

Blockly Kodowanie – pomoc.

Słowniczek:

Zmienna – posiada nazwę wywoływaną w programie oraz miejsce na przechowywanie wartości.

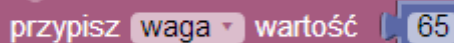
Instrukcja warunkowa – pozwala na wykonanie instrukcji w zależności od warunku zdefiniowanego przez koderka.

Pętla – pozwala na powtórzenie kodu określoną ilość razy.

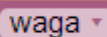
Procedura – umożliwia podział kodu na mniejsze części dzięki czemu łatwiej układać program. Czasami procedury zwane są funkcjami.

1. Zmienne

Blok przypisuje wartość do zmiennej i tworzy ją jeśli jeszcze nie istnieje

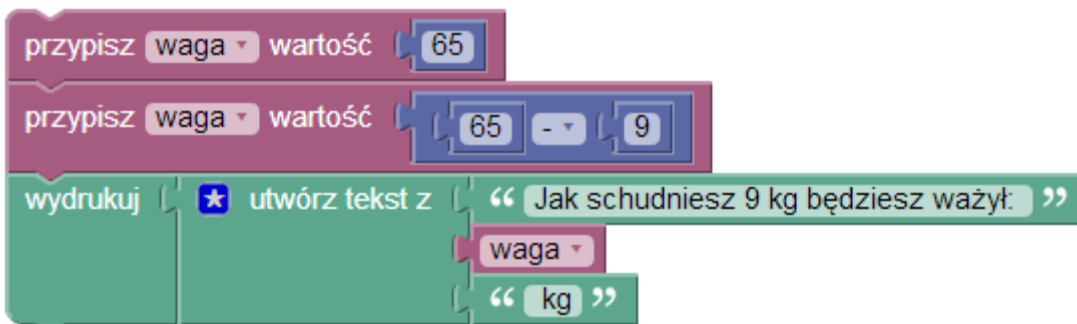


Przypisuje zmiennej waga wartość 65



zmienna waga zawiera teraz 65.

Przykład:

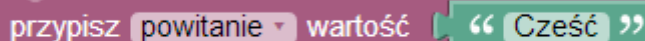


The code snippet consists of three blocks:

- A purple block: `przypisz waga wartość 65`
- A purple block: `przypisz waga wartość 65 - 9`
- A green block: `wydrukuj utwórz tekst z "Jak schudniesz 9 kg będziesz ważył: " waga " kg "`

Pierwszy rząd tworzy zmienną waga i przypisuje do niej wartość 65, drugi przypisuje do zmiennej waga wartość 65 – 9, trzeci wyświetla komunikat: ***Jak schudniesz 9 kg będziesz ważył: 56 kg.***

2. Tworzenie tekstu

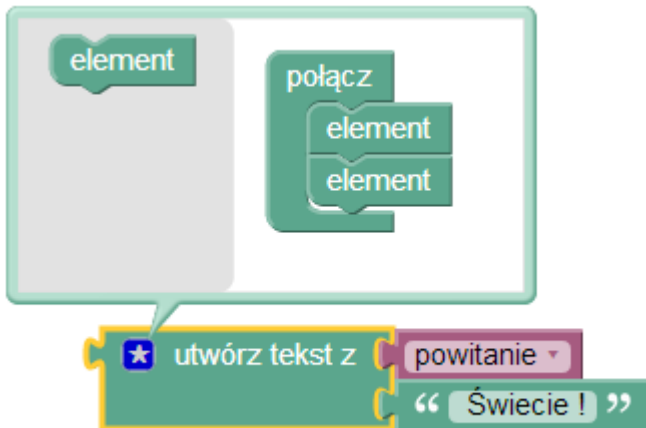


Kolejny (zielony) blok tworzy fragment tekstu "Cześć" i zapisuje go w zmiennej o nazwie ***powitanie***

Zielony blok zawiera dwa węzły i skleja tekst z zawartością zmiennej



Uzyskany komunikat to **Cześć Świecie !**



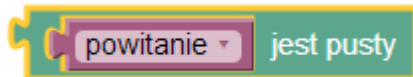
Aby zwiększyć liczbę wejść tekstowych kliknij gwiazdkę i przeciągnij element do bloku połącz:

