

# Komunikaty w EdScratch

Niniejszy poradnik zawiera opis wszystkich komunikatów pokazywanych w oknie błędów (z ang. Bug box) aplikacji EdScratch, dostępnej na stronie [www.edscratchapp.com](http://www.edscratchapp.com). Każdy z komunikatów posiada krótkie wytłumaczenie oraz przykłady obrazujące możliwe powody wystąpienia danego komunikatu.

Komunikaty są dzielone na dwa rodzaje: czerwone oraz żółte. Kiedy widzisz czerwony komunikat, program nie może zostać pobrany do pamięci robota Edison. Gdy widzisz żółty komunikat, program może zostać pobrany do pamięci robota Edison i może działać, ale również może nie działać zgodnie z oczekiwaniem.

**Komunikat czerwony:** Czerwone komunikaty są trochę jak znak 'STOP'. Można stwierdzić, że komunikaty te mówią "Przykro mi, ale ten program nie będzie miał sensu dla Edisona." Jeśli w oknie błędów pojawi się jakikolwiek czerwony komunikat, nie będziesz w stanie pobrać programu do Edisona, dopóki nie zostanie rozwiązany problem powodujący występowanie czerwonego komunikatu.

- Zobacz wszystkie [czerwone komunikaty](#)

**Komunikat żółty:** Żółte komunikaty to wiadomości ostrzeżenia. Można stwierdzić, że komunikaty te mówią "Uwaga! To może nie działać w taki sposób, w jaki chcesz żeby to działało." Nawet jeśli widzisz żółty komunikat, możesz pobrać program do pamięci Edisona. Jeśli jednak Twój program nie działa w taki sposób w jaki miał działać, dobrze jest przejrzeć i przemyśleć wszystkie żółte komunikaty. To często pomaga w odkryciu błędów logiki w programie, prowadzących do działania programu w sposób niezamierzony.

- Zobacz wszystkie [żółte komunikaty](#)

## Czerwone komunikaty

Ta sekcja zawiera listę wszystkich czerwonych komunikatów pojawiających się w EdScratch.

Czerwone komunikaty są trochę jak znak 'STOP'. Można stwierdzić, że komunikaty te mówią "Przykro mi, ale ten program nie będzie miał sensu dla Edisona." Jeśli w oknie błędów pojawi się jakikolwiek czerwony komunikat, nie będziesz w stanie pobrać programu do Edisona, dopóki nie zostanie rozwiązany problem powodujący występowanie czerwonego komunikatu.

### 'Backwards until' blocks need a condition input.

#### Znaczenie:

Dosłownie: Bloczek 'Cofaj dopóki' wymaga danych wejściowych w postaci warunku.

Bloczek z kategorii napędowej (kat. Drive) 'backwards until' – czyli "Cofaj dopóki..." – wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie (np. operator  $(\_) < (\_)$ ), umieszczonego wewnątrz wgłębienia w bloku 'backwards until'. Taka informacja będzie jednocześnie warunkiem działania bloczka jazdy z kategorii napędowej.

#### Przykłady:

Bloczek 'backwards until' jest dołączony do bloku 'start', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie), umieszczonego wewnątrz wgłębienia.

Bloczek 'backwards until' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie) umieszczonego wewnątrz wgłębienia.

### 'Bit shift left' blocks need a variable.

#### Znaczenie:

Dosłownie: Bloczek 'Przesuń bit w lewo' potrzebuje zmiennej.

Bloczek z kategorii danych (kat. Data) 'bit shift left' wymaga umieszczenia zmiennej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie. Bloczek potrzebuje informacji, z której dokładnie zmiennej ma pobrać informację a następnie przesunąć jej bit w lewą stronę.

### Przykłady:

Blok 'bit shift left' jest dołączony do bloku 'start', jednak nie zawiera w sobie zmiennej, umieszczonej wewnątrz wgłębienia.

Zmienna została umieszczona w okrągłym miejscu mówiącym o ile miejsc ma zostać przesunięty bit, ale nie umieszczono zmiennej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie wewnątrz bloku.

### 'Bit shift right' blocks need a variable.

#### Znaczenie:

Dosłownie: Bloczek 'Przesuń bit w prawo' potrzebuje zmiennej.

Bloczek z kategorii danych (kat. Data) 'bit shift right' wymaga umieszczenia zmiennej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie. Bloczek potrzebuje informacji, z której dokładnie zmiennej ma pobrać informację a następnie przesunąć jej bit w prawą stronę.

### Przykłady:

Blok 'bit shift right' jest dołączony do bloku 'start', jednak nie zawiera w sobie zmiennej, umieszczonej wewnątrz wgłębienia.

Zmienna została umieszczona w okrągłym miejscu mówiącym o ile miejsc ma zostać przesunięty bit, ale nie umieszczono zmiennej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie wewnątrz bloku.

### 'Decrement variable' blocks need a variable.

#### Znaczenie:

Dosłownie: Bloczek 'Dekrementuj zmienną' potrzebuje zmiennej.

Bloczek z kategorii danych (kat. Data) 'decrement variable' wymaga umieszczenia zmiennej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie. Bloczek potrzebuje informacji, z której dokładnie zmiennej ma pobrać informację a następnie zdekrementować jej wartość.

### Przykłady:

Blok 'decrement variable' jest dołączony do bloku 'start', jednak nie zawiera w sobie zmiennej, umieszczonej wewnątrz wgłębienia.

Bloczek 'decrement variable' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie zmiennej umieszczonej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie.

### Edison cannot detect claps while driving because the motors are too noisy.

#### Znaczenie:

Dosłownie: Edison nie może wykrywać klaśnięcia w trakcie jazdy ponieważ silniki są zbyt głośne.

Kiedy silniki Edisona są włączone, powodują hałas. Czujnik dźwięku robota nie jest w stanie rozróżnić hałasu generowanego przez silniki od innych dźwięków, takich jak klaśnięcie w dłonie. Z tego powodu, robot nie będzie w stanie jechać i jednocześnie wykrywać klaśnięcia.

#### Przykłady:

Blok jazdy 'forwards until' w programie posiada bloczek 'clap detected' umieszczony wewnątrz wgłębienia o diamentowym kształcie w bloku 'forwards until', jako warunek działania jazdy.

Blok jazdy 'backwards until' w programie posiada bloczek 'clap detected' umieszczony wewnątrz wgłębienia o diamentowym kształcie w bloku 'backwards until', jako warunek działania jazdy.

### 'Forwards until' blocks need a condition input.

#### Znaczenie:

Dosłownie: Bloczek 'Jedź naprzód dopóki' wymaga danych wejściowych w postaci warunku.

Bloczek z kategorii napędowej (kat. Drive) 'forwards until' – czyli "Jedź naprzód dopóki..." – wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie (np. operator  $(\_)<(\_)$ ), umieszczonego wewnątrz wgłębienia w bloku 'forwards until'. Taka informacja będzie jednocześnie warunkiem działania bloczka jazdy z kategorii napędowej.

#### Przykłady:

Bloczek 'forwards until' jest dołączony do bloku 'start', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie), umieszczonego wewnątrz wgłębienia.

