

عرض بوربوينت تدريسي للدرس الأول مقدمة إلى المايكروريت من الوحدة الرابعة البرمجة بواسطة المايكروريت لمقرر التقنية الرقمية



تم تحميل هذا الملف من موقع المناهج السعودية

موقع المناهج ← المناهج السعودية ← الصف الأول الثانوي ← المهارات الرقمية ← الفصل الأول ← ملفات متنوعة ← الملف

تاريخ إضافة الملف على موقع المناهج: 2026-01-07 23:54:03

ملفات اكتب للمعلم اكتب للطالب اختبارات الكترونية اختبارات احلول اعروض بوربوينت اوراق عمل
منهج انجليزي املخصات وتقديرات امذكرة وبنوك اامتحان النهائي للدرس

المزيد من مادة
المهارات
الرقمية:

إعداد: نجود دحمان

التواصل الاجتماعي بحسب الصف الأول الثانوي



الرياضيات



اللغة الانجليزية



اللغة العربية



ال التربية الاسلامية



المواد على تلغرام

صفحة المناهج
السعودية على
فيسبوك

المزيد من الملفات بحسب الصف الأول الثانوي والمادة المهارات الرقمية في الفصل الأول

توزيع منهج التقنية الرقمية 1

1

حل مذكرة التقنية الرقمية مسارات

2

اختبارات التقنية الرقمية (نظري) (الفترة الأولى 1447هـ) مسارات

3

مراجعة محلولة للوحدة الرابعة البرمجة باستخدام لغة ترميز النص الشعبي

4

مراجعة محلولة للوحدة الثانية العمل عبر الانترنت

5

مقرر التقنية الرقمية ١-٣ أول ثانوي

الفصل الدراسي الثالث



المعلمة

نجود دحمان

اللهم احفظ بلادنا وقيادتنا
من كيد الكاذبين وحسد الحاسدين وعبث العابثين



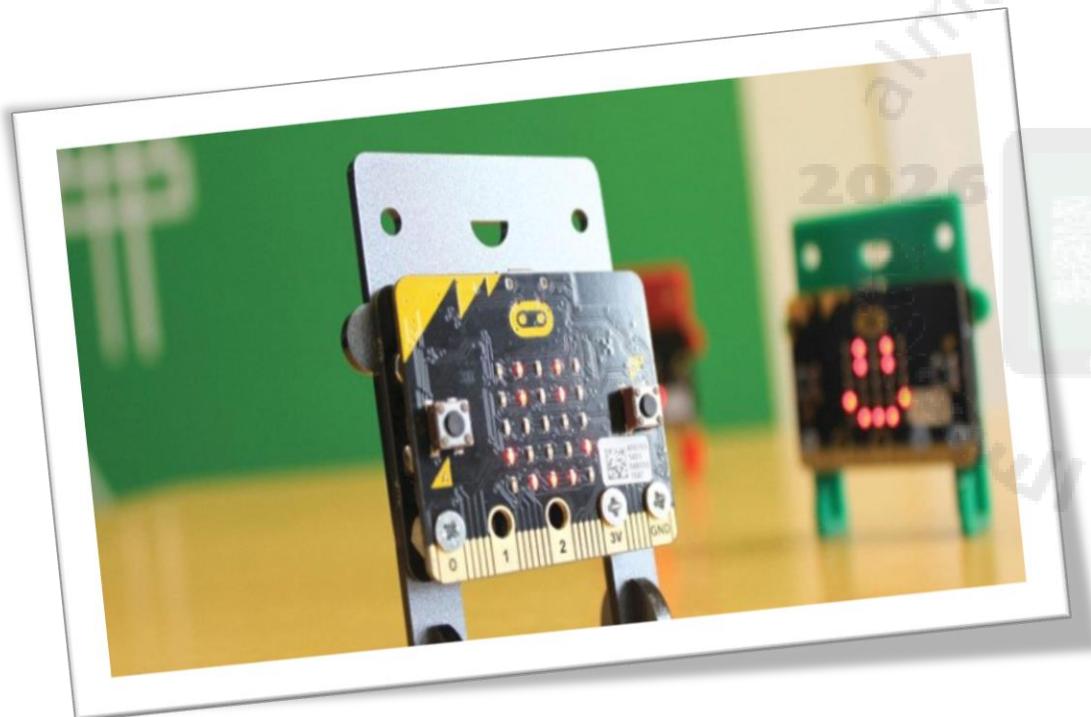
النشيد الوطني

محتويات المنهج



- ١ مستندات ونماذج وتقارير الأعمال
- ٢ الشبكات
- ٣ البرمجة بواسطة المايكروبت





الوحدة الثالثة

البرمجة بواسطة المايكروبوت

تهيئة

سنتعلم في هذه الوحدة كيفية برمجة متحكم دقيق باستخدام لغة نصية .

ستتعرف على أداة مايكروسوفت ميك كود Make code لبرنامج المايكروبوت .

سنتعلم كيفية البرمجة باستخدام لغة بايثون .

سنتعلم كيفية إنشاء أكواد أكثر تعقيداً باستخدام :

المتغيرات والدوال والحلقات واتخاذ القرارات من أجل إكمال المهام المعقدة .

أَهْدَافُ الْوَحْدَةِ

- ماهية المايكروبتس ومكوناته .
- استخدام مايكروسوفت ميك كود .
- أنواع المتغيرات والعمل عليها .
- التعامل مع الأرقام والإحداثيات بلغة بايثون .
- التكرارات في مايكروبتس بلغة بايثون وكيفية استخدامها.
- اتخاذ القرارات في مايكروبتس بلغة بايثون .

محتويات الوحدة

مقدمة إلى المايكروبوت .



المتغيرات والتكرارات .



اتخاذ القرارات .



المشروع



التقويم قبلي

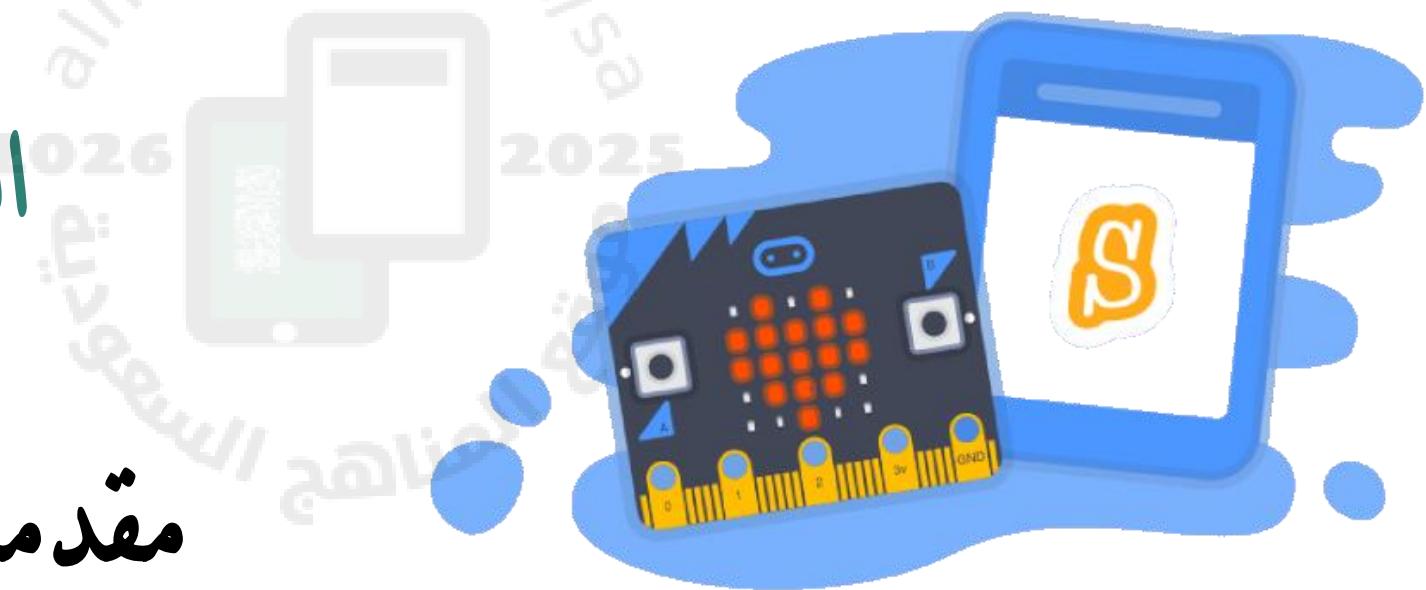
استراتيجية النقاش وال الحوار



- هل قمت ببرمجة روبوت أو متحكم دقيق من قبل ؟
ماذا كان هدفك ؟ وما الأوامر التي اعطيتها للجهاز ؟
- ما أدوات البرمجة التي استخدمتها حتى الآن لإنشاء برنامج ؟
هل كانت لغة برمجة هذه الأدوات قائمة على اللبنات البرمجية أم النصوص ؟
- هل سمعت من قبل أو استخدمت أي لغة برمجية نصية ؟ إذا كانت الإجابة نعم كيف أنشأت البرنامج ؟

مقدمة إلى المايكروبوت

الدرس الأول



أهداف الدرس الجزء الأول

ستتعلم في هذا الدرس

- مفهوم المايكروبت.
- مايكروسوفت ميك كود.
- إنشاء مقطع برمجي في المايكروبت .
- أمثلة برمجية .
- المتغيرات .
- تغيير الأمر .



وزارة التعليم
Ministry of Education

٢٠٢٦

٢٠٢٧

المنهاج المتكامل

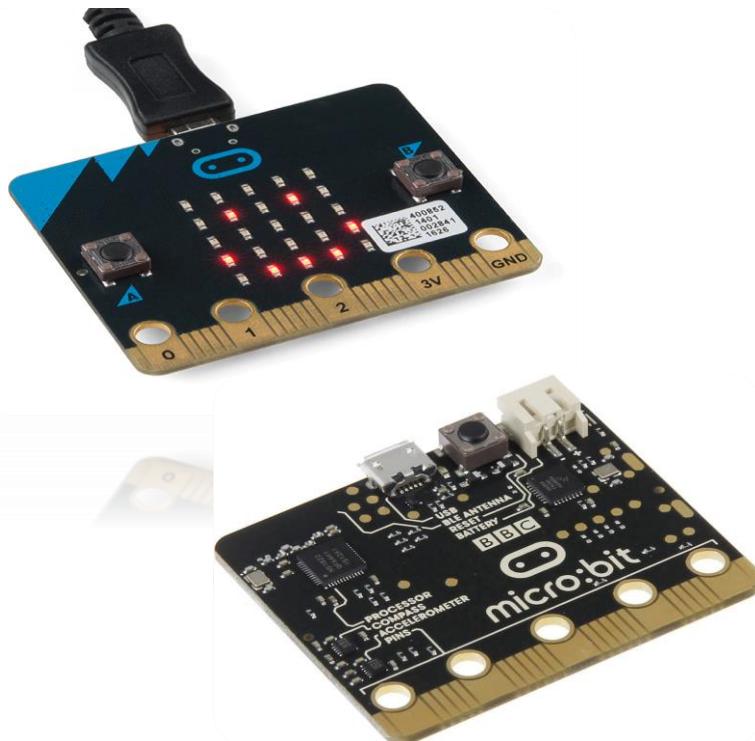
٢٠٢٥



elmanahj.com.sa

التعرف على المايكروبوت Micro : bit

المتحكمات الدقيقة هي : دوائر إلكترونية متكاملة تحتوي على معالج دقيق إلى جانب الذاكرة، وتدعم مختلف الأجهزة الطرفية القابلة للبرمجة والمستخدمة للإدخال والإخراج وتحكم في وظائف الجهاز أو النظام الإلكتروني.



