Zeszyt ćwiczeń do Dinoscript

Dinoscript z Pip

Napisz swój pierwszy program, stwórz trzy gry i rozwiąż zagadki matematyczne.



Francisco Claude-Faust

Cześć! Mam na imię Pip. Jestem triceratopsem z trzema rogami i klawiaturą. Otwórz Dinoscript i ucz się ze mną. Na każdej stronie znajdziesz coś krótkiego do wpisania, uruchomienia albo zmiany.

Pierwsze kroki
Strony 1-8

Gry tekstowe
Strony 9-11

Zagadki matematyczne
Strony 12-15

Copyright 2023-2026 Francisco Claude-Faust · Licencja Apache 2.0

Niech dinozaur przemówi

Pip mówi

“Mój pierwszy program powiedział „cześć”. Drugi poprosił o przekąskę.”

Otwórz Dinoscript. Wpisz poniższy wiersz w polu Kod lub kliknij Wypróbuj kod, jeśli czytasz tę lekcję w aplikacji.

Kliknij Uruchom albo naciśnij F5. Polecenie print mówi Dinoscriptowi, żeby coś wypisał. Tekst w podwójnym cudzysłowie pojawi się w panelu Wynik.

DINOSCRIPT

```
print("Cześć, dinozaurze!")
```

Co zobaczysz

```
Cześć, dinozaurze!
```

Wypróbuj

Zmień tekst w cudzysłowie i znów uruchom program. Czy uda ci się sprawić, żeby Pip przedstawiła się w dwóch wierszach?

Podpowiedź od Pip

Pod pierwszym print dopisz w nowym wierszu print("Mam na imię Pip!").

Twoje notatki

Miejsce na słowa i liczby

Pip mówi

“W plecaku mam jedną kanapkę i trzy kamyki. Zgadnij, co miało tam trafić.”

Tekst w podwójnym cudzysłowie nazywamy napisem. Liczby zapisujemy bez cudzysłowu. Przecinkami oddziel to, co chcesz wypisać w jednym wierszu.

Wiersz zaczynający się od # to komentarz. Możesz w nim zostawić wskazówkę dla kogoś, kto później przeczyta twój kod. Dinoscript pomija komentarze.

DINOSCRIPT

```
# Pip zagląda do kieszeni.
print("Kamyki w kieszeni: ", 3)
print("Gdy podniosę jeszcze dwa: ", 3 + 2)
```

Co zobaczysz

```
Kamyki w kieszeni: 3
Gdy podniosę jeszcze dwa: 5
```

Wypróbuj

Zapisz 3 + 2 w podwójnym cudzysłowie i uruchom program ponownie. Co się zmieniło? Dopisz komentarz o tym, skąd wzięły się kamyki.

Podpowiedź od Pip

W cudzysłowie Dinoscript widzi napis 3 + 2. Bez cudzysłowu dodaje liczby i wypisuje 5.

Twoje notatki

Zapisz wynik za pomocą let

Pip mówi

“Kilka jagód policzę na pazurach. Przy większej liczbie przyda mi się zmienna.”

Zmienna to nazwa, pod którą przechowujesz wartość. Pierwszy wiersz zapisuje 4 w zmiennej o nazwie jagody.

Następny wiersz odczytuje tę wartość, dodaje 3 i zapisuje nowy wynik pod tą samą nazwą. Ostatni wiersz znów odczytuje wartość.

DINOSCRIPT

```
let jagody = 4
let jagody = jagody + 3
print("Mam ", jagody, " jagód.")
```

Co zobaczysz

```
Mam 7 jagód.
```

Wypróbuj

Pip zjada dwie jagody, zanim je przeliczy. Dopisz nad print wiersz, który odejmie dwie jagody. Ile powinno zostać?

Podpowiedź od Pip

Wpisz let jagody = jagody - 2. Powinno zostać 5 jagód.

Twoje notatki

Poproś o odpowiedź

Pip mówi

“Wpisz 4, a obiecuję, że policzę wszystkie cztery dinozaury. Nawet jeśli są tylko w mojej wyobraźni.”

Polecenie print pokazuje pytanie. Puste nawiasy w readint() oznaczają, że niczego do nich nie wpisujesz. Funkcja czeka, aż podasz liczbę całkowitą.

Kliknij Uruchom. Gdy w panelu Wynik pojawi się pole, wpisz odpowiedź i naciśnij Enter. Jeśli program wciąż czeka, możesz kliknąć Zatrzymaj.