Bloczek 'forwards until' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie) umieszczonego wewnątrz wgłębienia.

### 'If' blocks need a condition input.

#### Znaczenie:

Dosłownie: Bloczek 'Jeżeli' wymaga danych wejściowych w postaci warunku.

Bloczek z kategorii kontrolowania przepływu programu (kat. Control) 'if' – czyli "jeżeli..." – wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie (np. operator  $(\_) < (\_)$ ), umieszczonego wewnątrz wgłębienia w bloku 'if'. Taka informacja będzie jednocześnie warunkiem działania programu zawartego w klamrze bloku kontrolowania przepływu programu.

#### Przykłady:

Bloczek 'if' jest dołączony do bloku 'start', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie), umieszczonego wewnątrz wgłębienia.

Bloczek 'if' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie) umieszczonego wewnątrz wgłębienia.

### 'If-else' blocks need a condition input.

#### Znaczenie:

Dosłownie: Bloczek 'Jeżeli-W przeciwnym wypadku' wymaga danych wejściowych w postaci warunku.

Bloczek z kategorii kontrolowania przepływu programu (kat. Control) 'if-else' – czyli "jeżeli... w przeciwnym wypadku..." – wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie (np. operator  $(\_) < (\_)$ ), umieszczonego wewnątrz wgłębienia w bloku 'if'. Taka informacja będzie jednocześnie warunkiem działania programu zawartego w klamrze bloku kontrolowania przepływu programu.

#### Przykłady:

Bloczek 'if-else' jest dołączony do bloku 'start', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie), umieszczonego wewnątrz wgłębienia.

Bloczek 'if-else' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie) umieszczonego wewnątrz wgłębienia.

### 'Increment variable' blocks need a variable.

#### Znaczenie:

Dosłownie: Bloczek 'Inkrementuj zmienną' potrzebuje zmiennej.

Bloczek z kategorii danych (kat. Data) 'increment variable' wymaga umieszczenia zmiennej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie. Bloczek potrzebuje informacji, z której dokładnie zmiennej ma pobrać informację a następnie zinkrementować jej wartość (podnieść wartość o 1).

#### Przykłady:

Blok 'increment variable' jest dołączony do bloku 'start', jednak nie zawiera w sobie zmiennej, umieszczonej wewnątrz wgłębienia.

Bloczek 'increment variable' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie zmiennej umieszczonej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie.

### 'Left until' blocks need a condition input.

#### Znaczenie:

Dosłownie: Bloczek 'W lewo dopóki' wymaga danych wejściowych w postaci warunku.

Bloczek z kategorii napędowej (kat. Drive) 'left until' – czyli "... w lewo, dopóki ..." – wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie (np. operator  $(\_)$   $<$   $(\_)$ ), umieszczonego wewnątrz wgłębienia w bloku 'left until'. Taka informacja będzie jednocześnie warunkiem działania bloczka jazdy z kategorii napędowej.

#### Przykłady:

Bloczek 'left until' jest dołączony do bloku 'start', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie), umieszczonego wewnątrz wgłębienia.

Bloczek 'left until' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie) umieszczonego wewnątrz wgłębienia.

**'Line detection' blocks do not work unless the line detection LED is turned on using the 'line tracking LED' block from the 'Sensing' category.**

**Znaczenie:**

Dosłownie: Bloczki 'Wykrywanie linii' nie działają dopóki dioda LED czujnika linii nie jest załączona przy użyciu bloczka 'LED czujnika linii' z kategorii bloczków czujników.

Bloczki 'line tracker on' o diamentowym kształcie z kategorii bloczków czujników (kat. Sensing) mogą być używane jako warunek dla innych bloków, takich jak 'until' lub 'if'. Aby Edison mógł w ten sposób skorzystać z czujnika linii, dioda LED czujnika musi zostać uprzednio załączona. Zamieść blok 'turn line tracking LED' z kategorii Sensing w funkcji włączającej (Nastawiony na 'on') odpowiednio przed bloczkiem warunku 'line tracker on' w swoim programie.

**Przykłady:**

Diamentowy bloczek warunku 'line tracker on' jest umieszczony w bloczku 'if', ale bloczek 'turn line tracking LED' nie jest zamieszczony w programie.

Diamentowy bloczek warunku 'line tracker on' jest umieszczony w bloczku 'if'. W programie zamieszczony jest również bloczek 'turn line tracking LED', jednak nie znajduje się w kolejności "przed" blokiem 'if'.

**'Line detection' events do not work unless the line detection LED is turned on using the 'line tracking LED' block from the 'Sensing' category in the main program.**

**Znaczenie:**

Dosłownie: Bloczki 'Wykrywanie linii' nie działają dopóki dioda LED czujnika linii nie jest załączona przy użyciu bloczka 'LED czujnika linii' z kategorii bloczków czujników, wewnątrz głównego programu.

Wszystkie bloczki z kategorii zdarzeń startowych (kat. Events) związane z czujnikiem linii wymagają włączonej diody LED czujnika linii Edisona. Jeżeli chcesz korzystać z któregoś z tych bloczków zdarzeń startowych, zamieść wcześniej bloczek 'turn line tracking LED' z kategorii Sensing w funkcji włączającej

(Nastawiony na 'on') w odpowiedniej kolejności wewnątrz programu głównego (tj. zbioru blozków dołączonego do żółtego bloku 'Start').

#### Przykłady:

Bloczek zdarzenia startowego 'line tracker on reflective surface' jest umieszczony wewnątrz pulpitu programistycznego, ale blok 'turn line tracking LED' nie jest umieszczony w pętli głównej.

Bloczek zdarzenia startowego 'line tracker on reflective surface' jest umieszczony wewnątrz pulpitu programistycznego, a blok 'turn line tracking LED' jest dołączony do bloczku zdarzenia startowego 'line tracker on reflective surface', jednak nie ma go w pętli głównej.

**'Obstacle detection' blocks do not work unless the obstacle detection beam is turned on using the 'obstacle detection beam' block from the 'Sensing' category.**

#### Znaczenie:

Dosłownie: Bloczki 'Wykrywanie przeszkód' nie działają dopóki promień wykrywający przeszkody nie są załączone przy użyciu bloczka 'Promień wykrywania przeszkód' z kategorii bloków czujników.

Bloczki 'obstacle detected' o diamentowym kształcie z kategorii bloków czujników (kat. Sensing) mogą być używane jako warunek dla innych bloków, takich jak 'until' lub 'if'. Aby Edison mógł w ten sposób skorzystać z czujnika przeszkód, promień wykrywający przeszkody musi zostać uprzednio załączony. Zamieść blok 'turn obstacle detection beam' z kategorii Sensing w funkcji włączającej (Nastawiony na 'on') odpowiednio przed bloczkiem warunku 'obstacle detected' w swoim programie.

#### Przykłady:

Diamentowy bloczek warunku 'obstacle detected' jest umieszczony w bloczku 'if', ale bloczek 'turn obstacle detection beam' nie jest zamieszczony w programie.