تعد المتحكمات الدقيقة حاسوباً صغيراً مبسطاً على شكل رقاقة صغيرة يمكن أن يعمل بأدنى حد من المكونات الخارجية. نظراً لأنظمتها الفرعية العديدة المدمجة.

يمكن العثور على المُتحكمات الدقيقة في مجموعة كبيرة من الأنظمة والأجهزة، وتستخدم على نطاق واسع في جميع الأنظمة المدمجة مثل:

□ الساعات الذكية.

□ الكاميرات الرقمية للبوابات الذكية.

□ الأجهزة الكهربائية.

□ المركبات ذاتية القيادة.

□ بناء الروبوتات.

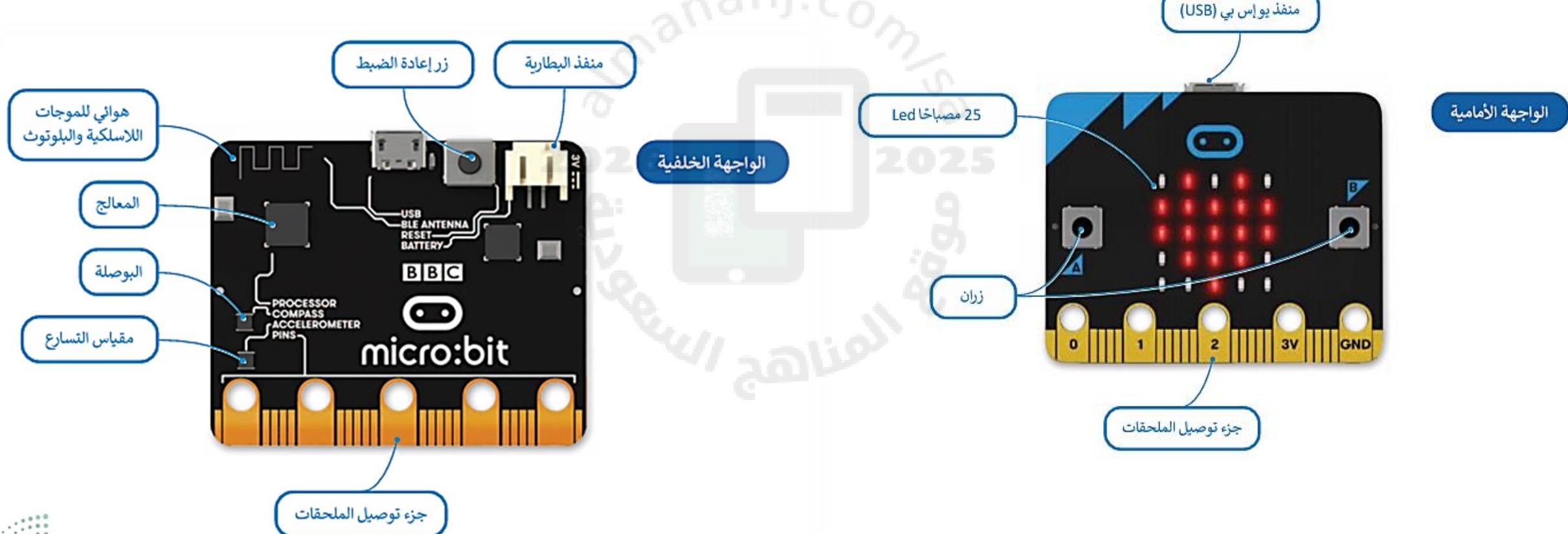
ما هو المايكروبوت؟



هو حاسب صغير الحجم تم إنشاؤه من قبل هيئة الإذاعة والتلفزيون BBC.

يمكنك استخدامه لإنشاء مشاريع رائعة وذلك من خلال توظيف المهارات البرمجية.

مكونات المايكروبوت



التطبيق العملي



مايكروسوفت ميك كود

Micros Make Code

يمكنك استخدام لغات برمجية مختلفة لبرمجة المايكروبوت كلغة مايكروسوفت ميك كود عبر الانترنت القائمة على اللبنات البرمجي أو لغة بايثون للبرمجة النصية .

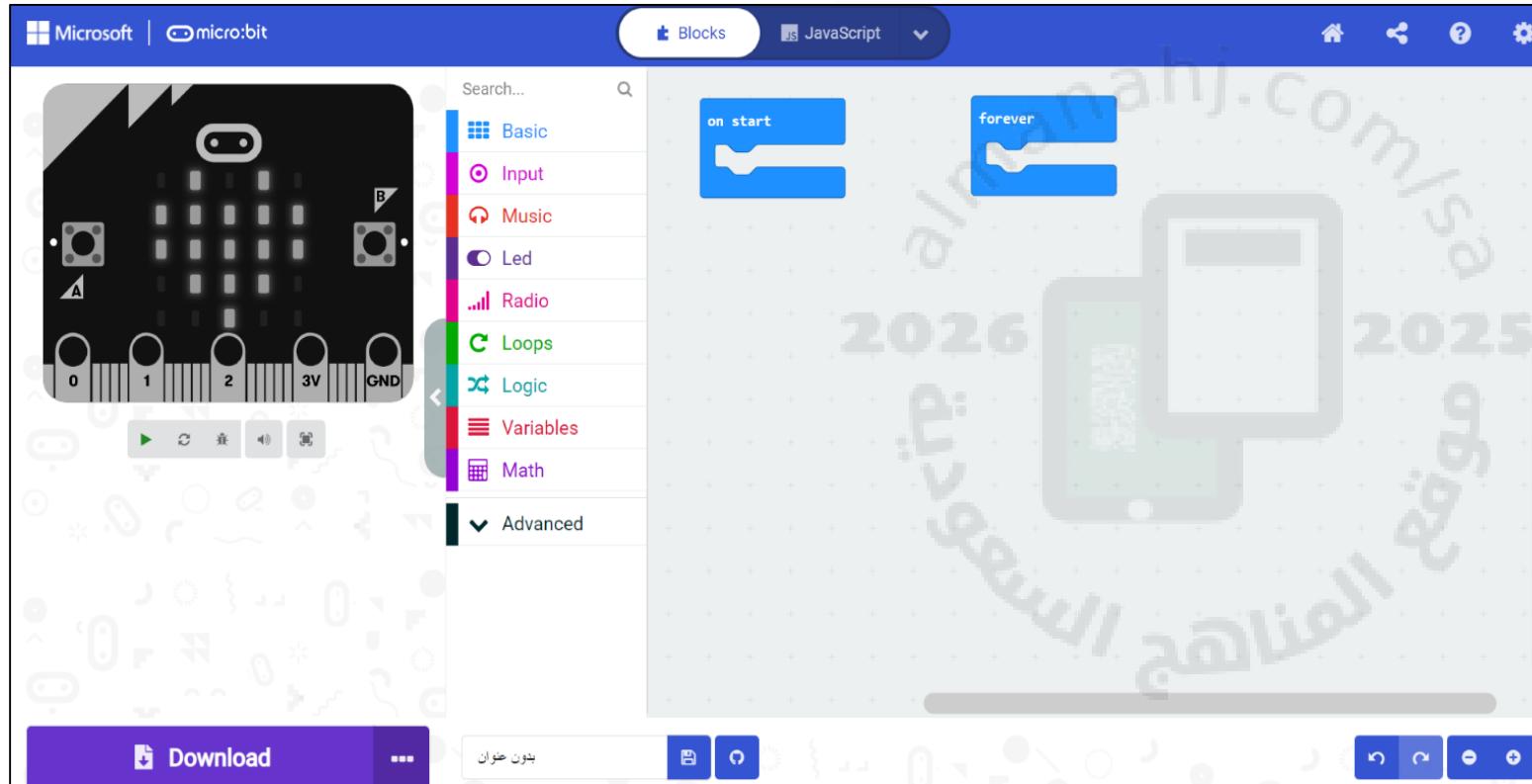


Scan Me

البرامـج

[Microsoft MakeCode for micro:bit](https://makecode.microbit.org/)

تطبيقات بديلة على الأجهزة الذكية



[Microsoft MakeCode for micro:bit](#)

واجهة ميك كود

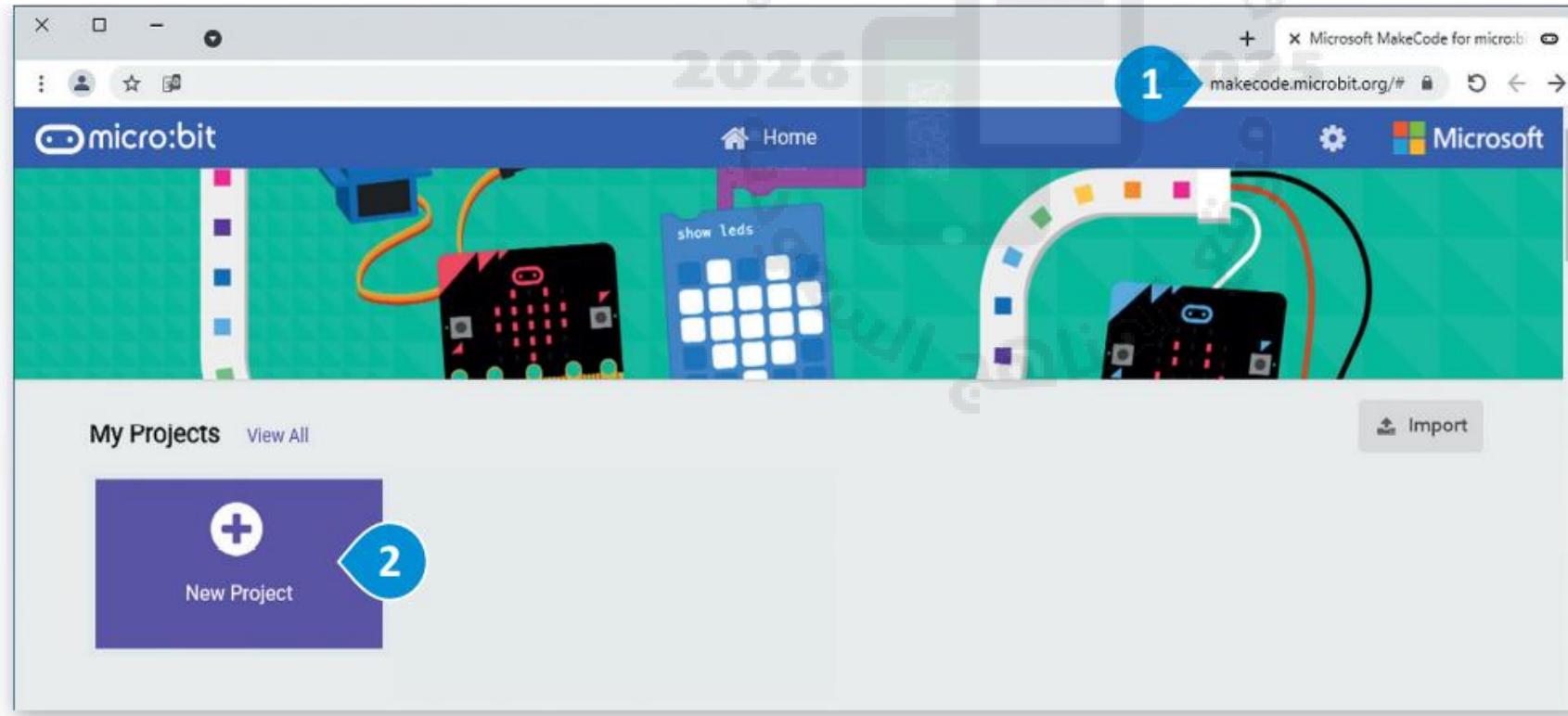
Micros Make Code





إنشاء مشروع جديد

١. اكتب makecode.microbit.org
٢. اضغط على **New Project** (مشروع جديد) لإنشاء مشروع جديد.
٣. الآن أصبح مشروعك جاهزا للبدء بالبرمجة.



