

Light

Textbook

Light



Outputs

An output is a place where information is produced on a computer. For example, the LED light display is an example of an output. It shows the code you create using its 25 LEDs arranged in a 5x5 grid on the front of the micro:bit. You can show words, pictures, and more using these lights. These LEDs also act as light sensors to sense light. Learn more by watching the following video:



Light Sensor

Another example of a variable for the micro:bit is light. This means the micro:bit has a light sensor in its processor. The light information collected by the micro:bit is an example of a variable since that information can be remembered, used, and changed with the micro:bit. The light sensor measures the amount of light in an area and responds by turning a light on or off. We see these used in everyday life, like when street lamps turn on when it gets dark, or when a nightlight turns off when the room's light turns on.

Coding with the Light Sensor

Let's practice coding with the light sensor together. Our goal is to create a program that tells the micro:bit, "if the light level drops then display an icon as a night light."

To do this, start with a forever block, since we want this night light code to work forever and not just once!

We also need to create a variable named `light_level` that holds the value of how much light the micro:bit is registering: `input.light_level()`

```
1 def on_forever():
2     light_level = input.light_level()
3
4 basic.forever(on_forever)
5
```

The value of 120 is the value of a normally lit room. So if I'm trying to build a nightlight that turns on in the dark I might say, "If light level < 120 then." So let's add an if statement with the appropriate conditional. If the light level is more than 120 I want the micro:bit to not show anything, so we will clear the screen.

```
1 def on_forever():
2     light_level = input.light_level()
3
4     if input.light_level() < 120:
5         basic.show_icon(IconNames.DIAMOND)
6     else:
7         basic.clear_screen()
8
```

```
9 basic.forever(on_forever)
10
```

Try playing around with different things you can do with light!

Adopted from microbit.org platform

Critical Thinking Questions

- Beyond streetlights and nightlights, what's another everyday device or system that could use a light sensor to automatically change its behavior based on how bright or dark it is? Explain how this automated response would be beneficial.
- Imagine you're designing a greenhouse. How could a program use a light sensor to decide when to automatically open or close vents, or turn on special grow lights, to help plants thrive? Why is the ability to "remember" (variable) and compare the light level important for this system?
- How does this dual capability of some components make technology more efficient or versatile in a way that separate input and output devices might not?

Questions (10)

1. A student wants to create a program that shows a picture when it gets bright instead of dark. How should they modify the condition in this code?

MULTIPLE CHOICE

```
if input.light_level() < 120: basic.show_icon(IconNames.DIAMOND) else: basic.clear_screen()
```

Choose the correct answer:

- A. Change < to >
- B. Change 120 to a smaller number
- C. Add more icons to show
- D. Remove the else statement

2. Why does the nightlight code use a forever block instead of running just once?

MULTIPLE CHOICE

Choose the correct answer:

- A. To make the light brighter
- B. To save battery power
- C. To keep checking the light level continuously
- D. To display multiple icons at once

3. What will happen if you change the number 120 to 50 in the nightlight code?

```
def on_forever(): light_level = input.light_level() if input.light_level() < 120: basic.show_icon(IconNames.DIAMOND) else: basic.clear_screen() basic.forever(on_forever)
```

Choose the correct answer:

- A. The nightlight will turn on in brighter rooms
- B. The nightlight will turn on in darker rooms only
- C. The nightlight will be dimmer
- D. The nightlight will flash on and off

4. A programmer wants to create a "bright light alarm" that shows a warning when the light level goes above 200. Which condition should they use?

MULTIPLE CHOICE

Choose the correct answer:

- A. if input.light_level() < 200:
- B. if input.light_level() > 200:
- C. if input.light_level() = 200:
- D. if input.light_level() + 200:

5. What is the main reason for creating a variable called light_level in the code?**Choose the correct answer:**

- A. To make the code run faster
- B. To store and use the current light reading
- C. To change the brightness of the LED display
- D. To count how many times the light changes

6. Based on the passage, what dual purpose do the LEDs serve on the micro:bit?**Choose the correct answer:**

- A. They show information and sense temperature
- B. They show information and sense light
- C. They play music and show pictures
- D. They sense light and play sounds

7. A student notices their nightlight program turns on and off rapidly in a room with changing light. What is most likely causing this problem?