Diamentowy bloczek warunku 'obstacle detected' jest umieszczony w bloczku 'if'. W programie zamieszczony jest również bloczek 'turn obstacle detection beam', jednak nie znajduje się w kolejności "przed" blokiem 'if'.



'Obstacle detection' events do not work unless the obstacle detection beam is turned on using the 'obstacle detection beam' block from the 'Sensing' category in the main program.

**Znaczenie:**

Dosłownie: Bloczki 'Wykrywanie przeszkód' nie działają dopóki promienie wykrywające przeszkody nie są załączone przy użyciu bloczka 'Promień wykrywania przeszkód' z kategorii bloczków czujników, wewnątrz głównego programu.

Wszystkie bloczki z kategorii zdarzeń startowych (kat. Events) związane z czujnikiem przeszkód wymagają włączenia promienia czujnika przeszkód Edisona. Jeżeli chcesz korzystać z któregoś z tych bloczków zdarzeń startowych, zamieść wcześniej bloczek 'turn obstacle detection beam' z kategorii Sensing w funkcji włączającej (Nastawiony na 'on') w odpowiedniej kolejności wewnątrz programu głównego (tj. zbioru bloczków dołączonego do żółtego bloku 'Start').

**Przykłady:**

Bloczek zdarzenia startowego 'any obstacle detected' jest umieszczony wewnątrz pulpitu programistycznego, ale blok 'turn obstacle detection beam' nie jest umieszczony w pętli głównej.

Bloczek zdarzenia startowego 'any obstacle detected' jest umieszczony wewnątrz pulpitu programistycznego, a blok 'turn obstacle detection beam' jest dołączony do bloczku zdarzenia startowego 'any obstacle detected', jednak nie ma go w pętli głównej.

'Play music in background' grouping blocks can only accept 'note' blocks.

**Znaczenie:**

Dosłownie: Blok grupowy 'Graj muzykę w tle' akceptuje wyłącznie bloczki 'dźwięk'.

Bloczek z kategorii dźwiękowej (kat. Sound) 'play music in background' pozwala na granie różnych dźwięków podczas wykonywania innych instrukcji w programie, takich jak np. jazda robotą. Jedyne bloczki które mogą zostać umieszczone w bloku 'play music in background' to bloki muzyczne.

**Przykłady:**

Bloczek 'beep' znajduje się wewnątrz bloczka 'play music in background'.

Bloczek 'set music tempo' znajduje się wewnątrz bloczka 'play music in background'.

'Play music in background' grouping blocks need to contain at least one 'note' block.

**Znaczenie:**

Dosłownie: Blok grupowy 'Graj muzykę w tle' musi posiadać przynajmniej jeden bloczek 'dźwięk'.

Bloczek z kategorii dźwiękowej (kat. Sound) 'play music in background' pozwala na granie różnych dźwięków podczas wykonywania innych instrukcji w programie, takich jak np. jazda robotą. Blok 'play music in background' musi mieć przynajmniej jeden bloczek z dźwiękiem zawarty wewnątrz klamry aby poprawnie działać.

**Przykłady:**

Blok 'play music in background' jest dołączony do bloku 'start', jednak nie zawiera w sobie żadnych bloczków muzycznych.

Bloczek 'play music in background' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie żadnych bloczków muzycznych.

Programs must have a 'start' event to work correctly. If you are seeing this error message, start a new program by going to the menu and selecting 'new'.

**Znaczenie:**

Dosłownie: Program musi posiadać wydarzenie startowe 'start' aby poprawnie działać. Jeśli widzisz ten komunikat błędu, stwórz nowy program wybierając w menu opcję 'nowy'.

Żółty blok 'start' jest wymagany w każdym programie EdScratch, dlatego blok 'start' powinien być zawsze obecny na pulpicie programistycznym. W bardzo rzadkich przypadkach, aplikacja EdScratch może się nie załadować poprawnie a bloczek 'start' może się nie pojawić. Aplikacja powinna zostać uruchomiona ponownie.

**Przykłady:**

Wystąpił krytyczny błąd sieci podczas ładowania aplikacji EdScratch, uniemożliwiając prawidłowe uruchomienie aplikacji.

Wystąpił krytyczny błąd urządzenia podczas ładowania aplikacji EdScratch, uniemożliwiając prawidłowe uruchomienie aplikacji.

## 'Repeat until' blocks need a condition input.

### Znaczenie:

Dosłownie: Bloczki 'Powtarzaj dopóki' wymagają danych wejściowych w postaci warunku.

Bloczek z kategorii kontrolowania przepływu programu 'repeat until' – czyli "Powtarzaj dopóki..." – wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie (np. operator  $(\_) < (\_)$ ), umieszczonego wewnątrz wgłębienia w bloku 'repeat until'.

### Przykłady:

Bloczek 'repeat until' jest dołączony do bloku 'start', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie), umieszczonego wewnątrz wgłębienia.

Bloczek 'repeat until' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie) umieszczonego wewnątrz wgłębienia.

## 'Right until' blocks need a condition input.

### Znaczenie:

Dosłownie: Bloczki 'W prawo dopóki' wymagają danych wejściowych w postaci warunku.

Bloczek z kategorii napędowej (kat. Drive) 'right until' – czyli "... w prawo, dopóki ..." – wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie (np. operator  $(\_) < (\_)$ ), umieszczonego wewnątrz wgłębienia w bloku 'right until'. Taka informacja będzie jednocześnie warunkiem działania bloczka jazdy z kategorii napędowej.

### Przykłady:

Bloczek 'right until' jest dołączony do bloku 'start', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie), umieszczonego wewnątrz wgłębienia.

Bloczek 'right until' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie) umieszczonego wewnątrz wgłębienia.

## 'Set variable' blocks need a variable.

### Znaczenie:

Dosłownie: Bloczki 'Nastaw zmienną' wymagają zmiennej.

Bloczek z kategorii danych (kat. Data) 'set variable' wymaga umieszczenia zmiennej wewnątrz zaciemnionego wgłębienia o okrągłym kształcie. Bloczek potrzebuje informacji, której dokładnie zmiennej ma ustawić wartość określoną przez użytkownika w drugim wejściu.

### Przykłady:

Blok 'set variable' jest dołączony do bloku 'start', jednak nie zawiera w sobie zmiennej, umieszczonej wewnątrz wgłębienia.

Zmienna została dodana do okrągłego wejścia mówiącego o wartości na którą ma być ustawiona dana zmienna, ale nie umieszczono bloczku zmiennej w zaciemnionym wgłębieniu o okrągłym kształcie w bloku 'set variable'.

## There are no blocks connected to the 'start' event, so there is no main program for Edison to run.

### Znaczenie:

Dosłownie: Nie ma bloczków podłączonych do bloku zdarzenia 'start', więc nie ma głównego programu dla Edisona do uruchomienia.

Aby Edison mógł uruchomić jakikolwiek program utworzony w EdScratch, program ten musi zawierać tzw. program główny. Każdy bloczek dołączony do żółtego bloku 'start' w EdScratch, tworzy wspólnie z tym blokiem program główny który zostanie uruchomiony przez Edisona kiedy program będzie skompilowany i przesłany do niego. Jeśli nie ma żadnych bloczków dołączonych do bloku 'start' – wtedy nie ma programu głównego.