3

Microsoft | micro:bit

Blocks Python

Search...

Basic

Input

Music

Led

Radio

Loops

Logic

Variables

Math

Advanced

on start

forever

Download ... Untitled

تشغيل مقطع من البرنامج بصورة متكررة (الأنهائية).

تعمل عند بدء البرنامج وقبل بدء أي حدث آخر، حيث يمكنك استخدامها لتهيئة برنامجك الخاص.

This image shows the Microsoft Scratch micro:bit editor interface. The interface includes a workspace with a micro:bit board icon, a blocks palette on the left, and a script editor at the bottom. Two blocks are currently selected: 'on start' and 'forever'. A large blue callout box points to the 'forever' block with the text: ' تشغيل مقطع من البرنامج بصورة متكررة (الأنهائية). ' (Play a segment of the program repeatedly (forever)). Another callout box points to the 'on start' block with the text: ' تعمل عند بدء البرنامج وقبل بدء أي حدث آخر، حيث يمكنك استخدامها لتهيئة برنامجك الخاص. ' (It works when the program starts and before any other event occurs, so you can use it to set up your program). A blue numbered circle '3' is in the top left corner of the image.



- إضافة لbinات إلى مقطعك البرمجي.
- الانتقال من اللبنات البرمجية إلى لغة بايثون.
- الدوال في لغة البايثون.
- حفظ المقطع البرمجي .
- تنزيل المقطع البرمجي على جهاز المايكروبوت عبر سلك يو أس بي .
- حذف اللبنات .
- صندوق أدوات الأوامر.

إضافة لبناء إلى مقطع البرمجي.



لإنشاء برنامج باستخدام اللبنانيات البرمجية:

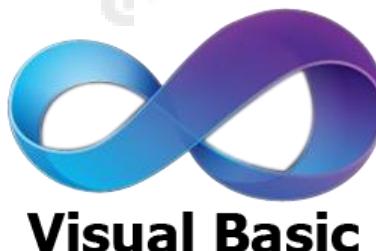
- 1 > اضغط على فئة لبنات **Basic** (أساسي).
 - 2 > اسحب وأفلت لبنة **show string "Hello!"** (إظهار السلسلة "Hello!") داخل لبنة **on start** (بداية).
 - 3 > اسحب وأفلت لبنة **show icon** (إظهار الرمز) داخل لبنة **forever** (للأبد).
 - 4 > سيعرض المحاكي رسالتك تلقائياً.
 - 5 > اضغط على زر التوقف لإيقاف المحاكي.

الانتقال من اللبنات البرمجية إلى لغة بايثون.

تعتبر هذه اللغات **لغات برمجة عالية المستوى** حيث تستخدم كلمات وحروف ورموز عادية من اللغة تتضمن لغة البرمجة عالية المستوى كلمات يجب تعلمها، وكذلك قواعد لبناء الجمل البرمجية يجب اتباعها، كما في اللغات التي يتحدثها البشر



JavaScript





لتحويل البرنامج إلى لغة بايثون:

- 1 > اضغط على القائمة المنسدلة الخاصة بلغات البرمجة.
- 2 > حدد لغة **Python** (بايثون).
- 3 > سيظهر البرنامج بلغة بايثون.

The screenshot shows the Microsoft MakeCode interface for a micro:bit. The Python code block on the right contains the following code:

```
1 basic.show_string("Hello!")
2
3 def on_forever():
4     basic.show_icon(IconNames.HEART)
5 basic.forever(on_forever)
```

A callout bubble with the number 3 and the text "تم تحويل اللينات البرمجية إلى أوامر نصية" (The programming blocks were converted into text commands) is pointing to the code block. Another callout bubble with the text "سيبقى النصف الأيسر من النافذة كما هو" (The left half of the window will remain as it is) is pointing to the micro:bit preview area.

الدوال في لغة البايثون.

الدالة عبارة عن جزء من تعليمات البرمجية التي تستخدم لمساعدتك في مهمة أو لتكرار حدث معين.

محتوى جملة الدالة في بايثون:

> يستخدم الجزء الأول من الدالة الكلمة `def` ويحتوي على تعريف الدالة.

> الجزء الثاني هو اسم الدالة.

> يوجد في نهاية رأس الدالة نقطتان.

> يأتي بعد ذلك هيكل الدالة، ويجب وضع مسافة بادئة لها.

فيما يلي دالة تطبع رسالة "Hello!" ("مرحبا!") عند الضغط على زر المايكروبوت.

يخبر `def` الحاسب أنك تريد تحديد وظيفة جديدة.

هيكل الدالة.

```
def on_button_pressed_a():
    basic.show_string("Hello!")
input.on_button_pressed(Button.A, on_button_pressed_a)
```

اسم الدالة
النقطتان

Hello!

في هذه الوحدة سوف نستخدم الدوال التالية

on_gesture_shake()

تنفذ الدالة جزء من الكود
عندما تقوم بهز
المایکروبت.

on_button_pressed_a()

تنفذ الدالة جزء من الكود
عندما يتم الضغط على زر
المایکروبت وتحریره مرة
أخرى.

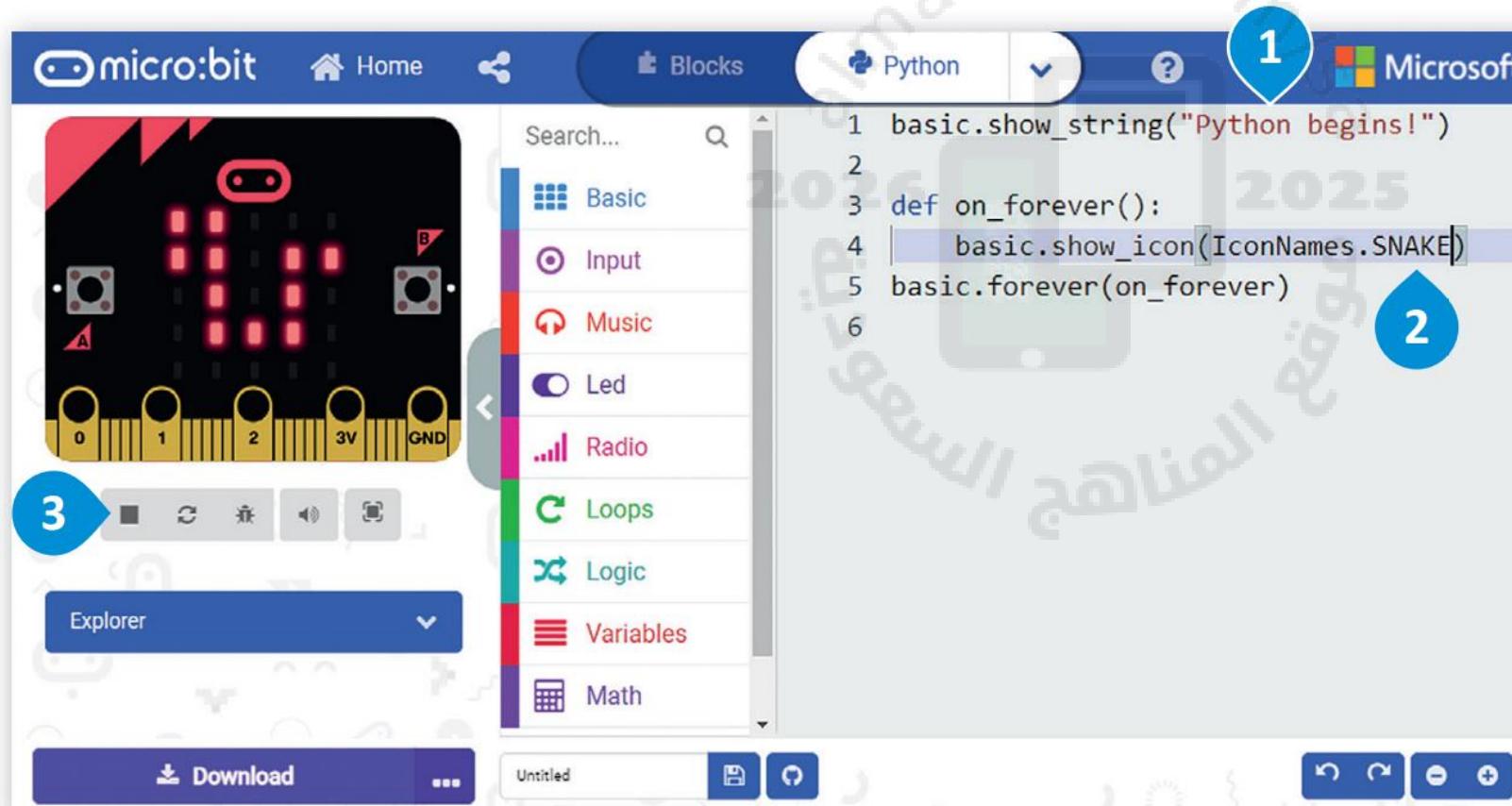
on_forever()

تنفذ الدالة جزء من
الکود بشكل لانهائي في
الخلفية.

< اضغط ضغطة مزدوجة على الأمر **show_string** (إظهار السلسلة) واستبدل كلمة **1** ("مرحبا!") بعبارة ("بايثون يبدأ!"). "Hello!"

< اضغط ضغطة مزدوجة على الأمر **show_icon** (إظهار الرمز) واستبدل كلمة **HEART** (قلب) بعبارة **SNAKE** (ثعبان). **2**

< اضغط على زر التشغيل لبدء المحاكاة. **3**



ملاحظة ..

في حال ظهور رسالة خطأ، يجب أن تتحقق من كتابة البرنامج بشكل صحيح تأكيد أولاً من عدم نسيان أي أقواس أو علامات اقتباس، وتحقق أيضاً من عدم وجود أي أخطاء إملائية.

للانتقال إلى اللبنات البرمجية



حفظ المقطع البرمجي



لحفظ البرنامج على الحاسب:

- 1 > اكتب اسمًا ل البرنامج.
- 2 > اضغط على أيقونة حفظ.
- 3 > اضغط على Done (تم) من النافذة المنسدلة، يتم حفظ البرنامج في مجلد التزيلات.



تنزيل المقطع البرمجي على جهاز المايكروبوت عبر سلك يو أس بي .

عليك أولاً توصيل المايكروبوت بجهاز الحاسب الخاص بك
باستخدام سلك USB.

بعد ذلك سيظهر كمحرك أقراص USB محمول.

لتنزيل المقطع البرمجي على المايكروبوت :

- 1 > اكتب اسمًا لمقاطعك البرمجي.
- 2 > اضغط على **Download** (تنزيل).
- 3 > افتح مجلد التنزيلات (Download) وانسخ الملف بامتداد **.hex**.
- 4 > افتح محرك أقراص **MICROBIT** (مايكروبوت) والصق الملف بامتداد **.hex**.

سيضيء المصباح الموجود على الجزء الخلفي من المايكروبوت لإظهار أن البرنامج يقوم بالنسخ.
عندما يتوقف عن الوميض، سيعمل البرنامج على المايكروبوت الخاص بك.