Modyfikacje tekstu

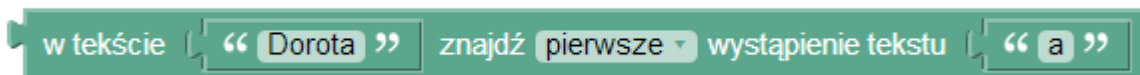
Blok zwracający długość podanego tekstu, w tym wypadku 3:



Sprawdza czy podany tekst jest pusty (zwraca prawdę lub fałsz)



Znajdowanie tekstu:



Zwraca wartość liczbową w tym wypadku 6

Wyodrębnianie tekstu:



W tym wypadku blok zwróci literę a



W tym wypadku również a (na drugiej pozycji).

w tekście `„ kodowanie jest super ”` pobierz pod słowo od `pierwszej litery` do `litery #` `3`

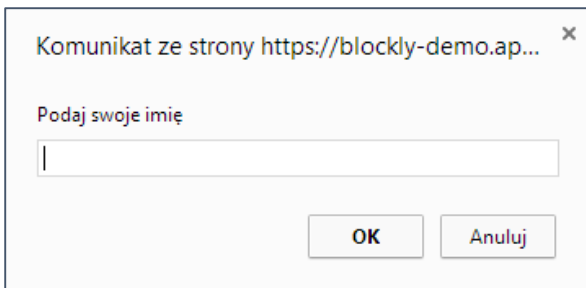
W tym wypadku zwróci pod słowo kod

Uzyskanie danych od użytkownika:

Kolejny blok tworzy wyskakujące okno z monitem o wprowadzenie nazwy. Wynik jest przechowywany w zmiennej `imie`:

przypisz `imie` wartość `poproś o tekst` z tą wiadomością `„ Podaj swoje imię ”`

Wynikiem będzie wyświetlenie okna które pozwala na wprowadzanie wartości do zmiennej w tym wypadku `imie`:



3. Instrukcja warunkowa

jeśli `element` \geq `0`
wykonaj `wydrukuj` `„ Podaneś prawidłową liczbę ”`

Jeśli zmienna `element` przechowuje wartość większą bądź równą `0` zostanie wyświetlony komunikat *Podaneś prawidłową liczbę*.

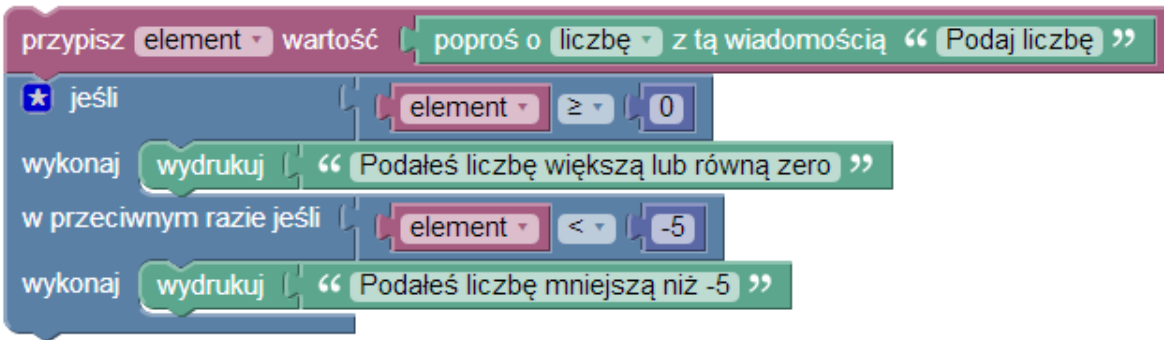


Kliknięcie gwiazdki pozwala na rozszerzenie instrukcji warunkowej:

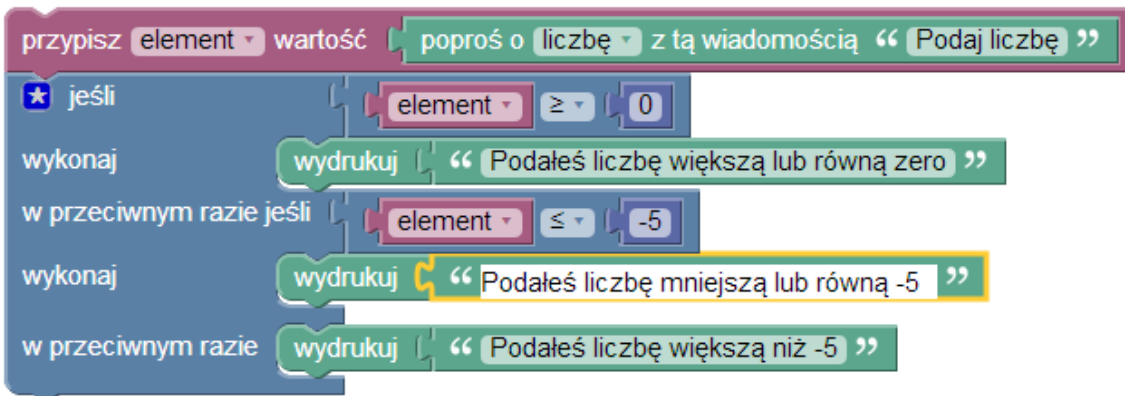
Przeciągnięcie bloku w przeciwnym razie. Zostanie wyświetlony komunikat *Podaneś nieprawidłową liczbę dla wartości niespełniających warunku*.

jeśli `element` \geq `0`
wykonaj `wydrukuj` `„ Podaneś prawidłową liczbę ”`
w przeciwnym razie `wydrukuj` `„ Podaneś nieprawidłową liczbę ”`

Następne instrukcje pokazują blok sprawdzający kolejny warunek w przypadku gdy pierwszy zwrócił fałsz.

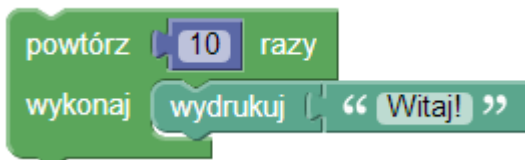


Brakuje komunikatu dla liczb większych niż -5. Blok z odpowiednim komunikatem i poprawkami:

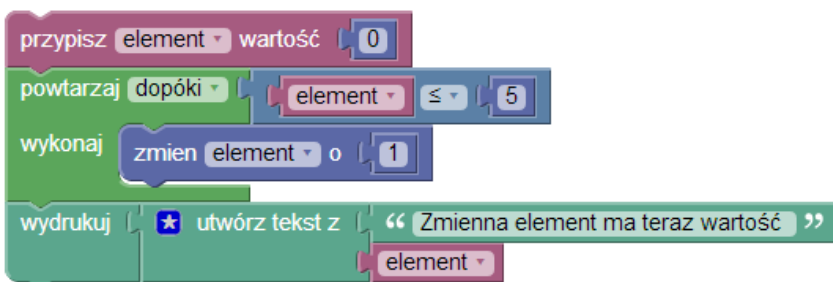


4. Pętle

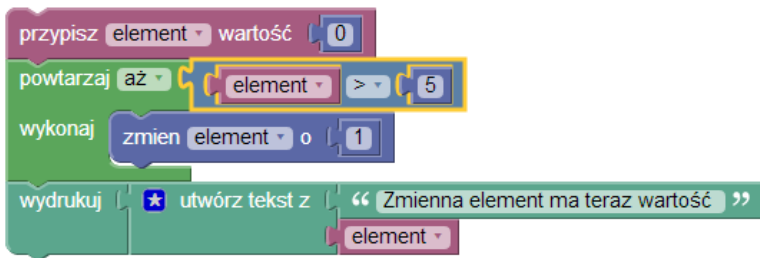
Prosta pętla która wyświetli 10 razy komunikat Witaj!:



W pierwszej linii zmiennej element przypisano wartość 0, w drugiej pętli sprawdza warunek - element mniejszy równy 5, dopóki warunek spełniony wartość elementu zwiększana jest o 1. Gdy warunek sterujący pętlą zwróci fałsz nastąpi wyjście z pętli i wyświetlenie komunikatu: *Zmienna element ma teraz wartość 6:*

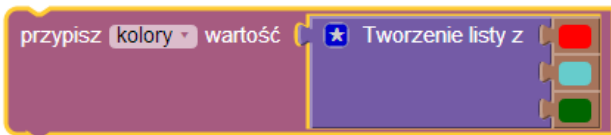
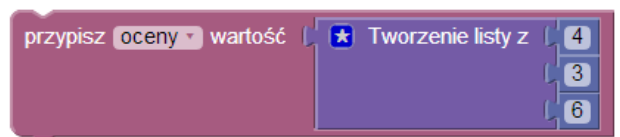
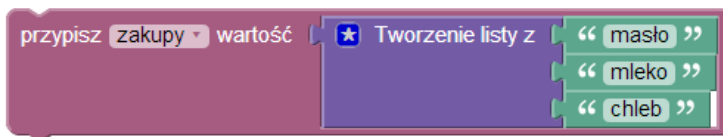