MULTIPLE CHOICE

Choose the correct answer:

- A. The forever block is running too fast
- B. The light level is very close to the threshold value of 120
- C. The diamond icon is too bright
- D. The variable is not working correctly

8. What would happen if you removed the else part of the nightlight code?

MULTIPLE CHOICE

```
if input.light_level() < 120: basic.show_icon(IconNames.DIAMOND) else: basic.clear_screen()
```

Choose the correct answer:

- A. The program would not work at all
- B. The icon would stay on screen even in bright light
- C. The icon would be brighter
- D. The program would run faster

9. Looking at the code structure, what is the purpose of the `input.light_level()` function?

MULTIPLE CHOICE

Choose the correct answer:

- A. To turn the LEDs on and off
- B. To get the current light measurement from the sensor
- C. To set how bright the display should be
- D. To count how many LEDs are working

10. Debug the following code:

DEBUG CODE

Code to Debug:

```
1 def on_forever():
2     light_level == input.light_level()
3
4 basic.forever(on_forever)
```

Robotics Challenges (5)

1. Sunrise Sunset

Challenge

Textbook

Sunrise Sunset

Create a program that displays a rising sun with rising light. Have you ever seen lamps that slowly turn on just like a sunrise? Create a similar lamp with the micro:bit.

Create the forever function, since we want this to always work.

Create the light_level variable and set it equal to the light in the room.

If the light_level is greater than 150, display the full sun.

Elif the light_level is greater than 125, display the sun going down a bit.

Elif the light_level is greater than 100, display the sun going down a bit further.

Elif the light_level is greater than 75, display the sun going down a bit further.

Elif the light_level is greater than 50, display the sun going down almost all the way.

Else, clear the screen.

This will allow the micro:bit to show a sun rising as the light increases.

Step 1

Create the forever function.

Sunrise Sunset Step 1 of 5

1

Next

Toolbox

Search

Basic

start

forever

Oops, there is a problem converting your code to Python.

We are unable to convert your code to Python.

Download

2. Flashlight

Challenge

Textbook

Flashlight

Create a program that will work like a flashlight! The micro:bit will turn on its lights if the light level drops below 75.

Create a forever function.

Create an if statement to see if the light level drops below 75.

If it does, turn on the LEDs.

Else, clear the screen.

Requirements

Create the forever function.

Inside the forever function, create a variable named light_level.

Create an if statement to see if the variable named light_level drops below 75.

Download

def on_forever():

pass

basic.forever(on_forever)

Explorer

3. Light Melodies

Challenge

Textbook

Light Melodies

Create a program that plays different melodies, depending on the amount of light in the room!

If the light level is less than 50, play a melody.

Elif the light level is greater than 150, play a different melody.

Else, display an icon.

Requirements

Create the forever function.

Inside the forever function, create a variable named light_level.

Create an if statement to see if the variable named light_level drops below 50.

If the variable named light_level drops below

Download

1 def on_forever():
2 pass
3 basic.forever(on_forever)
4

Explorer

4. Light Scale

Challenge

Textbook

Light Scale

In music, a scale is a musical melody that plays notes in order. Create a program that plays a scale with increasing light.

If the light level is equal to 100, play a Middle C note (262).

Elif the light level is equal to 110, play a Middle D note (294).

Elif the light level is equal to 120, play a Middle E note (330).

Elif the light level is equal to 130, play a Middle F note (349).

Elif the light level is equal to 140, play a Middle G note (392).

Elif the light level is equal to 150, play a Middle A note (440).

Elif the light level is equal to 160, play a Middle B note (494).

Elif the light level is equal to 170, play a High C note (523).

Else, display the EIGHTH_NOTE icon.

Reminder: To check if values are equal, you need to use TWO equals signs ==

Step 1

Create the forever function.

Light Scale Step 1 of 11

1 Next

Download

Toolbox

1 def on_forever():
2 pass
3 basic.forever(on_forever)
4

Explorer

5. Weather Watcher

Challenge

Textbook

Weather Watcher

Create a program that checks the weather!

If the light level is above 120, it will display the string **Sunny**

Elif the light level is below 100, it will display the string **Cloudy**

Else, it will clear the screen.

Requirements

- Create the forever function.
- Inside the forever function, create a variable named `light_level`.
- Create an if statement to see if the variable named `light_level` goes above 120
- If the variable named `light_level` goes above 120 display the string

Step 1

Create the forever function.