Microsoft | micro:bit

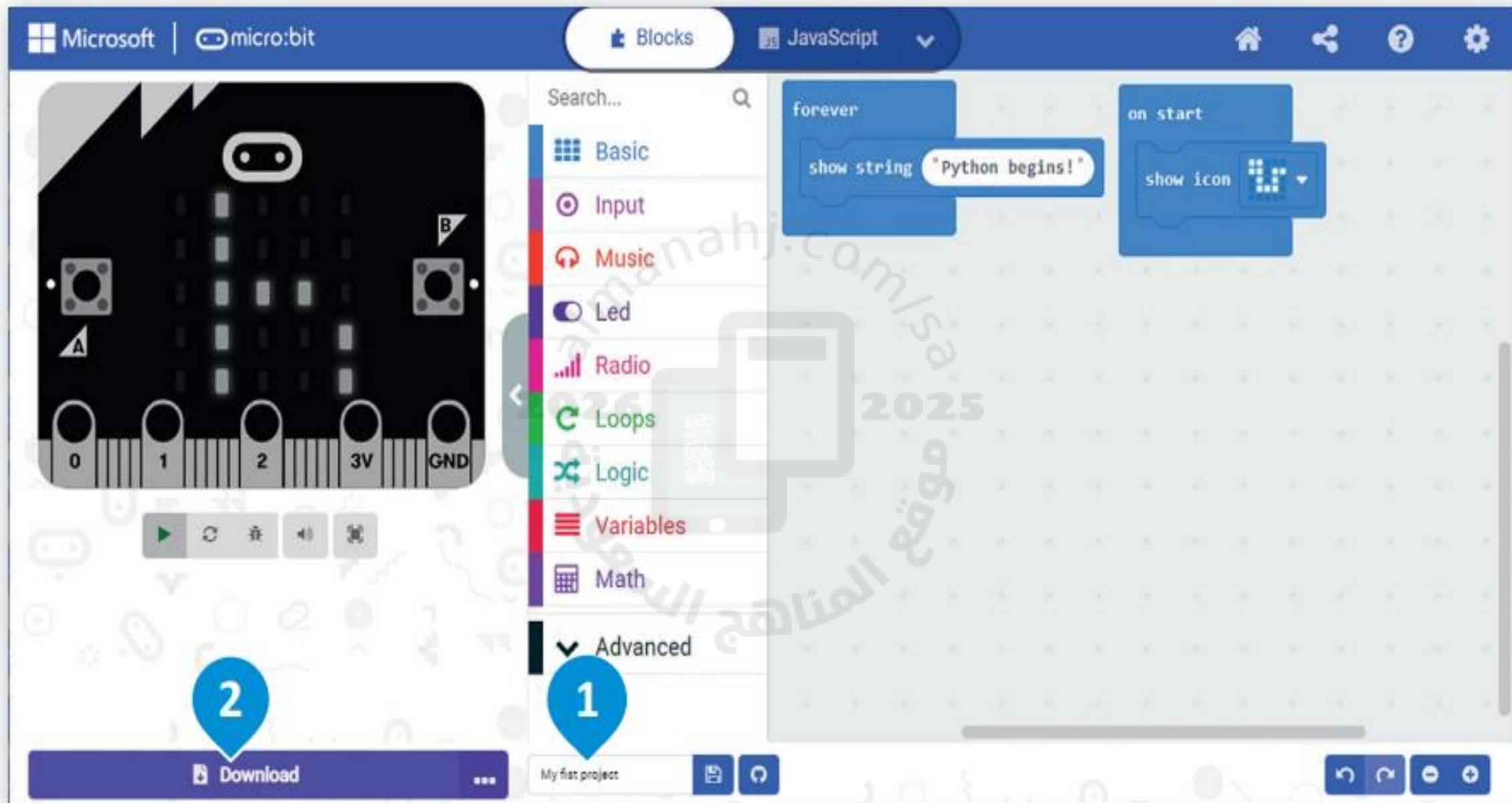
Blocks JavaScript

Search...

Basic Input Music Led Radio Loops Logic Variables Math Advanced

forever
show string "Python begins!"

