BOT یاد بدهید تا روی برد داخلی خودش نقاشی کند، شکل بکشد و حتی متن بنویسد  !



پیشرو خلاقیت آزاد مهر ویرا

با استفاده از این دستورات می‌توانید طراحی‌های خلاقانه و دیدنی بسازید. بیایید با هم این دستورات را بهتر بشناسیم:

حلقه رسم (DRAW LOOP)

- این قلب تپنده‌ی عملیات ترسیم در PISH BOT است! ❤️
- تمام بلوک‌هایی که داخل این حلقه قرار می‌گیرند، ۲۵ بار در هر ثانیه اجرا می‌شوند. یعنی خیلی سریع ⚡!
- ❌ این حلقه منتظر بلوک WAIT نمی‌ماند و اصلاً متوقف نمی‌شود.
- ⚠️ اگر از بلوک‌های معمولی مثل حرکت موتور یا بلوتوث داخل این حلقه استفاده کنید، ممکن است اختلال ایجاد شود.

بلوک متنی (TEXT)

- با این بلوک می‌توانید متغیرهای متنی بسازید. مثلاً نوشتن نام کاربر، پیام یا برچسب روی صفحه 🏷️.
- ⚠️ اگر این متغیر را در جای عددی یا منطقی استفاده کنید، برنامه خطا خواهد داد!

خط (LINE)

- این بلوک به شما امکان می‌دهد یک خط صاف از یک نقطه به نقطه دیگر بکشید.
- 📍 نقطه‌ی شروع $(X1, Y1)$:
- 📍 نقطه‌ی پایان $(X2, Y2)$:
- می‌توانید با این بلوک شکل‌هایی مثل مثلث، مربع یا حتی ستاره بسازید! ✨

چرخش (ROTATION)

- یک بلوک گزارشی 📊 است که زاویه‌ی چرخش را بین ۰ تا ۳۶۰ درجه تعیین می‌کند.
- 📌 هر شکلی که داخل بلوک DRAW باشد، اول به مرکز چرخش می‌رود و بعد به همان مقدار می‌چرخد.
- مناسب برای ساختن طرح‌های چرخشی، ساعت یا چرخ‌دنده!

پیشرو خلاقیت آزاد مهر ویرا

ضخامت و رنگ خط (STROKE)

- با این بلوک می‌توانید ضخامت و رنگ خط‌ها را تنظیم کنید.
- رنگ‌ها را با تنظیم عددی برای:
 - قرمز (R)
 - سبز (G)
 - آبی (B)
 - شفافیت (A)مشخص می‌کنید.
- مثلاً می‌توانید یک خط ضخیم قرمز با شفافیت ۵۰٪ رسم کنید! ❤️

مستطیل (RECTANGLE)

- رسم یک مستطیل با نقطه شروع در گوشه بالا-چپ. (X, Y)
- اندازه‌های قابل تنظیم:
 - عرض (Width)
 - ارتفاع (Height)
- برای رنگ‌آمیزی مستطیل، از بلوک **FILLING** استفاده می‌شود.

رنگ‌آمیزی (INFILL)

- این بلوک برای تعیین رنگ داخل اشکال به کار می‌رود.
- کاربرد آن در: مستطیل، بیضی، و متن است.
- از پارامترهای R, G, B, A برای انتخاب رنگ دلخواه با میزان شفافیت استفاده کنید. 🎨

بیضی (ELLIPSE)

- رسم یک بیضی زیبا 🌟 در نقطه. (X, Y)
- تنظیم عرض و ارتفاع برای تغییر اندازه بیضی.

پیشرو خلاقیت آزاد مهر ویرا

- رنگ داخلی آن را با بلوک FILLING مشخص می کنید.
- عالی برای طراحی چهره، خورشید یا چشم‌ها 🌞👁️ !

نوشتن متن (TEXT)

- نوشتن یک جمله یا کلمه دلخواه در مختصات (X, Y).
- می توانید اندازه فونت را تعیین کنید 
- رنگ متن هم با همان بلوک FILLING مشخص می شود.

نتیجه گیری:

با استفاده از این بلوک‌ها، می توانید صفحه‌ی نمایش PISH BOT را به یک بوم نقاشی دیجیتال تبدیل کنید!
از ترسیم خطوط و شکل‌ها تا نوشتن پیام‌های خلاقانه، همه چیز در دست شماست  🧠 !