A Dinoscript activity book

Dinoscript with Pip

Write your first program, make three games, and solve math puzzles.



Francisco Claude-Faust

I'm Pip, a triceratops with three horns and a keyboard. Open Dinoscript and come learn with me. Each page gives you one small thing to type, run, or change.

Start coding
Pages 1 to 8

Text games
Pages 9 to 11

Math puzzles
Pages 12 to 15

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

Make a dinosaur talk

Pip says

“My first program said hello. My second program asked for snacks.”

Open Dinoscript. Type the line below in Code, or choose Try code when you read this page in the app.

Click Run or press F5. The word print tells Dinoscript to show something. The words between the double quotes show up in Output.

DINOSCRIPT

```
print("Hello, dinosaur!")
```

What you'll see

```
Hello, dinosaur!
```

Try it

Change the words between the quotes. Run it again. Can you make Pip introduce herself in two lines?

Pip's hint

Try putting print("I am Pip!") on a new line under the first print.

Your notes

Give words and numbers a home

Pip says

“My bag has one sandwich and three rocks. Guess which I meant to pack.”

Text in double quotes is called a string. A number goes in bare. Put commas between the pieces you want to print on one line.

A line starting with # is a comment. It's a note you can leave for the next person who reads the code. Dinoscript skips it.

DINOSCRIPT

```
# Pip checks her pockets.
print("Rocks in my pocket: ", 3)
print("If I pick up two more: ", 3 + 2)
```

What you'll see

```
Rocks in my pocket: 3
If I pick up two more: 5
```

Try it

Put double quotes around 3 + 2 and run again. What changed? Then add a comment that tells me where the rocks came from.

Pip's hint

With quotes, Dinoscript prints the characters 3 + 2. Without quotes, it adds and prints 5.

Your notes

Keep score with let

Pip says

“I can count berries on my claws. Beyond that, I need a variable.”

A variable is a name for a value you want to keep. The first line puts 4 in a variable called berries.

The next line reads the value first, adds 3, and puts the new value back under the same name. The last line reads it again.

DINOSCRIPT

```
let berries = 4
let berries = berries + 3
print("I have ", berries, " berries.")
```

What you'll see

```
I have 7 berries.
```

Try it

Pip eats two berries before she counts them. Add a line above print that takes two away. What should the new total be?

Pip's hint

Write let berries = berries - 2. You should see 5 berries.

Your notes

Ask for an answer

Pip says

“If you type 4, I promise I will count the four imaginary dinosaurs.”

print asks the question. The empty parentheses in readint() tell you it takes no pieces of its own. It waits for you to type a whole number.

Click Run. When the input box lights up in Output, type your answer and press Enter. You can use Stop if your program is still waiting.

DINOSCRIPT

```
print("How many dinosaurs are coming?")
let guests = readint()
print("We need ", guests * 2, " leaf plates.")
```

Try entering 4

```
How many dinosaurs are coming?
We need 8 leaf plates.
```

Try it

Run with 4 guests. Then make each guest need three leaf plates. Try 0 guests, too.

Pip's hint

With the original code, 4 guests need 8 plates. With three plates each, they need 12.

Your notes

Choose what happens

Pip says

“I can reach the picnic branch if I have at least two stepping stones.”

if tests something that can be true or false. Here, >= means greater than or equal to. When the test is true, the lines after then run.

When the test is false, the lines after else run instead. end if closes the whole choice.

DINOSCRIPT

```
let stones = 2
if stones >= 2 then
  print("Across we go!")
else
  print("Find another stepping stone.")
end if
```

What you'll see

```
Across we go!
```

Try it

Try 1, 2, and 3 stones. Can you predict which message each number will show?

Pip's hint

With 1 you need another stone. With 2 or 3 you can cross.

Your notes

Count fireflies with a loop

Pip says

“One, two, three, four, five. Please stay still, fireflies.”

while checks its test before each trip through the block. The loop continues as long as the test is true.

The last line inside the loop changes fly, so the test eventually becomes false. You can click Stop if you forget to make the number change.