on start
show icon

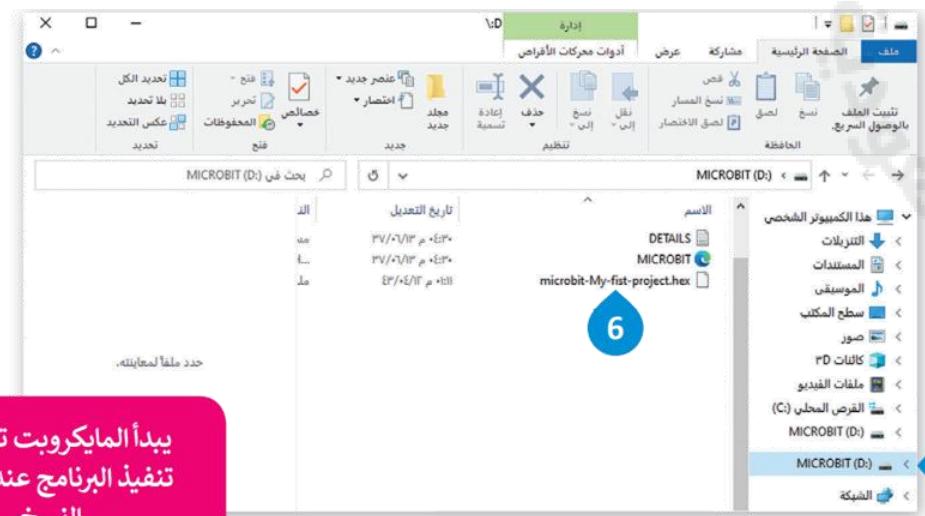


2

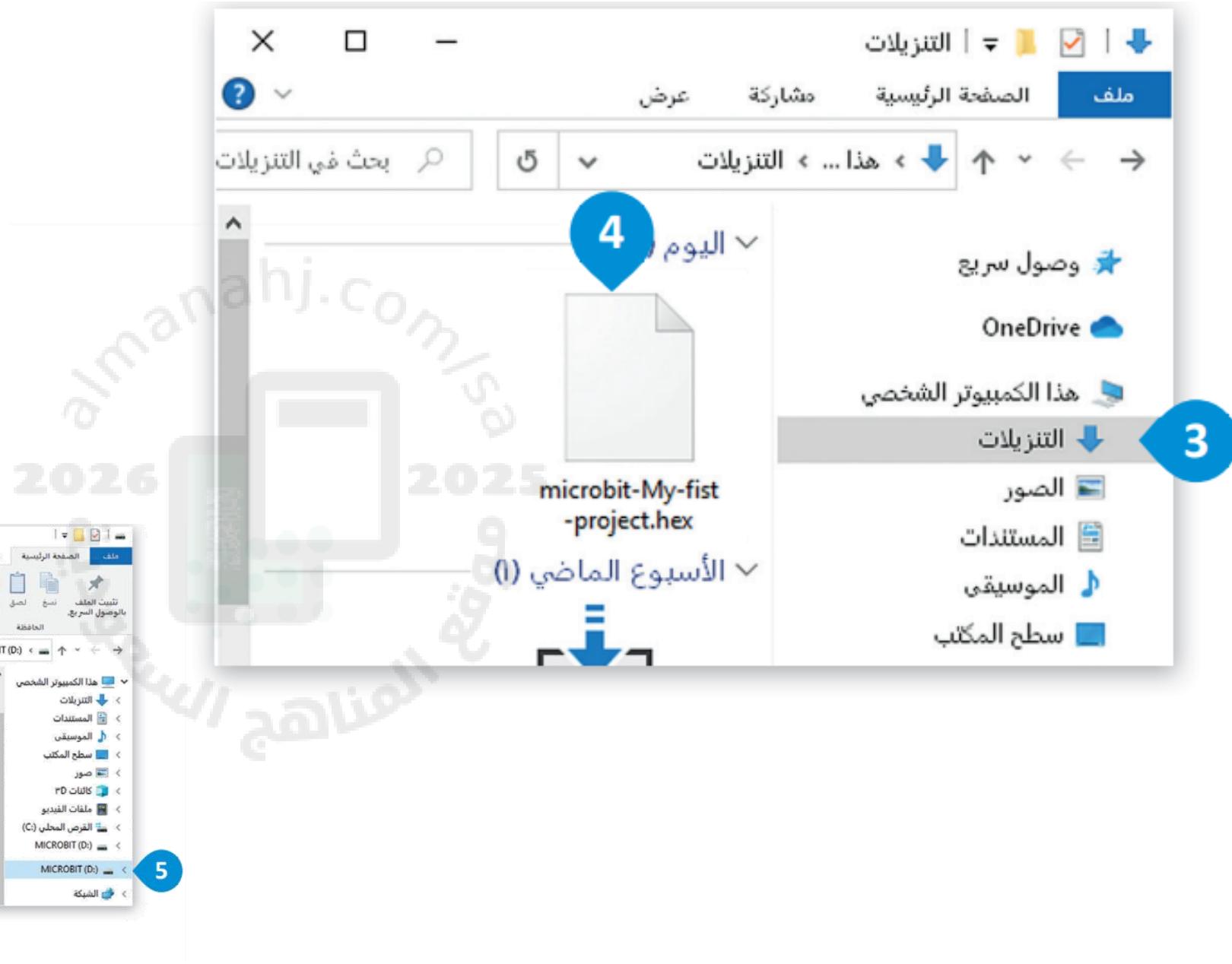
Download

My first project

1



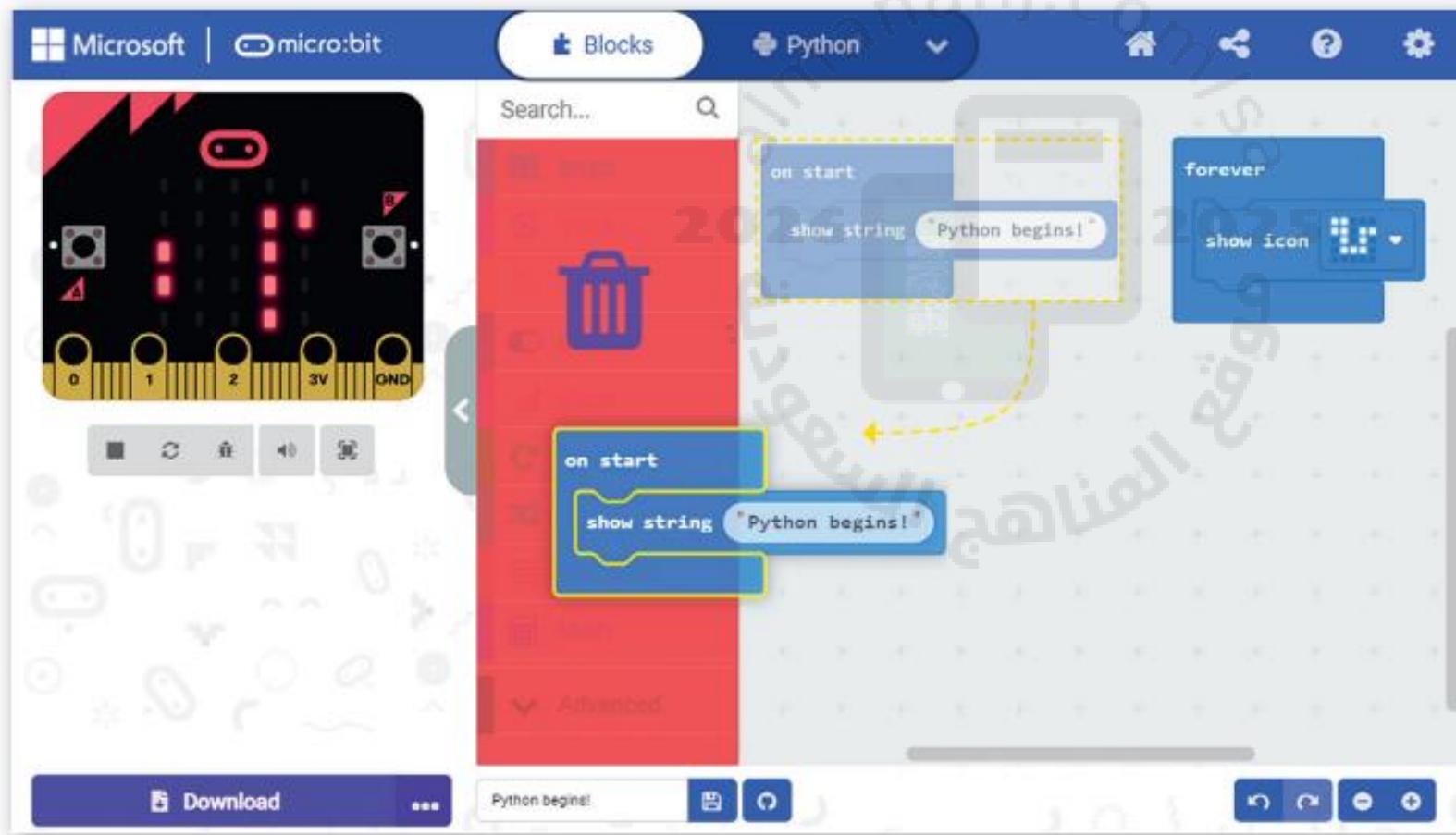
يبدأ المايكروبوت تلقائياً في تنفيذ البرنامج عند اكتمال النسخ.



حذف اللبنات

لحذف لبنة أو مجموعة من اللبنات، عليك سحبها وإفلاتها مرة أخرى في مربع أدوات

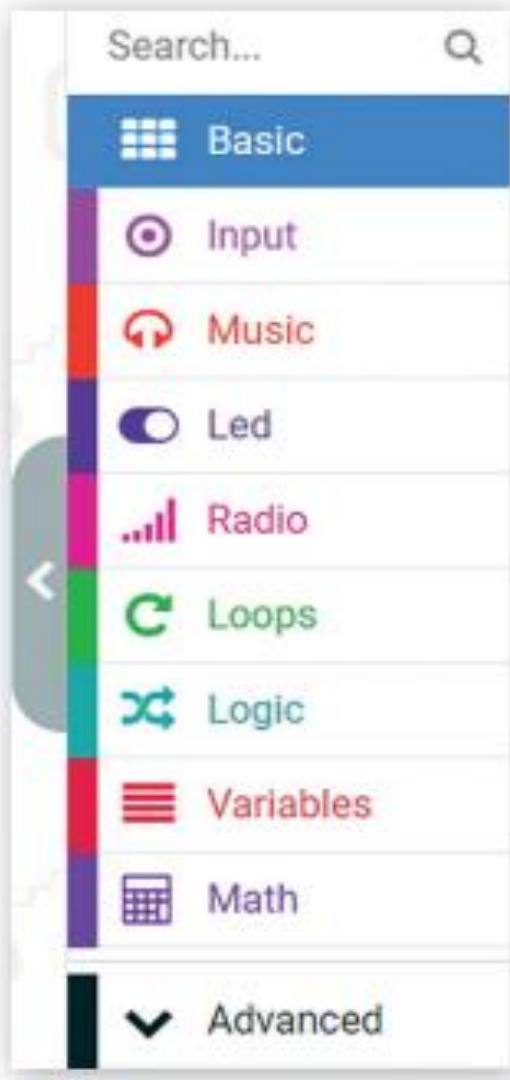
اللبنات (Blocks)



صندوق أدوات الأوامر.

يتم تنظيم دوال مايكروبوت في نطاقات بأسماء مطابقة لأسماء التبويبات، وبنفس طريقة تنظيم اللينات البرمجية ضمن فئات (تبويبات).

يُعد استدعاء إحدى دوال بايثنون المضمنة في مايكروبوت سهل الطرق لبدء استخدام مايكروبوت في بايثنون.



لإنشاء مقطع برمجي بلغة بايثون:

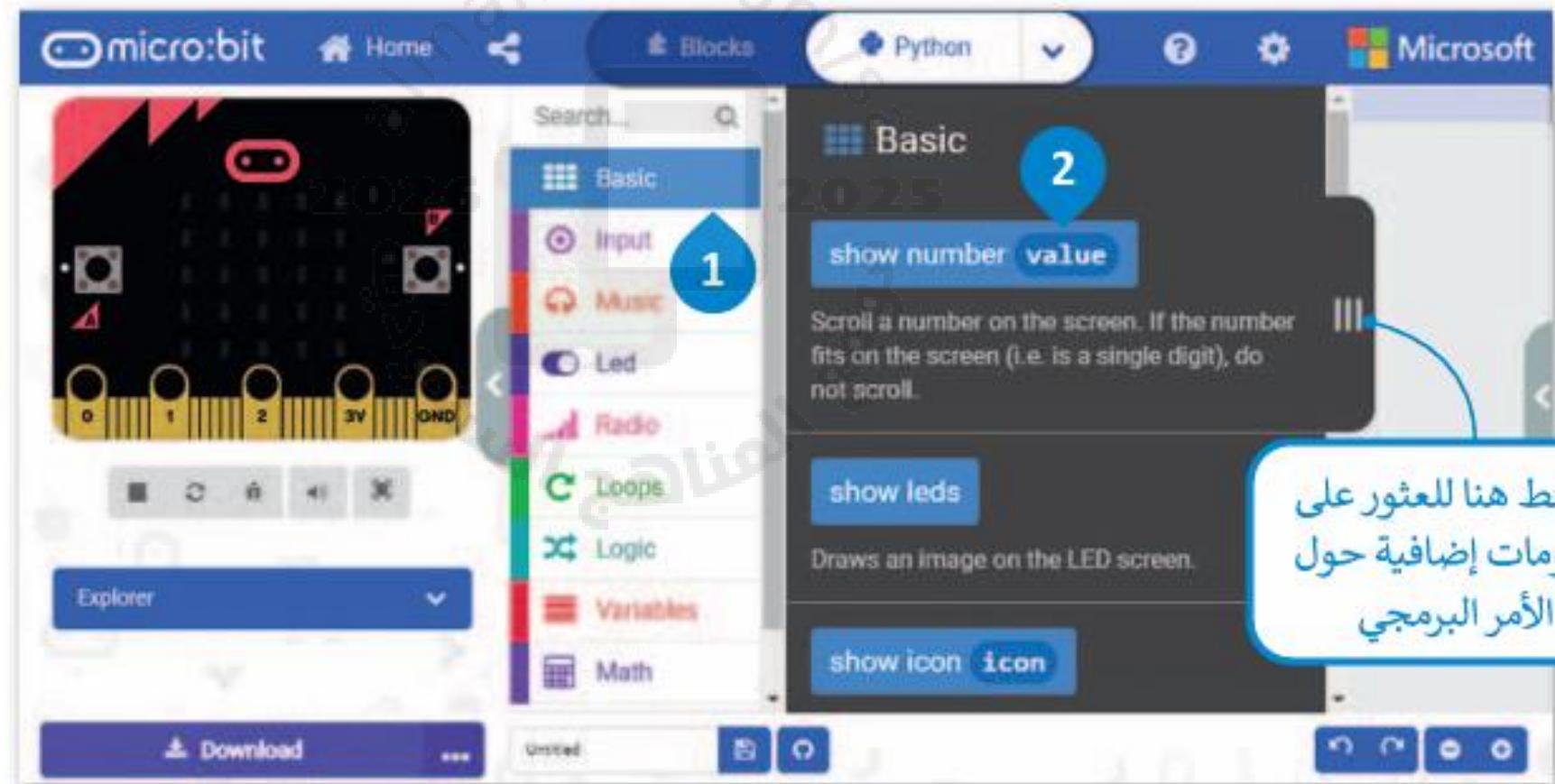
> اضغط على فئة **Basic** (أساسي) الأساسية. ①

> اسحب وأفلت أمر **show number** (إظهار الرقم) في المحرر. ②

> اكتب الرقم الذي تريده إظهاره. ③

> اضغط على زر التشغيل لبدء المحاكي الذي سيعرض الرسالة السابقة على الشاشة. ④

عند سحبك لدالة بايثون وإفلاتها في المحرر، سيتم تنفيذها عند الضغط على أيقونة التشغيل بصورة افتراضية.





يُعبر الجزء الأول قبل
النقطة عن فئة الأوامر
التي ينتمي إليها الأمر
البرمجي

احفظ عملك دائمًا.

التقويم الختامي



تقويم ختامي



<input checked="" type="checkbox"/>	1 ي تكون المايكروبوت من واجهة أمامامية فقط لا غير	
<input checked="" type="checkbox"/>	2 يمكن العثور على المتحكمات الدقيقة في الساعات الذكية	
<input checked="" type="checkbox"/>	3 يمكن استخدام محرر مايكروسوفت ميک کود لبرمجة المايكروبوت	

انتهي الجزء الأول من الدرس ☺



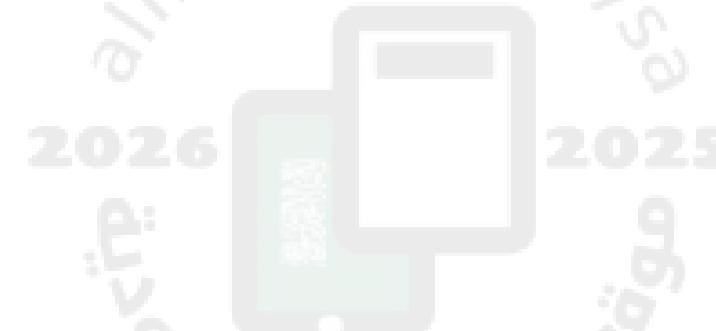
أهداف الدرس الجزء الثاني

ستتعلم في هذا الدرس

- مفهوم المايكروبت.
- مايكروسوفت ميك كود.
- إنشاء مقطع برمجي في المايكروبت .
- أمثلة برمجية .
- المتغيرات .
- تغيير الأمر .