### Przykłady:

Program został utworzony i skompilowany (czyli naciśnięto przycisk 'program Edison'), następnie wszystkie bloczki dołączone do bloku 'start' zostały usunięte, pozostawiając wyłącznie blok 'start' na pulpicie programistycznym.

Bloczek zdarzenia startowego 'clap detected' z podprogramem został utworzony, ale do bloku 'start' nie dołączono żadnego innego bloku z instrukcją, a następnie naciśnięto przycisk 'program Edison'.

## 'Wait milliseconds' blocks need an input value.

### Znaczenie:

Dosłownie: Bloczki 'Czekaj milisekundy' potrzebuje

Bloczek z kategorii kontrolowania przepływu programu (kat. Control) 'wait milliseconds' potrzebuje informacji wejściowej wpisanej wewnątrz owalnego wgłębienia. Wejściem jest wartość, informująca jak długo blok ma "czekać".

### Przykłady:

Bloczek 'wait milliseconds' jest dołączony do bloku 'start', jednak nie zawiera w sobie wartości (czyli wpisanej liczby lub bloczka o okrągłym kształcie), umieszczonej wewnątrz wgłębienia.

Bloczek 'wait milliseconds' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie wartości (czyli wpisanej liczby lub bloczka o okrągłym kształcie), umieszczonej wewnątrz wgłębienia.

## 'Wait until' blocks need a condition input.

### Znaczenie:

Dosłownie: Bloczki 'Czekaj dopóki' potrzebuje

Bloczek z kategorii kontrolowania przepływu programu (kat. Control) 'wait until' wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie (np. operator  $(\_) < (\_)$ ), umieszczonego wewnątrz wgłębienia w bloku 'wait until'. Taka informacja będzie jednocześnie warunkiem działania programu zawartego w klamrze bloku kontrolowania przepływu programu.

### Przykłady:

Bloczek 'wait until' jest dołączony do bloku 'start', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie), umieszczonego wewnątrz wgłębienia.

Bloczek 'wait until' jest dołączony do bloku zdarzenia startowego (kat. Events) 'any obstacle detected', jednak nie zawiera w sobie warunku działania (czyli bloczka o diamentowym kształcie) umieszczonego wewnątrz wgłębienia.

## Yellow messages

Ta sekcja zawiera listę wszystkich żółtych komunikatów pojawiających się w EdScratch.

Żółte komunikaty to wiadomości ostrzeżenia. Można stwierdzić, że komunikaty te mówią "Uwaga! To może nie działać w taki sposób, w jaki chcesz żeby to działało." Nawet jeśli widzisz żółty komunikat, możesz pobrać program do pamięci Edisona. Jeśli jednak Twój program nie działa w taki sposób w jaki miał działać, dobrze jest przejrzeć i przemyśleć wszystkie żółte komunikaty. To często pomaga w odkryciu błędów logiki w programie, prowadzących do działania programu w sposób niezamierzony.

### A 'forever' loop will continue looping until you press the square button on Edison.

#### Znaczenie:

Dosłownie: Nieskończona pętla będzie zapętlać dopóki nie naciśniesz kwadratowego przycisku Edisona

Bloczek pętli z kategorii kontrolowania przepływu programu (kat. Control) 'forever' jest pętlą nieskończoną, bez wbudowanego punktu końca pętli. Uruchomiona w robocie Edison, pętla będzie działać cały czas, dopóki nie zostanie wciśnięty przycisk 'stop' (kwadratowy) lub dopóki nie zostaną wyczerpane baterie Edisona. Program nie przejdzie do żadnej instrukcji następującej w kolejności po pętli nieskończonej.

#### Przykłady:

Bloczek pętli nieskończonej 'forever' znajduje się wewnątrz bloczku pętli 'repeat' w programie.

Bloczek pętli nieskończonej 'forever' znajduje się wewnątrz bloczku 'if' w programie.

An 'operator' block inside a 'drive backwards until' block may cause Edison to drive forever or skip the drive block.

**Znaczenie:**

Dosłownie: Bloczek 'operatora' wewnątrz bloczka 'jedź do tyłu dopóki' może spowodować, że Edison będzie jeździć bez końca lub pominie bloczek jazdy.

Bloczek z kategorii napędowej (kat. Drive) 'backwards until' wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie aby ustalić warunek działania bloczka 'drive'. Bloczki z kategorii 'Operators' (bloczki o diamentowym kształcie) są wyrażeniami logicznymi, których rezultat będzie albo Prawdą albo Fałszem. W zależności od wejść w bloku operatora, wyrażenie może okazać się zawsze Prawdą lub zawsze Fałszem (na przykład, kiedy wejścia operatora nie zmieniają się w trakcie ruchu silników). Ten stały stan może spowodować, że bloczek 'drive' (bloczek jazdy) będzie zawsze spełniony lub zawsze pomijany.

**Przykłady:**

Bloczek operatora (kat. operators) 'Większy od' jest umieszczony wewnątrz wejścia bloczku 'backwards until' jako warunek działania programu.

Bloczek operatora (kat. operators) 'Nie jest równy' jest umieszczony wewnątrz wejścia bloczku 'backwards until' jako warunek działania programu.

An 'operator' block inside a 'drive forwards until' block may cause Edison to drive forever or skip the drive block.

**Znaczenie:**

Dosłownie: Bloczek 'operatora' wewnątrz bloczka 'jedź do przodu dopóki' może spowodować, że Edison będzie jeździć bez końca lub pominie bloczek jazdy.

Bloczek z kategorii napędowej (kat. Drive) 'forwards until' wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie aby ustalić warunek działania bloczka 'drive'. Bloczki z kategorii 'Operators' (bloczki o diamentowym kształcie) są wyrażeniami logicznymi, których rezultat będzie albo Prawdą albo Fałszem. W zależności od wejść w bloku operatora, wyrażenie może okazać się zawsze Prawdą lub zawsze Fałszem (na przykład, kiedy wejścia operatora nie zmieniają się w trakcie ruchu silników). Ten stały stan może spowodować, że bloczek 'drive' (bloczek jazdy) będzie zawsze spełniony lub zawsze pomijany.

### Przykłady:

Bloczek operatora (kat. operators) 'Większy od' jest umieszczony wewnątrz wejścia bloczku 'forwards until' jako warunek działania programu.

Bloczek operatora (kat. operators) 'Nie jest równy' jest umieszczony wewnątrz wejścia bloczku 'forwards until' jako warunek działania programu.

### An 'operator' block inside a 'drive left until' block may cause Edison to drive forever or skip the drive block.

#### Znaczenie:

Dosłownie: Bloczek 'operatora' wewnątrz bloczka 'jedź w lewo dopóki' może spowodować, że Edison będzie jeździć bez końca lub pominie bloczek jazdy.