DINOSCRIPT

```
print("Ile dinozaurów przyjdzie na piknik?")
let goscie = readint()
print("Potrzebne talerzyki z liści: ", goscie * 2)
```

Wpisz 4

```
Ile dinozaurów przyjdzie na piknik?
Potrzebne talerzyki z liści: 8
```

Wypróbuj

Uruchom program dla 4 gości. Potem zmień go tak, żeby każdy gość potrzebował trzech talerzyków. Sprawdź też, co będzie przy 0 gości.

Podpowiedź od Pip

W pierwszej wersji programu dla 4 gości potrzeba 8 talerzyków. Po trzy na osobę to 12.

Twoje notatki

Zdecyduj, co się wydarzy

Pip mówi

“Żeby przejść przez strumień na piknik, potrzebuję przynajmniej dwóch kamieni.”

Instrukcja if sprawdza, czy warunek jest prawdziwy, czy fałszywy. Tutaj >= znaczy „większe lub równe”. Gdy warunek jest prawdziwy, wykona się kod po then.

Gdy warunek jest fałszywy, wykona się kod po else. Cały blok kończy się zapisem end if.

DINOSCRIPT

```
let kamienie = 2
if kamienie >= 2 then
  print("Możemy przejść!")
else
  print("Poszukaj jeszcze jednego kamienia.")
end if
```

Co zobaczysz

```
Możemy przejść!
```

Wypróbuj

Wypróbuj 1, 2 i 3 kamienie. Czy umiesz przewidzieć, jaki komunikat zobaczysz za każdym razem?

Podpowiedź od Pip

Przy 1 kamieniu potrzebujesz drugiego. Przy 2 albo 3 możesz przejść.

Twoje notatki

Policz świetliki za pomocą pętli

Pip mówi

“Raz, dwa, trzy, cztery, pięć. Świetliki, nie wierćcie się!”

Pętla while sprawdza warunek przed każdym powtórzeniem. Działa tak długo, jak długo warunek jest prawdziwy.

Ostatni wiersz pętli zmienia wartość zmiennej swietlik. Dzięki temu warunek w końcu stanie się fałszywy. Jeśli zapomnisz zmienić liczbę, możesz kliknąć Zatrzymaj.

DINOSCRIPT

```
let swietlik = 1
while swietlik <= 5 do
  print("Świetlik ", swietlik)
  let swietlik = swietlik + 1
end while
```

Wypróbuj

Policz wstecz od 5 do 1. Zmień wartość początkową zmiennej swietlik, warunek i wiersz, który zmienia tę wartość.

Podpowiedź od Pip

Zacznij od 5, użyj warunku swietlik >= 1 i za każdym razem odejmij 1.

Twoje notatki

Napisz własną funkcję

Pip mówi

“Na podjazd dla trzech dinozaurów potrzeba sześciu cegieł. Często tak liczę.”

Funkcja to kilka instrukcji, którym nadajesz wspólną nazwę. Funkcja ileCegiel ma jeden parametr: goscie. Gdy wywołasz ileCegiel(3), parametr goscie ma wartość 3.

Instrukcja output wewnątrz funkcji zwraca wartość do miejsca, w którym ją wywołano. Za pomocą print pokażesz tę wartość w panelu Wynik.

DINOSCRIPT

```
function ileCegiel(goscie)
  output goscie * 2
end function

print(ileCegiel(3))
print(ileCegiel(5))
```

Co zobaczysz

```
6
10
```

Wypróbuj

Na każdego dinozaura przypadają dwie cegły, a na cały podjazd jedna dodatkowa. Zmień funkcję w jednym miejscu, żeby poprawić oba wyniki.

Podpowiedź od Pip

Użyj output goscie * 2 + 1. Program powinien wypisać 7 i 11.

Twoje notatki

Wytrop błąd

Pip mówi

“Każdy błąd zostawia ślad. Niektóre chowają się za jednym cudzysłowem.”

Uruchom program. Potem usuń end if i uruchom go ponownie. Przeczytaj komunikat w panelu Wynik i kliknij Przejdź do błędu.

Dinoscript przeniesie cię do wiersza, w którym zaczyna się niedomknięty blok. Dopisz end if pod wierszem z print i uruchom program jeszcze raz.