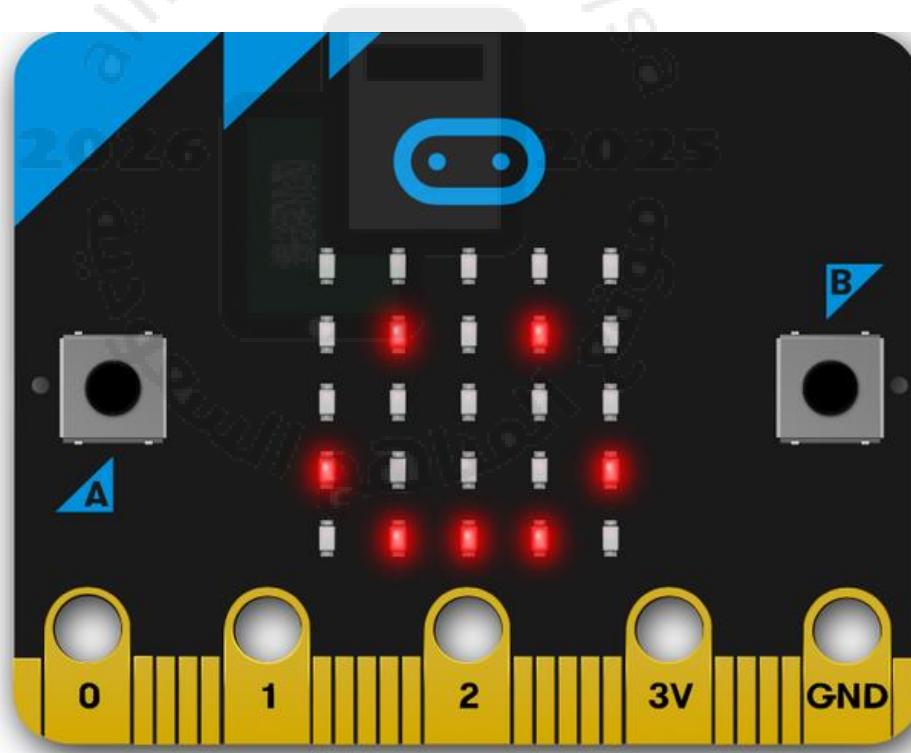
أزرار المايكروب



دالة الاهتزاز Shake

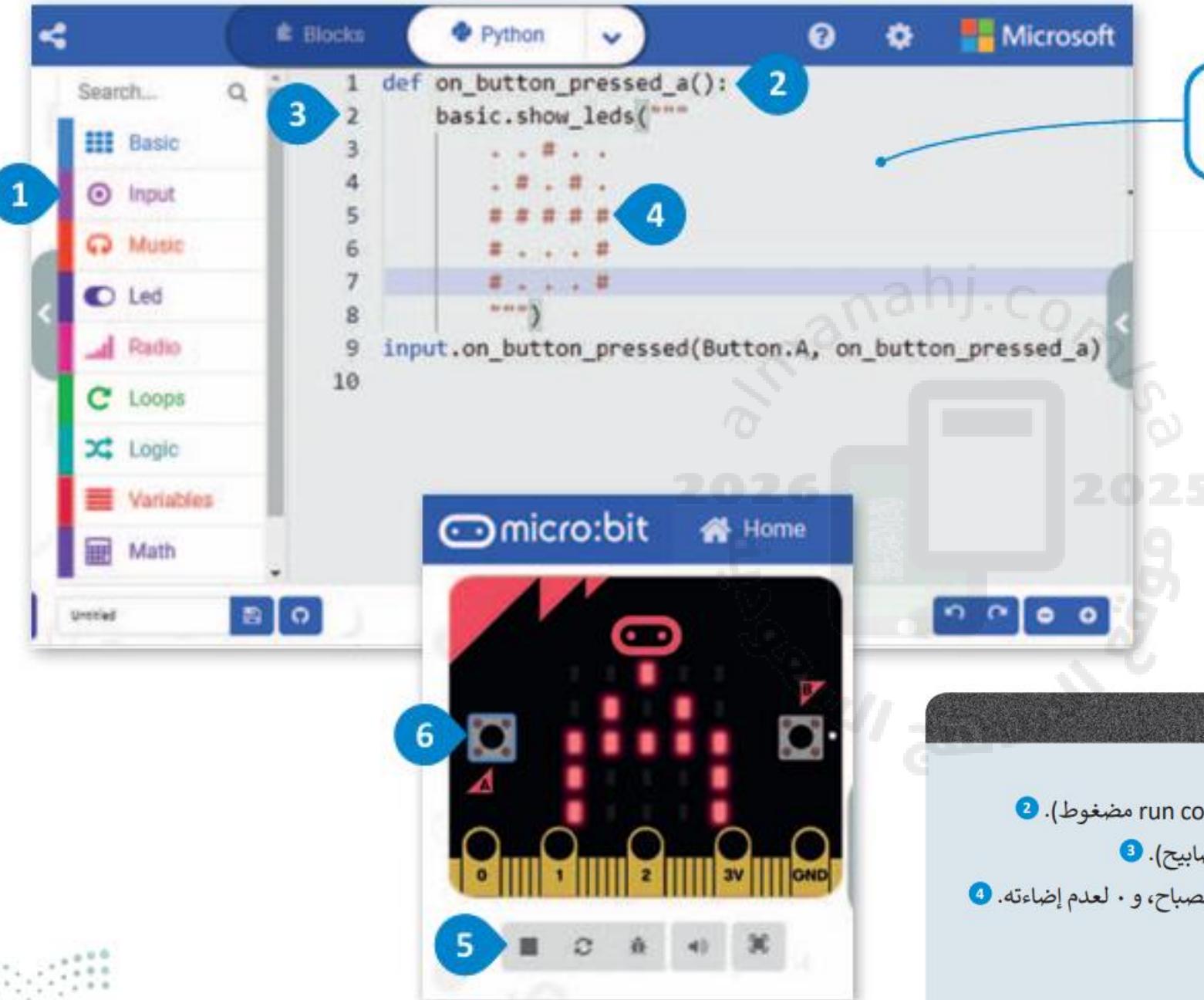
أزرار المايكروبوت

يمكنك استخدام الأزرار الموجودة في المايكروبوت عند الضغط على كلاً من الزر (A) والزر (B).





برمجة زر A

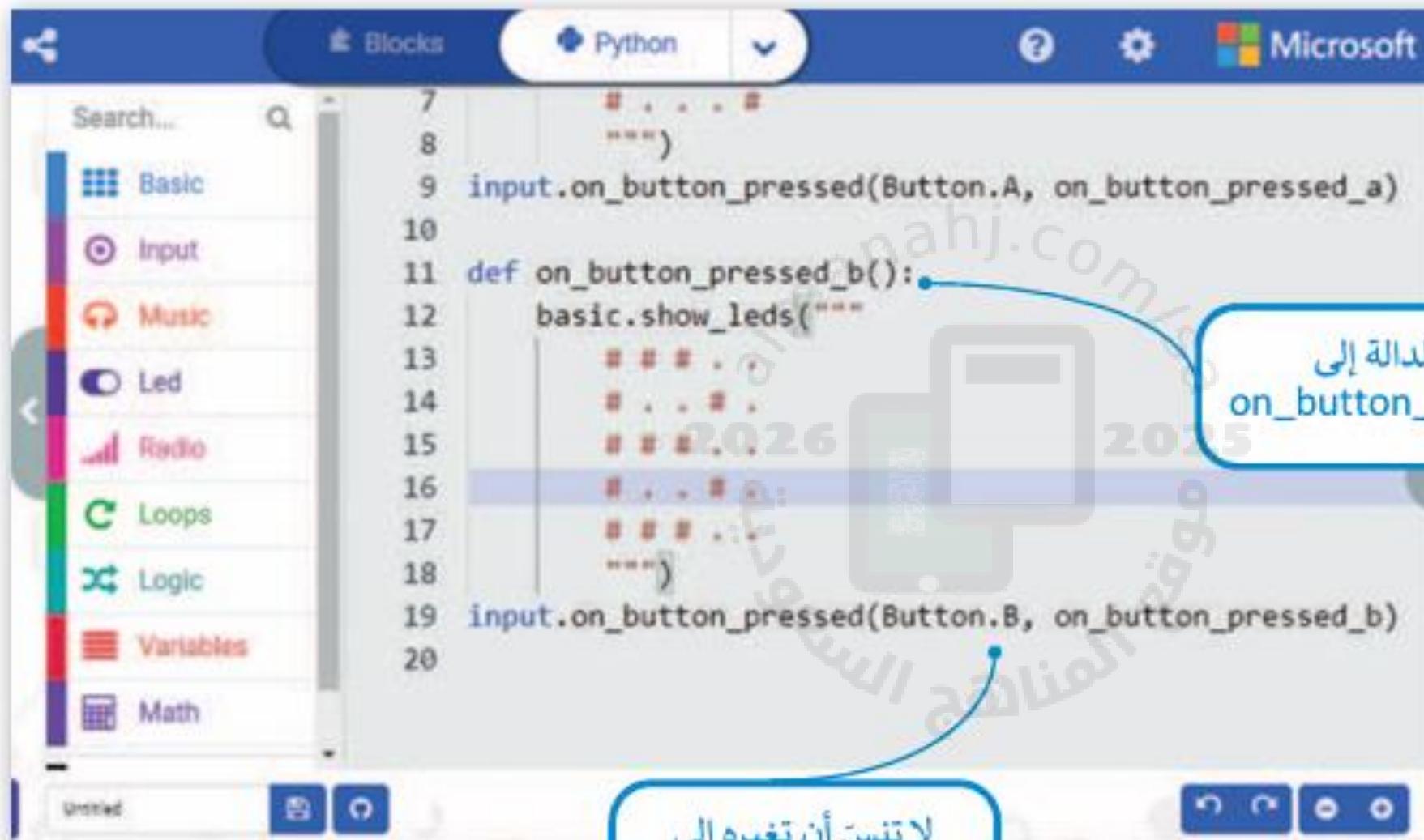


يتم تمثيل كل رمز #
بمصابح في Led

لبرمجة زر A:

- 1> اضغط على فئة أوامر **input** (الإدخال).
- 2> اسحب وأفلت أمر **run code on button pressed** (عندما يكون زر run code مضغوط).
- 3> من فئة أوامر **Basic** (أساسي)، اسحب وأفلت أمر **show leds** (إظهار المصابيح).
- 4> داخل الأمر إظهار المصابيح، أنشئ الحرف **A** في مصابيح **Led**، # لإضاءة المصابح، و . لعدم إضاءته.
- 5> اضغط على زر التشغيل لبدء البرنامج.
- 6> اضغط على الزر **A** في المحاكي لمعاينة النتيجة.

برمجة زر B



Scratch script:

```
7    # . . .
8    """
9    input.on_button_pressed(Button.A, on_button_pressed_a)
10
11 def on_button_pressed_b():
12     basic.show_leds("""
13         # # # .
14         # . . # .
15         # # # .
16         # . . # .
17         # # # .
18         # . . .
19     """)
20     input.on_button_pressed(Button.B, on_button_pressed_b)
```

The script uses the `on_button_pressed` event for Button A to call the `on_button_pressed_b` function. Inside this function, it displays a pattern of LED lights and then waits for Button B to be pressed again to repeat the sequence.