Bloczek z kategorii napędowej (kat. Drive) 'drive left until' wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie aby ustalić warunek działania bloczka 'drive'. Bloczki z kategorii 'Operators' (bloczki o diamentowym kształcie) są wyrażeniami logicznymi, których rezultat będzie albo Prawdą albo Fałszem. W zależności od wejść w bloku operatora, wyrażenie może okazać się zawsze Prawdą lub zawsze Fałszem (na przykład, kiedy wejścia operatora nie zmieniają się w trakcie ruchu silników). Ten stały stan może spowodować, że bloczek 'drive' (bloczek jazdy) będzie zawsze spełniony lub zawsze pomijany.

### Przykłady:

Bloczek operatora (kat. operators) 'Większy od' jest umieszczony wewnątrz wejścia bloczku 'drive left until' jako warunek działania programu.

Bloczek operatora (kat. operators) 'Nie jest równy' jest umieszczony wewnątrz wejścia bloczku 'drive left until' jako warunek działania programu.

### An 'operator' block inside a 'drive right until' block may cause Edison to drive forever or skip the drive block.

#### Znaczenie:

Dosłownie: Bloczek 'operatora' wewnątrz bloczka 'jedź w prawo dopóki' może spowodować, że Edison będzie jeździć bez końca lub pominie bloczek jazdy.

Bloczek z kategorii napędowej (kat. Drive) 'drive right until' wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie aby ustalić



warunek działania bloczka 'drive'. Bloczki z kategorii 'Operators' (bloczki o diamentowym kształcie) są wyrażeniami logicznymi, których rezultat będzie albo Prawdą albo Fałszem. W zależności od wejść w bloku operatora, wyrażenie może okazać się zawsze Prawdą lub zawsze Fałszem (na przykład, kiedy wejścia operatora nie zmieniają się w trakcie ruchu silników). Ten stały stan może spowodować, że bloczek 'drive' (bloczek jazdy) będzie zawsze spełniony lub zawsze pomijany.

#### Przykłady:

Bloczek operatora (kat. operators) 'Większy od' jest umieszczony wewnątrz wejścia bloczku 'drive left until' jako warunek działania programu.

Bloczek operatora (kat. operators) 'Nie jest równy' jest umieszczony wewnątrz wejścia bloczku 'drive left until' jako warunek działania programu.

### An 'operator' block inside a 'repeat until' block may cause Edison to loop forever or skip the 'repeat' block.

#### Znaczenie:

Dośłownie: Bloczek 'operatora' wewnątrz bloczka 'powtarzaj dopóki' może spowodować, że Edison będzie ciągle zapętłony lub ominie blok 'powtarzaj'.

Bloczek z kategorii kontrolowania przepływu programu (kat. Control) 'repeat until' wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie aby ustalić warunek działania bloczka 'repeat until'. Bloczki z kategorii 'Operators' (bloczki o diamentowym kształcie) są wyrażeniami logicznymi, których rezultat będzie albo Prawdą albo Fałszem. W zależności od wejść w bloku operatora, wyrażenie może okazać się zawsze Prawdą lub zawsze Fałszem (na przykład, kiedy wejścia operatora nie zmieniają się w trakcie działania zapętłonego programu). Ten stały stan może spowodować, że bloczek 'repeat' (bloczek powtarzania) będzie zawsze spełniony, tworząc w ten sposób pętlę nieskończoną, lub zawsze pomijany, powodując że bloczki zawarte w pętli nigdy nie zostaną uruchomione.

#### Przykłady:

Bloczek operatora (kat. operators) 'Większy od' jest umieszczony wewnątrz wejścia bloczku 'repeat until' jako warunek działania programu.

Bloczek operatora (kat. operators) 'Nie jest równy' jest umieszczony wewnątrz wejścia bloczku 'repeat until' jako warunek działania programu.

## An 'operator' block inside a 'repeat' block may cause Edison to skip the 'repeat' block.

### Znaczenie:

Dosłownie: Bloczek 'operatora' wewnątrz bloczka 'powtórz' może spowodować, że Edison pominie blok 'powtórz'.

Bloczek z kategorii kontrolowania przepływu programu (kat. Control) 'repeat' wymaga informacji wejściowej w postaci wartości lub bloczka o okrągłym kształcie (np. z kategorii 'Operators'). Wszystkie bloczki z kategorii operatorów o okrągłym kształcie dają informacje w postaci wartości numerycznej. W zależności od wejść bloka operatora wartość może być równa zero, co w takim przypadku spowoduje, że blok powtarzania i wszystkie instrukcje w nim zawarte zostaną pominięte.

### Przykłady:

Bloczek operatora dzielenia jest zawarty wewnątrz bloczka 'powtarzaj' w programie.

Bloczek operatora odejmowania jest zawarty wewnątrz bloczka 'powtarzaj' w programie.

## An 'operator' block inside a 'wait until' block may cause Edison to wait forever or skip the 'wait' block.

### Znaczenie:

Dosłownie: Bloczek 'operatora' wewnątrz bloczka 'czekaj dopóki' może spowodować, że Edison będzie ciągle czekać lub ominie blok 'czekaj'.

Bloczek z kategorii kontrolowania przepływu programu (kat. Control) 'wait until' wymaga informacji wejściowej w postaci wybranego bloczka o diamentowym kształcie aby ustalić warunek działania bloczka 'wait until'. Bloczki z kategorii 'Operators' (bloczki o diamentowym kształcie) są wyrażeniami logicznymi, których rezultat będzie albo Prawdą albo Fałszem. W zależności od wejść w bloku operatora, wyrażenie może okazać się zawsze Prawdą lub zawsze Fałszem (na przykład, kiedy wejścia operatora nie zmieniają się w trakcie działania bloczka czekania). Ten stały stan może spowodować, że bloczek 'wait' (bloczek czekania) będzie zawsze spełniony, powodując że program będzie ciągle czekać, lub zawsze pomijany, powodując że bloczek czekania będzie zupełnie pominięty.

### Przykłady:

Bloczek operatora (kat. operators) 'Większy od' jest umieszczony wewnątrz wejścia bloczku 'wait until' jako warunek działania programu.

Bloczek operatora (kat. operators) 'Nie jest równy' jest umieszczony wewnątrz wejścia bloczku 'wait until' jako warunek działania programu.

## An 'operator' block inside a 'wait' block may cause Edison to skip the 'wait' block.

### Znaczenie:

Dosłownie: Bloczek 'operatora' wewnątrz bloczka 'poczekaj' może spowodować, że Edison pominie blok 'poczekaj'.

Bloczek z kategorii kontrolowania przepływu programu (kat. Control) 'wait' wymaga informacji wejściowej w postaci wartości lub bloczka o okrągłym kształcie (np. z kategorii 'Operators'). Wszystkie bloczki z kategorii operatorów o okrągłym kształcie dają informacje w postaci wartości numerycznej. W zależności od wejść bloka operatora wartość może być równa zero, co w takim przypadku spowoduje, że blok czekania i wszystkie instrukcje w nim zawarte zostaną pominięte.

### Przykłady:

Bloczek operatora dzielenia jest zawarty wewnątrz bloczka 'wait milliseconds' w programie.

Bloczek operatora odejmowania jest zawarty wewnątrz bloczka 'wait milliseconds' w programie.