Pętla wykonująca się aż zostanie osiągnięty zakładany limit (jaki wpływ będzie miało użycie operatora większe równe ?). Jest to modyfikacja poprzedniego przykładu. Po wyjściu z pętli zostanie wyświetlony komunikat jak poprzednio:

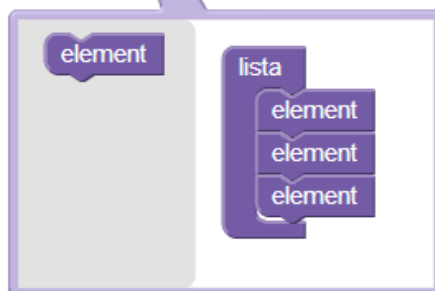


5. Listy

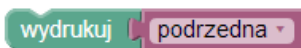
Utworzenie list różnego typu zawierających po 3 elementy:



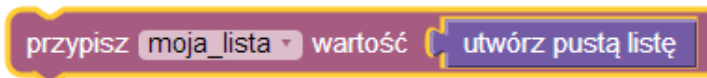
Zwiększenie liczby węzłów (przeciągnięcie elementu):



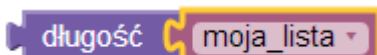
Wydrukowanie listy:



Utworzenie pustej listy:

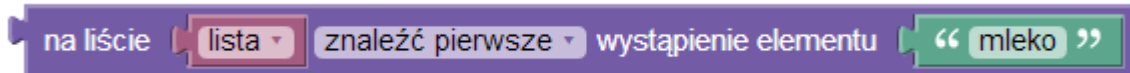


Pozyskanie liczby elementów na liście:

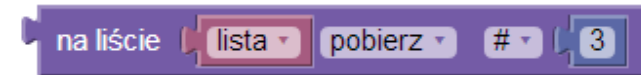


Znajdowanie elementu na liście:

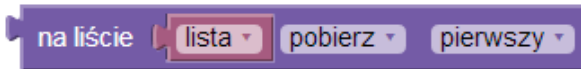
Poniższe komendy zwrócą numer pozycji poszukiwanego tekstu (pierwsze wystąpienie):



Pobranie trzeciego elementu na liście:



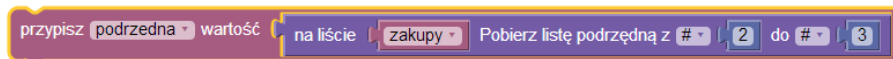
Pobranie pierwszego elementu na liście:



Usunięcie 2 elementu na liście:

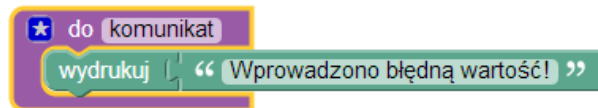


Utworzenie listy podrzędnej od elementu 2 do 3 na liście:

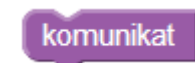


6. Procedury

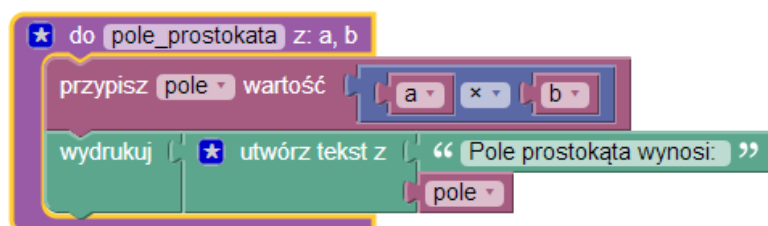
Definiowanie procedury:



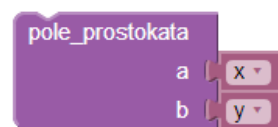
Wywołanie procedury:



Procedura z parametrami (parametry pozwalają na przekazanie wartości do procedury):



i jej wywołanie:

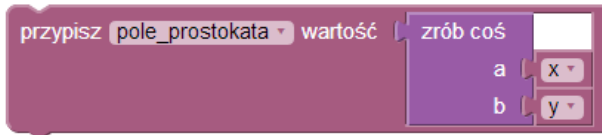


Procedura zwracająca wynik:

Definicja:



Wywołanie:



KONIEC