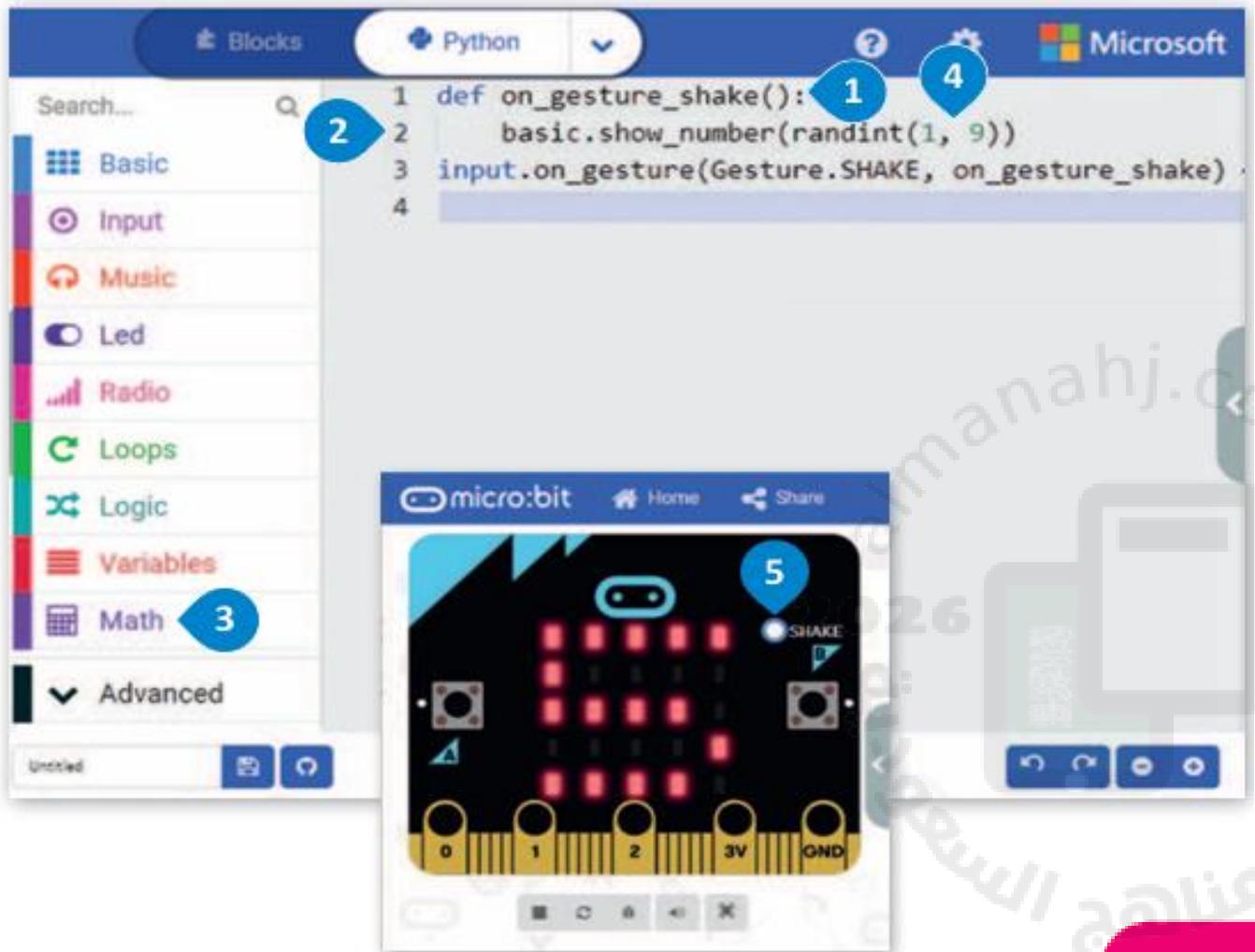
غير اسم الدالة إلى
on_button_pressed_b()

لاتنس أن تغيره إلى
(Button B)

دالة الاهتزاز Shake

يستخدم مايكروبوت مقياس التسارع الخاص به لاكتشاف أي تغيرات في الحركة.

ستنشئ برنامجاً يعرض رقمًا مختلفاً على شاشة المايكروبوت كل مرة يهتز بها المايكروبوت.



استخدام دالة الاهتزاز (Shake):

- 1 < من فئة أمر **Input** (الإدخال)، اسحب وأفلت دالة **on shake** (الاهتزاز).
- 2 < من فئة **Basic** (أساسي)، اسحب أمر **show number** (إظهار الرقم) وأفلته.
- 3 < اضغط على فئة أوامر **Math** (رياضيات).
- 4 < حدد أمر **randint**، ضعه داخل الأمر **show number** (إظهار الرقم) واضبط نطاق القيم إلى (1,9).
- 5 < اضغط على زر **SHAKE** (اهتزاز) في المحاكي لاختبار برنامجك.

يقوم الأمر **randint** بوضع رقم عشوائي داخل النطاق المحدد (بين أدنى وأعلى قيمة في النطاق).



ترتبط المتغيرات بموقع تخزين البيانات، ويتم منح كل متغير اسمًا رمزيًا، يمكن أن تتغير قيمة المتغير أثناء تنفيذ البرنامج، ويمكن أن تمثل المتغيرات أنواع مختلفة من البيانات

الفئران الرئيسان للمتغيرات هما: **الأرقام والنصوص**
تدعم لغة البايثون نوعين من الأرقام هما: **الأعداد الصحيحة والعشرية**
وتسمى المتغيرات النصية **(سلسل نصية)**.

يمكن أن يكون المتغير **اسم مختصر** مثل: **(*у* , *x*)** أو **اسم وصفي** مثل:
(age , CarModel , total_volume)

الأعداد (القيم العددية)

لا يمكن استخدام بعض الكلمات لتسمية المتغيرات لكونها كلمات خاصة أو مفاتيحية مستخدمة بواسطة لغة البرمجة، ويطلق على هذه الكلمات اسم الكلمات المحجوبة مثل:

```
def      and
return   not
while    True
else     False
global   None
if       import
```

```
MyAge=12
level=3
score=1200
```

النصوص (السلالسل النصية)

```
MyName="Salman"
EmailAddress="salmansa.bl@outlook.com"
color="Green"
```

الإعلان عن المتغيرات

المتغيرات النصية

2026 2025



الإعلان عن المتغيرات

هو عملية تعيين قيمة ومحرف (اسم فريد) للمتغير باستخدام علامة المساواة (=)

استخدام علامة المساواة (=) في البرمجة يختلف عن استخدامها في الرياضيات

والعمليات الحسابية

على سبيل المثال :

يشير استخدام علامة المساواة بهذا الشكل ($MyAge = 12$) إلى أننا نريد تمرير القيمة 12 كرقم

ليتم تعيينها إلى المتغير المسمى `MyAge`.

تعيين قيمة متغير عددي



الدرس الأول: مقدمة إلى المايكروبوت Micro:bit

تقنية رقمية

2026 2025

المناهج المنشورة



لتعيين قيمة متغير عددي:

لكل متغير في البرمجة
اسم وقيمة فريدة.

يمكنك أثناء برمجتك
بلغة بايثون كتابة الأوامر
التي تتذكرة، ولا يُعدُّ
ضروريًا اختيارها من
فئات الأوامر مرة أخرى.

1 > اضغط على فئة أوامر **Variables** (متغيرات).

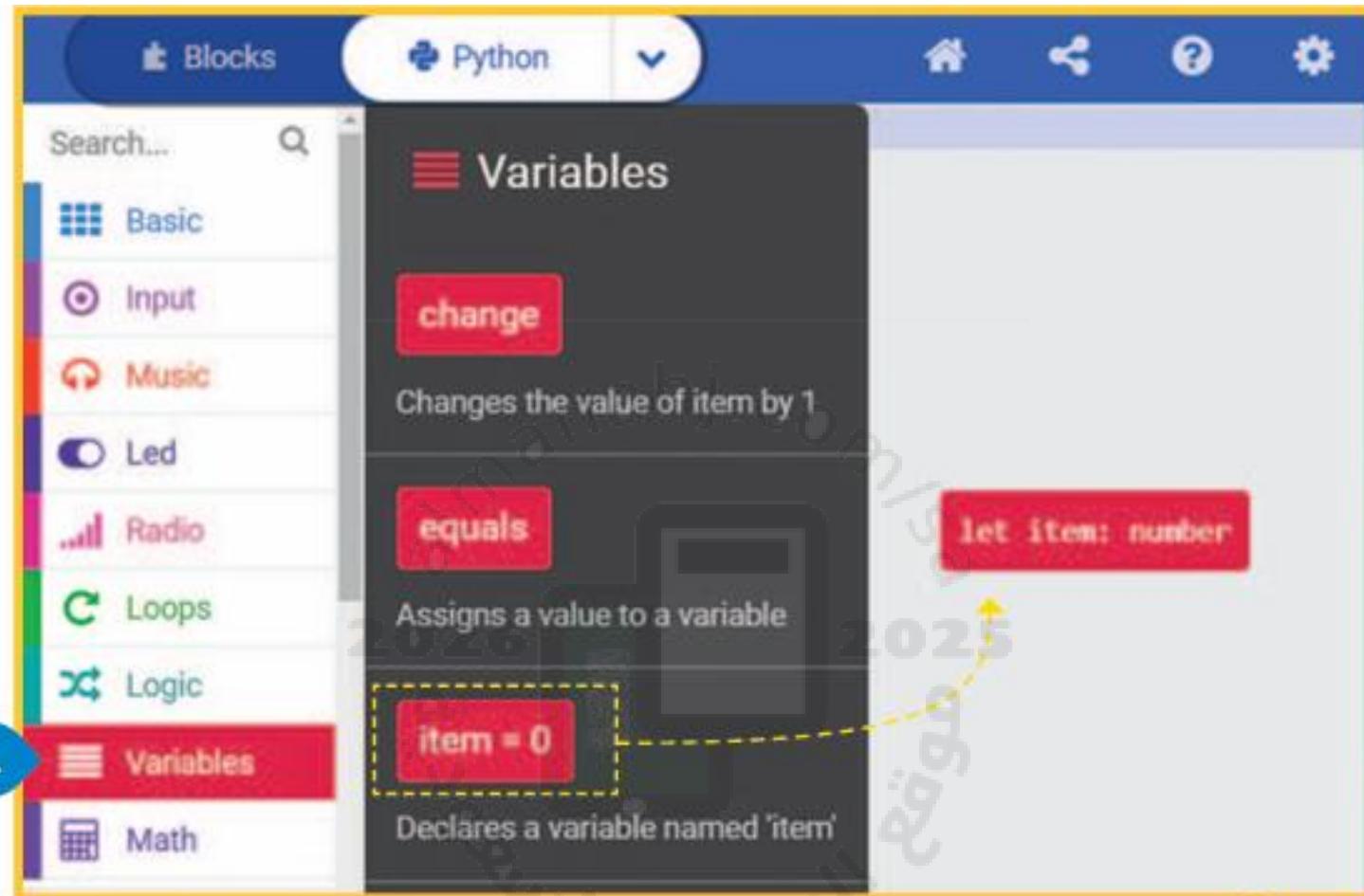
2 > اسحب وأفلت أمر **item = 0** (العنصر = 0) داخل محرر التعليمات البرمجية.

3 > اكتب وأضبط اسم المتغير ليكون **MyAge = 12** (عمرى = 12).

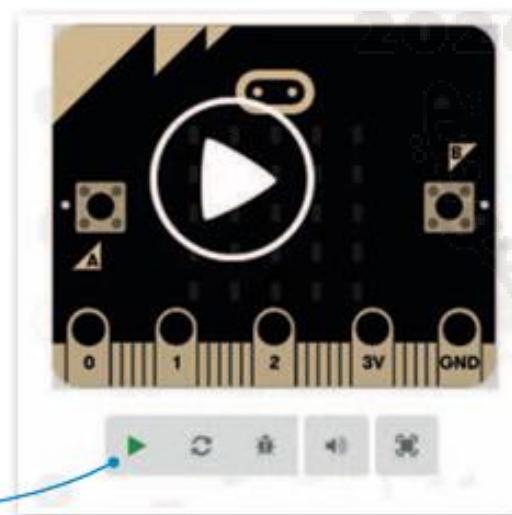
4 > من فئة أوامر **Basic** (أساسي) اسحب وأفلت أمر **show number** (إظهار الرقم).

5 > اكتب اسم المتغير داخل الأقواس.





The image shows the Scratch script editor with the 'Python' tab selected. The script contains a single line of code: '1 item = 0'. A blue arrow labeled '2' points to the 'item' variable in this line, indicating the variable being declared.