## Blocks not connected to a yellow 'event' block will not be programmed into Edison.

### Znaczenie:

Dosłownie: Bloczki nie połączone z żółtym blokiem 'zdarzenia startowego' nie zostaną zaprogramowane w Edisonie.

Bloczki z różnych kategorii mogą być dodawane do pulpitu programistycznego EdScratch, ale nie oznacza to, że są połączone z jakimkolwiek innym bloczkiem. Takie bloki, które nie są połączone z żółtym bloczkiem zdarzenia startowego (np. 'Start') będą ignorowane przez kompilator, kiedy naciśniesz przycisk 'program Edison' aby przesłać program do pamięci robota.

### Przykłady:

Bloczek z kategorii dźwiękowej (kat. Sound) 'beep' jest na pulpicie programistycznym EdScratch ale nie jest dołączony do żadnego innego bloczku.

Zestaw połączonych ze sobą bloczków umieszczony na pulpicie programistycznym nie jest połączony z blokiem 'start' lub jakimkolwiek innym żółtym bloczkiem zdarzenia startowego z kategorii Events.

Comments are notes to help keep track of things. Comment blocks will not be programmed into Edison.

**Znaczenie:**

Dosłownie: Komentarze są notatkami pomagającymi zrozumieć rzeczy. Bloczki komentarzy nie będą zaprogramowane w Edisonie.

Różowe bloczki komentarzy dodawane z kategorii 'Comment' do bloków programu w EdScratch, nie są kompilowane i w rezultacie nie będą przesyłane do pamięci robota Edison, kiedy naciśniesz przycisk 'program Edison'. Komentarze są jak notatki na karteczkach samoprzylepnych: krótkie notatki przyklejone, aby pomóc programistom pamiętać o pewnych rzeczach. Te bloczki służą programistom do tego, aby dodawać wyjaśnienia kodu lub istotne informacje dla samego programisty. Komentarze będą zawsze ignorowane przez kompilator i nie wpływają w żaden sposób na działanie zaprogramowanego robota – pomimo tego, że można dołączać bloczki z komentarzem np. do programu głównego.

**Przykłady:**

Bloczek 'comment' jest dołączony do bloku 'start' w programie.

Bloczek 'comment' jest dołączony do bloku zdarzenia startowego 'any obstacle detected' w programie.

Consider turning the line detection LED on using the 'line tracking LED' block from the 'Sensing' category if measuring reflectiveness of the driving surface.

**Znaczenie:**

Dosłownie: Rozważ włączenie diody LED czujnika linii poprzez bloczek 'wykrywanie linii LED' z kategorii 'Czujniki' jeśli mierzysz odbicie światła od powierzchni jazdy.

Czujnik linii robota Edison składa się z wielu części, włączając w to fototranzystor oraz diodę LED. Fototranzystor jest zawsze włączony i umożliwia wykrywanie widzialnego światła, ale jeśli spód robota jest bardzo blisko matowej powierzchni (np. kiedy robot jeździ po biurku), fotorezystor wykryje bardzo małe ilości światła. Aby otrzymać jak najlepszy pomiar światła odbitego od powierzchni po której porusza się Edison, konieczne może być włączenie diody LED czujnika linii, wykorzystując blok 'turn line tracking LED' z kategorii czujników (kat. Sensing).

**Przykłady:**

Blok 'if' z kategorii kontrolowania przepływu programu (kat. Control) zawiera bloczek o diamentowym kształcie z kategorii 'Operators' jako warunek wejściowy. Jednym z wejść tego bloku operatora jest okrągły bloczek 'light level of sensor' z kategorii 'Sensing', ustawiony na 'line tracking'.

Blok 'wait until' z kategorii kontrolowania przepływu programu (kat. Control) zawiera bloczek o diamentowym kształcie z kategorii 'Operators' jako warunek wejściowy. Jednym z wejść tego bloku operatora jest okrągły bloczek 'light level of sensor' z kategorii 'Sensing', ustawiony na 'line tracking'.

**Driving the motors creates noise which may cause the 'clap' event to trigger. This may cause the 'clap event' blocks to trigger repeatedly while Edison is driving.**

**Znaczenie:**

Dosłownie: Działające silniki tworzą hałas który może spowodować, że zadziała zdarzenie startowe 'kłaśnięcie'.

Kiedy silniki robota Edison są włączone, wytwarzają hałas. Czujnik dźwięku Edisona nie jest w stanie odróżnić hałasu wynikającego z pracujących silników od innych dźwięków, takich jak kłaśnięcia. Z tego powodu, jeżeli w programie jest używane zdarzenie startowe 'clap detected' oraz bloczki jazdy, zdarzenie startowe uruchamiające podprogram po wykryciu kłaśnięcia może zadziałać w momencie wykonywania instrukcji dotyczących jazdy robota.

**Przykłady:**

Program główny zawiera bloczki 'drive' z kategorii napędowej, jednocześnie istnieje podprogram załączany po wykryciu kłaśnięcia - 'clap detected' z kategorii bloczków zdarzeń startowych (kat. Event).

Istnieje podprogram załączany przez wykrycie przeszkody – 'any obstacle detected' – z kategorii bloczków zdarzeń startowych (kat. Event), który zawiera w sobie bloczki 'drive' z kategorii napędowej. Jednocześnie istnieje podprogram załączany przez wykrycie kłaśnięcia – 'clap detected' – z kategorii bloczków zdarzeń startowych.

**Edison cannot receive messages while 'obstacle detection' is on. Make sure to turn obstacle detection off in the program before attempting to receive a message.**

**Znaczenie:**

Dosłownie: Edison nie może otrzymać wiadomości gdy 'wykrywanie przeszkód' jest załączone. Upewnij się że wyłączasz wykrywanie przeszkód w programie przed próbą odbioru wiadomości.

Roboty Edison wykorzystują te same elementy czujników do odbierania wiadomości i wykrywania przeszkód. Te elementy nie mogą robić obu tych czynności jednocześnie. Aby otrzymać wiadomość IR od innego robota Edison, wykrywanie przeszkód musi być wyłączone. Można to ustawić wykorzystując bloczek 'turn obstacle detection beam' z kategorii czujników (kat. Sensing), nastawionego na 'off' czyli wyłączenie.

#### Przykłady:

Program główny zawiera bloczek 'turn obstacle detection beam' z kategorii bloczków czujników (kat. Sensing), oraz jednocześnie zawiera bloczek o diamentowym kształcie 'IR message detected' jako warunek wejściowy dla bloczka 'if' z kategorii kontrolowania przepływu programu, umieszczony w dalszej części programu.

Program główny zawiera bloczek 'turn obstacle detection beam' z kategorii bloczków czujników (kat. Sensing). Jednocześnie istnieje podprogram załączany przez wykrycie kłaśnięcia - 'clap detected' - z kategorii bloczków zdarzeń startowych. Podprogram zawiera bloczek o diamentowym kształcie 'IR message detected' jako warunek wejściowy dla bloczka 'if' z kategorii kontrolowania przepływu programu.