DINOSCRIPT

```
let czasNaJedzenie = true
if czasNaJedzenie then
  print("Pora na przekąskę!")
end if
```

Co zobaczysz

```
Pora na przekąskę!
```

Wypróbuj

Teraz usuń cudzysłów na końcu tekstu w wierszu z print. Co trzeba poprawić według nowego komunikatu?

Podpowiedź od Pip

Wstaw cudzysłów z powrotem po wykrzykniku, przed nawiasem zamykającym.

Twoje notatki

Rozstaje wśród paproci

Pip mówi

“Mam mapę. Według niej paproć po lewej podejrzanie przypomina tę po prawej.”

Liczby działają tu jak przyciski. Gra czeka, aż wybierzesz drogę. Blok if pozwala jej opowiedzieć inne zakończenie dla każdej ścieżki.

DINOSCRIPT

```
print("1. Szlak kryształów")
print("2. Jaskinia porośnięta mchem")
let droga = readint()
if droga = 1 then
  print("Dostrzegasz bursztynowe kryształy!")
else
  if droga = 2 then
    print("Nieśmiały nietoperz pokazuje ci sekretne drzwi.")
  else
    print("Na mapie są tylko drogi 1 i 2.")
  end if
end if
```

Wypróbuj

Wypróbuj 1, 2 i 9. Potem dopisz pytanie, które zada nietoperz. Co ma się stać, jeśli gracz wpisze 1 na znak, że odpowiada „tak”?



Podpowiedź od Pip

Dodaj print i readint() w części kodu o jaskini, przed jej end if. Sprawdź odpowiedź za pomocą kolejnego if.

Ukryta skamielina

Pip mówi

“Pod kamieniem z numerem jest skamielina. Masz cztery próby, żeby ją znaleźć.”

Każdy strzał kosztuje jedną próbę. Pętla działa, póki są próby i zmienna znaleziona ma wartość false. Spróbuj wygrać, a potem przegrać.

DINOSCRIPT

```
let sekret = 7
let proby = 4
let znaleziona = false
print("Zgadnij numer kamienia, od 1 do 10.")
while proby > 0 and not znaleziona do
  let liczba = readint()
  if liczba = sekret then
    print("Skamielina znaleziona!")
    let znaleziona = true
  else
    if liczba < sekret then
      print("Za mało.")
    else
      print("Za dużo.")
    end if
  end if
  let proby = proby - 1
end while
if not znaleziona then
  print("Skamielina była pod kamieniem numer ", sekret, ".")
end if
```

Wypróbuj

Niech jedna osoba zmieni sekret, gdy druga nie patrzy. Dodaj komunikat z liczbą prób, które zostały.

Podpowiedź od Pip

Tuż po odjęciu jednej próby dodaj print("Pozostałe próby: ", proby). Użyj if, jeśli komunikat ma się nie pojawiać po wygranej.

Pip sprzedaje przekąski

Pip mówi

“Jagodzianki są po trzy muszelki. Masz dziesięć muszelek i wielki apetyt.”

Funkcja oblicza koszt zamówienia. Program sprawdza, czy podana liczba nie jest ujemna. Jeśli zamówienie jest poprawne, dowiesz się, czy wystarczy ci muszelek.

DINOSCRIPT

```
function koszt(bulki)
  output bulki * 3
end function

let muszelki = 10
print("Ile jagodzianek chcesz kupić?")
let zamowienie = readint()
let cena = koszt(zamowienie)
if zamowienie < 0 then
  print("Podaj zero albo większą liczbę.")
else
  if cena <= muszelki then
    print("Jagodzianki: ", zamowienie)
    print("Pozostałe muszelki: ", muszelki - cena)
  else
    print("Brakujące muszelki: ", cena - muszelki)
  end if
end if
```

Wypróbuj

Kup tyle jagodzianek, na ile cię stać. Potem wymyśl inną przekąskę i ustal jej cenę. Na ile sztuk wystarczy ci teraz muszelek?

Podpowiedź od Pip

Trzy jagodzianki kosztują 9 muszelek, więc zostanie ci jedna. Cztery kosztują 12, więc zabraknie ci dwóch.

Koszyki na jajka

Pip mówi

“Mam 23 jajka. Do każdego koszyka mieszczą się cztery. Pomożesz mi je zapakować?”

Użyj //, aby policzyć pełne koszyki, a %, żeby sprawdzić, ile jajek zostanie. Spróbuj obliczyć oba wyniki dla 23 jajek przed uruchomieniem kodu.