اضغط تشغيل
لاختبار عمل المقطع
البرمجي.

Blocks Python ? Microsoft

Search... 3 1 MyAge = 12
2 basic.show_number(MyAge)

4 5 يمكنك إدخال
أي قيمة تريدها.

Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
Advanced

Untitled

Untitled

المتغيرات النصية

لا يقتصر استخدام المتغيرات على تخزين الأرقام فقط بل يمكن استخدامها لتخزين النصوص أيضاً، تسمى المتغيرات التي تخزن النصوص **متغيرات نصية** ويجب وضع النص فيها داخل علامات الاقتباس (" ")

تعيين قيمة لمتغير نصي :

- لتعيين قيمة لمتغير نصي :
- > من فئة الأوامر **Variables** (المتغيرات) اسحب الأمر **item = 0** (العنصر = 0) وأفلته، اكتب اسم المتغير وقيمتة. ①
 - > من فئة **Basic** (أساسي)، اسحب وأفلت أمر **show string** (إظهار السلسلة). ②
 - > اضغط بالفأرة داخل القوسين وأحذف علامة التعجب، ثم اكتب "+" واسم المتغير **MyName** (اسمي). ③

يجب دائمًا عند استخدام المتغيرات النصية وضع النص بين علامتي الاقتباس " ".





يمكن استخدام المتغيرات لأداء مجموعة متنوعة من المهام.

ويقوم الأمر تغيير (change) الموجود في فئة أوامر المتغيرات (Variables)

بتغيير قيمة المتغير المحدد بالقيمة المعيينة التي يتم إدخالها.

يقتصر استخدام هذه الطريقة على المتغيرات العددية.

زيادة قيمة المتغير

تقليل قيمة المتغير

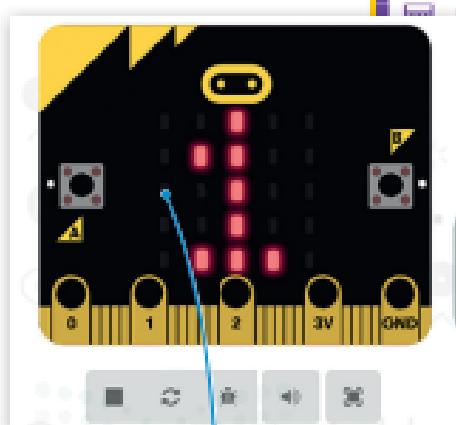
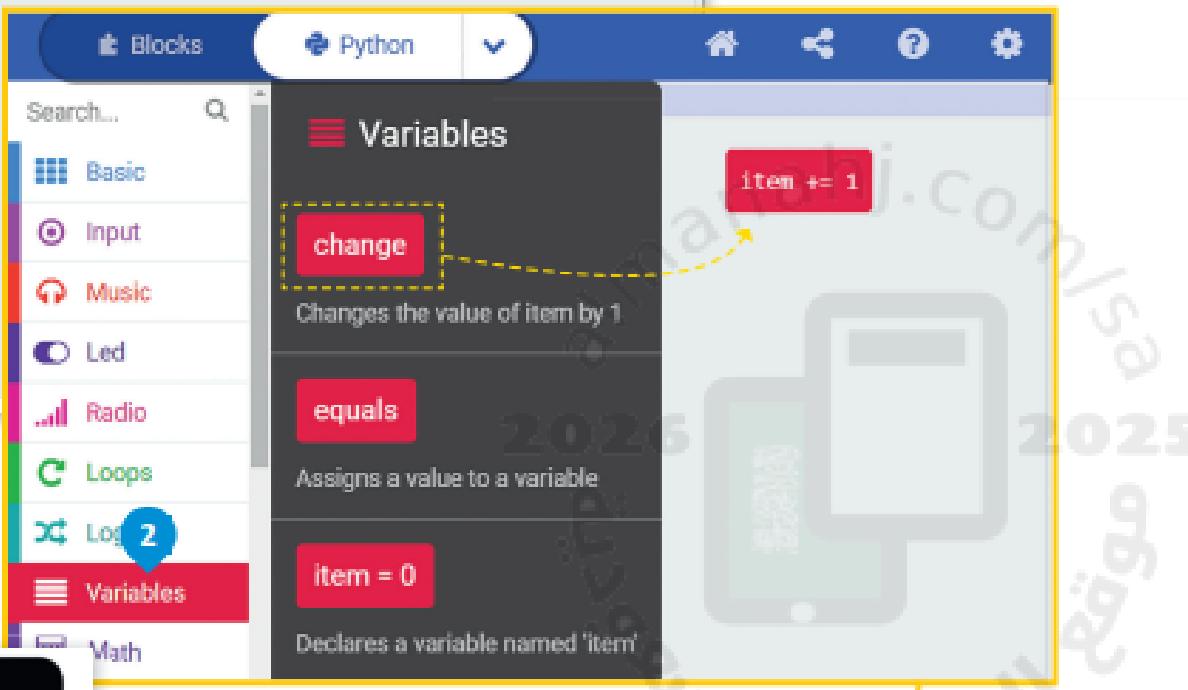
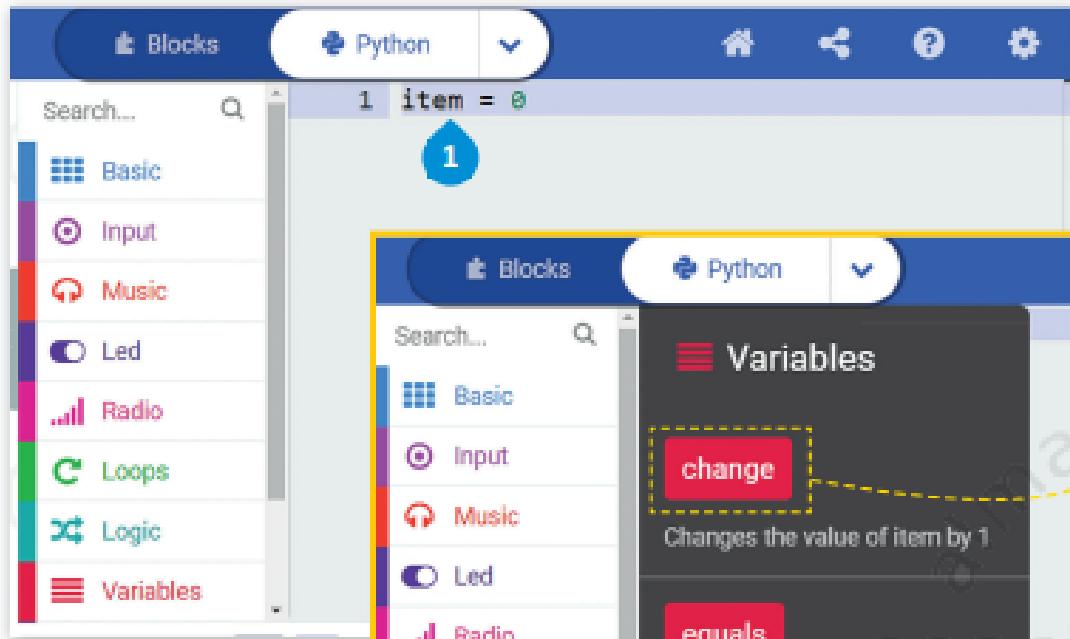
عدد item +=

عدد item -=

في المثال التالي يقوم البرنامج زيادة قيمة المتغير العنصر بمقدار 1 :

لتغيير قيمة متغير عددي:

- > من فئة الأوامر **Variables** (المتغيرات) اسحب الأمر **item = 0** (العنصر = 0) وأفلته. **1**
- > اضغط على فئة الأوامر **Variables** (المتغيرات). **2**
- > اسحب وأفلت الأمر **change** (تغيير). **3**
- > من فئة الأوامر **Basic** (أساسي) اسحب وأفلت الأمر **show number** (إظهار الرقم) **4** واتكتب داخل الأقواس اسم المتغير **item** (العنصر).



اضغط على زر التشغيل وشاهد النتيجة.

استبدل علامة (+) الموجودة في الأمر **item + = 1**
بعلامة (-) ثم فّعل الكود مرة أخرى ولاحظ الفرق

المتغيرات المحلية والمتغيرات العامة

يتم تصنيف المتغيرات إلى متغيرات محلية ومتغيرات عامة بناءً على نطاقها.

ونطاق المتغير هو الجزء من البرنامج الذي يمكن من خلاله الوصول إلى المتغير ورؤيته واستخدامه.

المتغيرات العامة

يتم تعريف المتغيرات العامة خارج أي دالة ويمكن الوصول إليها بشكل عام في البرنامج بأكمله، وبمعنى آخر يمكن استخدامها في أي مكان في البرنامج وليس فقط في النطاق الذي تم تحديده، كداخل الدالة على سبيل المثال.

```
variable = 0

def name():
    global variable
    command
close.def()
```

نطاق المتغير العام

المتغيرات المحلية

يتم تعريف المتغيرات المحلية داخل دالة ولذا تنتهي فقط إلى هذه الدالة المحددة، ولا يمكن الوصول إليها إلا من خلال تلك الدالة التي تم تعريفها داخلها.

```
variable = 0

def name():
    command
close.def()
```

نطاق المتغير المحلي

أنشئ برنامجًا بحيث تتغير قيمة المتغير `myVar` بمقدار 1 في كل مرة تضغط فيها على زر من المايكروبوت. سترسل المايكروبوت رسالة إلى المتصفح كلما تم الضغط على زر A. ستستخدم الأمر عام `global` للدلالة على أن `myVar` هو متغير عام، مما يعني أن تعيين قيمة `myVar` داخل الدالة يغير ما سيتم عرضه عند استخدام القيمة `myVar` في القسم الرئيس من البرنامج.

أنشئ البرنامج التالي:



يمكن الوصول للمتغيرات العامة في المقطع البرمجي من جميع الدوال.

عرف المتغير قبل استخدامه

نطاق المتغير العام `myVar`

```
Blocks Python Microsoft
1 myVar = 0
2
3 def on_button_pressed_a():
4     global myVar
5     myVar += 1
6     basic.show_number(myVar)
7 input.on_button_pressed(Button.A, on_button_pressed_a)
```

الله أنعم علينا بالعقل لنتعلم ونخترع.

البرمجة والتقنية هي من صور استثمار العقل الذي أمرنا الله باستخدامه في الخير.

يمكن برمجة المايكروبوت لخدمة الناس (مثلاً، صنع أدوات مساعدة لذوي

الاحتياجات الخاصة) وهذا من باب العمل الصالح وخدمة المجتمع،

وهو مما يحث عليه الإسلام.

القيمة الدينية

تعلم المايكروبوت والبرمجة يساهم في بناء مجتمع معرفي وتقني قوي، مما يعزز منازع الاستقلال التكنولوجي لوطنا.

يمكن استخدام المايكروبوت لصنع مشاريع تخدم البيئة، الصحة، الأمن...
مما يساهم في تنمية الوطن ورفعه.

القيمة الوطنية

التقويم الختامي



تقويم ختامي



<input checked="" type="checkbox"/>	لا تختلف استخدام علامة المساواة (=) في البرمجة عن استخدامها في الرياضيات	١
<input checked="" type="checkbox"/>	لا يقتصر استخدام المتغيرات على تخزين الأرقام فقط	٢
<input checked="" type="checkbox"/>	يتم تعريف المتغيرات المحلية داخل دالة ولذا تنتهي فقط إلى هذه الدالة المحددة	٣



انتهت الحصة