**Edison cannot receive messages while 'obstacle detection' is on. Make sure to turn obstacle detection off in the main program before attempting to receive a message.**

#### Znaczenie:

Dosłownie: Edison nie może otrzymać wiadomości gdy 'wykrywanie przeszkód' jest załączone. Upewnij się że wyłączasz wykrywanie przeszkód w głównym programie przed próbą odbioru wiadomości.

Roboty Edison wykorzystują te same elementy czujników do odbierania wiadomości i wykrywania przeszkód. Te elementy nie mogą robić obu tych czynności jednocześnie. Aby otrzymać wiadomość IR od innego robota Edison, wykrywanie przeszkód musi być wyłączone. Można to ustawić wykorzystując bloczek 'turn obstacle detection beam' z kategorii czujników (kat. Sensing), nastawionego na 'off' czyli wyłączenie.

#### Przykłady:

Program główny zawiera bloczek 'turn obstacle detection beam' z kategorii czujników, jednocześnie istnieje podprogram działający po otrzymaniu wiadomości podczerwonej - 'IR message received' z kategorii bloczków zdarzeń startowych (kat. Event).

Program główny zawiera wiele bloczków 'turn obstacle detection beam' z kategorii bloczków czujników (kat. Sensing). Jednocześnie istnieje podprogram załączany przez bloczek 'IR message received' z kategorii bloczków zdarzeń startowych (kat. Event).

**Edison cannot receive remote control codes while 'obstacle detection' is on. Make sure to turn obstacle detection off in the program before attempting to receive a remote code.**

**Znaczenie:**

Dosłownie: Edison nie może otrzymać sygnałów zdalnego sterowania gdy 'wykrywanie przeszkód' jest załączone. Upewnij się że wyłączasz wykrywanie przeszkód w programie przed próbą odbioru sygnałów zdalnego sterowania.

Roboty Edison wykorzystują te same elementy czujników do odbierania sygnałów IR z pilota i wykrywania przeszkód. Te elementy nie mogą robić obu tych czynności jednocześnie. Aby otrzymać wiadomość IR z pilota do zdalnego sterowania, wykrywanie przeszkód musi być wyłączone. Można to ustawić wykorzystując bloczek 'turn obstacle detection beam' z kategorii czujników (kat. Sensing), nastawionego na 'off' czyli wyłączenie.

**Przykłady:**

Program główny zawiera bloczek 'turn obstacle detection beam' z kategorii bloczków czujników (kat. Sensing), oraz jednocześnie zawiera bloczek o diamentowym kształcie 'remote code received' jako warunek wejściowy dla bloczka 'if' z kategorii kontrolowania przepływu programu, umieszczony w dalszej części programu.

Program główny zawiera bloczek 'turn obstacle detection beam' z kategorii bloczków czujników (kat. Sensing). Jednocześnie istnieje podprogram załączany przez wykrycie kłaśnięcia - 'clap detected' - z kategorii bloczków zdarzeń startowych. Podprogram zawiera bloczek o diamentowym kształcie 'remote code received' jako warunek wejściowy dla bloczka 'if' z kategorii kontrolowania przepływu programu.

**Edison cannot receive remote control codes while 'obstacle detection' is on. Make sure to turn obstacle detection off in the main program before attempting to receive a remote code.**

**Znaczenie:**

Dosłownie: Edison nie może otrzymać sygnałów zdalnego sterowania gdy 'wykrywanie przeszkód' jest załączone. Upewnij się że wyłączasz wykrywanie

przeszkód w głównym programie przed próbą odbioru sygnałów zdalnego sterowania.

Roboty Edison wykorzystują te same elementy czujników do odbierania sygnałów IR z pilota i wykrywania przeszkód. Te elementy nie mogą robić obu tych czynności jednocześnie. Aby otrzymać wiadomość IR z pilota do zdalnego sterowania, wykrywanie przeszkód musi być wyłączone. Można to ustawić wykorzystując bloczek 'turn obstacle detection beam' z kategorii czujników (kat. Sensing), nastawionego na 'off' czyli wyłączenie.

#### Przykłady:

Program główny zawiera bloczek 'turn obstacle detection beam' z kategorii czujników, jednocześnie istnieje podprogram działający po otrzymaniu wiadomości podczerwonej - 'remote code received' z kategorii bloczków zdarzeń startowych (kat. Event).

Program główny zawiera wiele bloczków 'turn obstacle detection beam' z kategorii bloczków czujników (kat. Sensing). Jednocześnie istnieje podprogram załączany przez bloczek 'remote code received' z kategorii bloczków zdarzeń startowych (kat. Event).

### Light level will return a very high value which may cause Edison to loop for a long time.

#### Znaczenie:

Dosłownie: Poziom natężenia światła zwraca bardzo dużą wartość co może spowodować zapętlenie Edisona na długi czas.

Fototranzystory Edisona mierzą wykryte światło widzialne i zwracają wartość natężenia tego zmierzonego przez robota światła. Poziomy natężenia światła w większości środowisk, np. w pokoju podczas dnia, będą wysokimi wartościami. Kiedy ta wartość jest używana jako wejście określające powtórzenia pętli 'repeat', może to spowodować, że pętla będzie powtarzać się bardzo wiele razy.

#### Przykłady:

Program główny zawiera pętlę 'repeat' z okrągłym bloczkiem 'light level of' z kategorii czujników (kat. Sensing), jako wartość wejściowa dla pętli.

Program zawiera podprogram uruchamiany przez bloczek z kategorii zdarzeń startowych (kat. Events) 'clap detected'. Wewnątrz podprogramu znajduje się pętla 'repeat' z bloczkiem o okrągłym kształcie 'light level of' z kategorii bloczków czujników (kat. Sensing) jako wartość wejściowa dla tej pętli.



Some sensor data is stored in Edison's memory. You may need to clear the sensor data for your program to work correctly.

**Znaczenie:**

Dośćownie: Pewne dane z czujników są zachowane w pamięci Edisona. Możesz potrzebować wyczyścić te dane aby Twój program działał prawidłowo.

Wszystkie czujniki Edisona wytwarzają dane przy określonych zdarzeniach. Niektóre z tych danych są zapisywane w pamięci Edisona. Te dane mogą czasami stanowić problem, w przypadkach gdy robot reaguje na jakieś stare zdarzenie ponieważ ciągle je pamięta.

Dla przykładu, kiedy Edison sprawdza czy pewien warunek został spełniony, jeśli istnieją w pamięci dane z określonego czujnika, może się okazać, że robot uzna te zapisane dane i rozpatrzy warunek jako spełniony. Nawet pomimo tego, że w rzeczywistości warunek ten nie został spełniony, lub nie miał prawa zostać spełnionym. Z tego właśnie powodu, czyszczenie danych z czujników jest dobrą praktyką programistyczną. W EdScratch można wyczyścić dane z czujników używając bloczka 'clear sensor data' z kategorii bloczków czujników (kat. Sensing).