DINOSCRIPT

```
let jajka = 23
let pojemnosc = 4
print("Pełne koszyki: ", jajka // pojemnosc)
print("Pozostałe jajka: ", jajka % pojemnosc)
```

Wypróbuj

Masz 23 jajka. Dobierz koszyki mieszczące od 3 do 7 jajek tak, żeby zostały dokładnie 3 jajka. Czy pasują dwie różne pojemności?



Podpowiedź od Pip

Przy pojemności 4 albo 5 zostają 3 jajka. W pierwszym wypadku zapełnisz 5 koszyków, w drugim 4.

Twoje notatki

Schodki z kamyków

Pip mówi

“Moje schodki są z kamyków. Na pierwszy stopień poszedł jeden kamyk, na drugi dwa i tak dalej.”

Program dodaje kamyki z każdego kolejnego stopnia. Dla czterech stopni oblicza 1 + 2 + 3 + 4. Razem wychodzi 10.

DINOSCRIPT

```
let stopnie = 4
let stopien = 1
let suma = 0
while stopien <= stopnie do
  let suma = suma + stopien
  let stopien = stopien + 1
end while
print("Kamyki: ", suma)
```

Co zobaczysz

```
Kamyki: 10
```

Wypróbuj

Masz dokładnie 28 kamyków. Ile pełnych stopni da się zbudować? Sprawdź to w programie i wytłumacz komuś, skąd się wziął wynik.

Podpowiedź od Pip

Sprawdź 6 stopni, a potem 7. Na sześć idzie 21 kamyków. Na kolejny potrzeba jeszcze 7.

Twoje notatki

Sygnał dinozaurów

Pip mówi

“Zarycz przy wielokrotności trójki. Tupnij przy wielokrotności piątki. A co z piętnastką?”

Liczba jest wielokrotnością 3, gdy reszta z dzielenia przez 3 wynosi zero. Ten kod już wypisuje „Ryk” przy takich liczbach. Dodaj reguły dla „Tup” oraz „Ryk i tup”.

DINOSCRIPT

```
let n = 1
while n <= 15 do
  if n % 3 = 0 then
    print("Ryk")
  else
    print(n)
  end if
  let n = n + 1
end while
```

Wypróbuj

Niech program dla 3 wypisze „Ryk”, dla 5 „Tup”, a dla 15 „Ryk i tup”. Jeśli żadna reguła nie pasuje, wypisz samą liczbę. Gdy zadziała, sprawdź też 30.

Podpowiedź od Pip

Najpierw sprawdź oba warunki naraz za pomocą n % 3 = 0 and n % 5 = 0. Jeśli nie są spełnione jednocześnie, sprawdź każdy z osobna. Dla 30 powinno wyjść „Ryk i tup”.

Twoje notatki

Pip czyta w myślach

Pip mówi

“Policz w myślach. Założę się, że i tak odgadnę twoją liczbę.”

Zagraj przed zajrzeniem do kodu. Naciskaj Enter przy przerwach. Podaj wynik, gdy Pip o niego zapyta.

pause() czeka na Enter, a readint() odczytuje wynik. Pip odejmuje 6, bo 10 - 4 = 6. Potem używa // 2, żeby cofnąć podwojenie.

DINOSCRIPT

```
print("Pomyśl o liczbie całkowitej od 1 do 100. Nie zdradzaj jej!")
print("Naciśnij Enter, żeby zacząć.")
pause()
print("Pomnóż tę liczbę przez 2 i naciśnij Enter.")
pause()
print("Dodaj 10 i naciśnij Enter.")
pause()
print("Odejmij 4 i naciśnij Enter.")
pause()
print("Jaki wynik wyszedł?")
let wynik = readint()
if wynik < 8 or wynik > 206 or wynik % 2 != 0 then
  print("Sprawdź obliczenia i uruchom sztuczkę jeszcze raz.")
else
  print("Twoja sekretna liczba to ", (wynik - 6) // 2, "!")
end if
```

Wypróbuj

Pokaż tę sztuczkę komuś znajomemu. Potem zamiast 10 dodaj 12. Co jeszcze trzeba zmienić w kodzie, żeby sztuczka działała?

Podpowiedź od Pip

Teraz na końcu masz o 8 więcej. Odejmij 8 we wzorze i dopuść wyniki od 10 do 208.