Weather Watcher Step 1 of 5



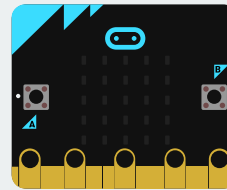
1

Next

Toolbox

Search

```
1 def on_forever():
2     pass
3     basic.forever(on_forever)
4
```



Download

Answer Keys & Solutions

Questions

1. A student wants to create a program that shows a picture when it gets bright instead of dark. How should they modify the condition in this code?

MULTIPLE CHOICE

Correct Answer:

- | | |
|-----------------------------------|-------------|
| A. Change < to > | ✓ Correct |
| B. Change 120 to a smaller number | ✗ Incorrect |
| C. Add more icons to show | ✗ Incorrect |
| D. Remove the else statement | ✗ Incorrect |

Explanation:

Think about what symbol means "greater than" instead of "less than."

2. Why does the nightlight code use a forever block instead of running just once?

MULTIPLE CHOICE

Correct Answer:

- | | |
|--------------------------------------------------|-------------|
| A. To make the light brighter | ✗ Incorrect |
| B. To save battery power | ✗ Incorrect |
| C. To keep checking the light level continuously | ✓ Correct |
| D. To display multiple icons at once | ✗ Incorrect |

Explanation:

A nightlight needs to respond to changes in room lighting all the time.

3. What will happen if you change the number 120 to 50 in the nightlight code?

MULTIPLE CHOICE

Correct Answer:

- A. The nightlight will turn on in brighter rooms ✓ Correct
- B. The nightlight will turn on in darker rooms only ✗ Incorrect
- C. The nightlight will be dimmer ✗ Incorrect
- D. The nightlight will flash on and off ✗ Incorrect

Explanation:

A lower number means the condition will be true more often.

4. A programmer wants to create a "bright light alarm" that shows a warning when the light level goes above 200. Which condition should they use?

MULTIPLE CHOICE

Correct Answer:

- A. `if input.light_level() < 200:` ✗ Incorrect
- B. `if input.light_level() > 200:` ✓ Correct
- C. `if input.light_level() = 200:` ✗ Incorrect
- D. `if input.light_level() + 200:` ✗ Incorrect

Explanation:

An alarm for bright light needs to trigger when light levels are high.

5. What is the main reason for creating a variable called `light_level` in the code?

MULTIPLE CHOICE

Correct Answer:

- A. To make the code run faster ✗ Incorrect
- B. To store and use the current light reading ✓ Correct
- C. To change the brightness of the LED display ✗ Incorrect

D. To count how many times the light changes

✗ Incorrect

Explanation:

Variables hold information that can be used later in the program.

6. Based on the passage, what dual purpose do the LEDs serve on the micro:bit?

MULTIPLE CHOICE

Correct Answer:

A. They show information and sense temperature

✗ Incorrect

B. They show information and sense light

✓ Correct

C. They play music and show pictures

✗ Incorrect

D. They sense light and play sounds

✗ Incorrect

Explanation:

The passage mentions the LEDs can both display things and act as sensors.

7. A student notices their nightlight program turns on and off rapidly in a room with changing light. What is most likely causing this problem?

MULTIPLE CHOICE

Correct Answer:

A. The forever block is running too fast

✗ Incorrect

B. The light level is very close to the threshold value of 120

✓ Correct

C. The diamond icon is too bright

✗ Incorrect

D. The variable is not working correctly

✗ Incorrect

Explanation:

When light levels hover around the cutoff point, small changes cause rapid switching.

8. What would happen if you removed the else part of the nightlight code?

MULTIPLE CHOICE

Correct Answer:

- A. The program would not work at all ✗ Incorrect
- B. The icon would stay on screen even in bright light ✓ Correct
- C. The icon would be brighter ✗ Incorrect
- D. The program would run faster ✗ Incorrect

Explanation:

Without the else statement, nothing tells the screen to clear when it's bright.

9. Looking at the code structure, what is the purpose of the input.light_level() function?

MULTIPLE CHOICE

Correct Answer:

- A. To turn the LEDs on and off ✗ Incorrect
- B. To get the current light measurement from the sensor ✓ Correct
- C. To set how bright the display should be ✗ Incorrect
- D. To count how many LEDs are working ✗ Incorrect

Explanation:

Input functions gather information from sensors and devices.

10. Debug the following code:

DEBUG CODE

Incorrect Code:

```
1 def on_forever():  
2     light_level == input.light_level()  
3  
4 basic.forever(on_forever)
```

Correct Solution:

```
1 def on_forever():  
2     light_level = input.light_level()  
3  
4 basic.forever(on_forever)
```

Explanation:

This code only needs one equals sign.