Czyszczenie danych z czujników jest szczególnie ważne, kiedy wykorzystuje się zdarzenia dotyczące czujników (np. w bloczkach 'if' lub 'until') zagnieżdżonych w pętli. Czyszczenie tych danych zapobiega oddziaływaniu poprzednio wykonanej pętli na następną iterację. Możesz zapobiec takim sytuacjom, dodając bloczek 'clear sensor data' do czujnika który wykorzystujesz, wybranego z rozwijanej listy w bloczku 'clear sensor data', po sprawdzeniu zdarzenia będącego warunkiem w danej iteracji.

Dobrze jest również czyścić dane z czujników zaraz po rozpoczęciu programu, na wypadek gdyby robot zachował stare dane z poprzedniego uruchomionego programu. Dane można usunąć przez wstawienie bloczka 'clear sensor data' z wybranymi czujnikami, na początku programu, po zdarzeniu startowym 'start'.

Możesz dowiedzieć się więcej o bloczku 'clear sensor data' z EdScratch w ćwiczeniu R4-1.4 *Odkrywamy łączenie i zagnieżdżanie if* z Rozdziału 4 w [Lekcje z EdScratch](#).

**Przykłady:**

Program zawiera bloczek 'if' z kategorii kontrolowania przepływu programu (kat. Control) z bloczkiem o diamentowym kształcie 'clap detected' z kategorii bloczków czujników (kat. Sensing) jako wejście warunkowe funkcji 'if'.

Program zawiera bloczek 'if' z kategorii kontrolowania przepływu programu (kat. Control) z bloczkiem o diamentowym kształcie 'obstacle detected' z kategorii bloczków czujników (kat. Sensing) jako wejście warunkowe funkcji 'if'.

The 'drive strain detected' block will only detect strain while the motors are running.

**Znaczenie:**

Dosłownie: Bloczek 'Wykryto obciążenie napędu' wykryje obciążenie tylko wtedy, gdy silniki pracują.

Bloczek o diamentowym kształcie 'drive strain detected' z kategorii bloczków czujników (kat. Sensing) może zostać wykorzystany do wykrywania obciążenia i zatrzymania silników Edisona, kiedy ten wykonuje instrukcje jazdy i poruszania się. Jeśli silniki robota się nie ruszają, Edison może wykryć takie obciążenie.

**Przykłady:**

Program zawiera bloczek 'if' z kategorii kontrolowania przepływu programu (kat. Control) z bloczkiem o diamentowym kształcie 'drive strain detected' z kategorii bloczków czujników (kat. Sensing) jako wejście warunkowe funkcji 'if'.

Istnieje podprogram załączany przez wykrycie kłaśnięcia – 'clap detected' – z kategorii bloczków zdarzeń startowych (kat. Events), który zawiera w sobie bloczek 'if' z kategorii kontroli przepływu programu. Bloczek 'if' zawiera w sobie wejście warunkowe w postaci bloczka o diamentowym kształcie 'drive strain detected' z kategorii czujników (kat. Sensing).

The 'drive strained' event can only trigger while the motors are running.

**Znaczenie:**

Dosłownie: Zdarzenie startowe 'Obciążenie napędu' może zostać uruchomione kiedy pracują silniki.

Bloczek zdarzenia startowego 'drive strained' (kat. Events) może zostać użyty do uruchomienia podprogramu, jeśli silniki Edisona zatrzymają się pod wpływem obciążenia, podczas wykonywania instrukcji jazdy. Jeśli jednak silniki robota się nie poruszają, Edison nie wykryje zatrzymania przez co podprogram się nie uruchomi.

**Przykłady:**

Istnieje podprogram załączany przez wykrycie zatrzymania silnika – 'drive strained' – z kategorii bloczków zdarzeń startowych (kat. Events), jednocześnie istnieją bloczki z instrukcjami jazdy w programie głównym.

Istnieje podprogram załączany przez wykrycie zatrzymania silnika – 'drive strained' – z kategorii bloczków zdarzeń startowych (kat. Events), jednak w programie głównym nie istnieją bloczki z instrukcjami jazdy.

The 'set both motors' block will only turn the motors on. Make sure there are additional blocks used in the program to control the motor's duration.

**Znaczenie:**

Dosłownie: Blok 'ustaw oba silniki' jedynie załączy silniki. Upewnij się, że w programie są dodatkowe bloczki kontrolujące czas pracy silników.

Bloczek 'set both motors' z kategorii napędowej (kat. Drive) załączy silniki Edisona, ale nie określa wymaganych dla silnika do pracy danych wejściowych (parametrów które mają doprecyzować instrukcję ruchu silników). Przez to, robot załączy silniki i od razu przejdzie do następnego bloczka w programie. Jeśli nie ma żadnego bloczka po 'set both motors', program zakończy się bez ruchu Edisona. Jeśli po bloczku 'set both motors' są inne bloki określające ruch silników, Edison wykona je w trakcie jazdy.

**Przykłady:**

Program główny zawiera blok 'set both motors' z kategorii napędowej (kat. Drive).

Istnieje podprogram załączany przez wykrycie klaśnięcia – 'clap detected' – z kategorii bloczków zdarzeń startowych (kat. Events), zawierający bloczek 'set both motors' z kategorii napędowej (kat. Drive).

The 'set left motor' block will only turn the motors on. Make sure there are additional blocks used in the program to control the motor's duration.

**Znaczenie:**

Dosłownie:

Bloczek 'set left motor' z kategorii napędowej (kat. Drive) załączy lewy silnik Edisona, ale nie określa wymaganych dla silnika do pracy danych wejściowych (parametrów które mają doprecyzować instrukcję ruchu silników). Przez to, robot załączy silnik i od razu przejdzie do następnego bloczka w programie. Jeśli nie ma żadnego bloczka po 'set left motor', program zakończy się bez ruchu Edisona. Jeśli po bloczku 'set left motor' są inne bloki określające ruch silnika, Edison wykona je w trakcie jazdy.

**Przykłady:**

Program główny zawiera blok 'set left motor' z kategorii napędowej (kat. Drive).

Istnieje podprogram załączany przez wykrycie klaśnięcia – 'clap detected' – z kategorii bloczków zdarzeń startowych (kat. Events), zawierający bloczek 'set left motor' z kategorii napędowej (kat. Drive).

The 'set right motor' block will only turn the motors on. Make sure there are additional blocks used in the program to control the motor's duration.

**Znaczenie:**

Dosłownie:

Bloczek 'set right motor' z kategorii napędowej (kat. Drive) załączy prawy silnik Edisona, ale nie określa wymaganych dla silnika do pracy danych wejściowych (parametrów które mają doprecyzować instrukcję ruchu silników). Przez to, robot załączy silnik i od razu przejdzie do następnego bloczka w programie. Jeśli nie ma żadnego bloczka po 'set right motor', program zakończy się bez ruchu Edisona. Jeśli po bloczku 'set right motor' są inne bloki określające ruch silnika, Edison wykona je w trakcie jazdy.

**Przykłady:**

Program główny zawiera blok 'set right motor' z kategorii napędowej (kat. Drive).

Istnieje podprogram załączany przez wykrycie kłaśnięcia – 'clap detected' – z kategorii bloczków zdarzeń startowych (kat. Events), zawierający bloczek 'set right motor' z kategorii napędowej (kat. Drive).