DINOSCRIPT

```
let fly = 1
while fly <= 5 do
  print("Firefly ", fly)
  let fly = fly + 1
end while
```

Try it

Count backward from 5 to 1. Change where fly starts, the test, and the update.

Pip's hint

Start at 5, use fly >= 1, and subtract 1 each trip.

Your notes

Make your own command

Pip says

“A slope for three dinosaurs needs six bricks. I use that trick a lot.”

A function is a few lines you give a name. bricksFor has one input, guests. Calling bricksFor(3) gives guests the value 3.

Inside a function, output sends a value back to the line that called it. print is how you show that value in the Output pane.

DINOSCRIPT

```
function bricksFor(guests)
  output guests * 2
end function

print(bricksFor(3))
print(bricksFor(5))
```

What you'll see

```
6
10
```

Try it

Each dinosaur now needs two bricks and the ramp needs one extra. Change the function once to fix both answers.

Pip's hint

Use output guests * 2 + 1. The two lines should say 7 and 11.

Your notes

Follow a bug to its hiding place

Pip says

“A bug is a clue. Some are small enough to hide behind one quote.”

Run this program once. Then remove end if and run it again. Read the message in Output and click Go to error.

Dinoscript jumps to the line that starts the unfinished block. Put end if back below the print line, then run again.

DINOSCRIPT

```
let hungry = true
if hungry then
  print("Snack time!")
end if
```

What you'll see

```
Snack time!
```

Try it

Next, remove the closing double quote on the print line. What does the new message ask you to fix?

Pip's hint

Put the quote back after the exclamation point and before the closing parenthesis.

Your notes

Fork in the Ferns

Pip says

“I packed a map. It says the fern on the left looks a lot like the fern on the right.”

Use the numbers as buttons. The game waits for a choice and uses an if block to tell a different ending for each path.

DINOSCRIPT

```
print("1. Crystal trail")
print("2. Mossy cave")
let path = readint()
if path = 1 then
  print("You spot amber crystals!")
else
  if path = 2 then
    print("A shy bat shows you a secret door.")
  else
    print("The trail map only shows 1 and 2.")
  end if
end if
```

Try it

Try 1, 2, and 9. Then make the bat ask a second question. What happens if the player says yes by typing 1?



Pip's hint

Add a new print and readint() inside the cave branch, before its end if. Use another if to check the answer.

The Secret Fossil

Pip says

“I hid a fossil under a numbered rock. You get four guesses.”

Every guess spends one turn. The loop keeps going while you have a turn left and found is false. Test both a win and a round that runs out of turns.

DINOSCRIPT

```
let secret = 7
let turns = 4
let found = false
print("Guess the fossil number from 1 to 10.")
while turns > 0 and not found do
  let guess = readint()
  if guess = secret then
    print("You found the fossil!")
    let found = true
  else
    if guess < secret then
      print("Too low.")
    else
      print("Too high.")
    end if
  end if
  let turns = turns - 1
end while
if not found then
  print("The fossil was under rock ", secret, ".")
end if
```

Try it

Let someone change secret while the other player looks away. Can you add a message that shows how many turns are left?

Pip's hint

Right after turns goes down, add print(turns, " turns left."). Put it in an if block if you want to hide it after a win.

Pip's Snack Stand

Pip says

“Berry buns are three shells each. You have ten shells and an excellent appetite.”

The function works out the bill. The shop checks for a negative order and tells you whether you can afford a valid one.

DINOSCRIPT

```
function bill(buns)
  output buns * 3
end function

let shells = 10
print("How many berry buns would you like?")
let order = readint()
let cost = bill(order)
if order < 0 then
  print("Order zero or more buns.")
else
  if cost <= shells then
    print("Buns: ", order)
    print("Shells left: ", shells - cost)
  else
    print("You need ", cost - shells, " more shells.")
  end if
end if
```

Try it

Try to buy the most buns you can. Then invent a new snack with a different price. How does that change what you can afford?

Pip's hint

Three buns cost 9 shells, so you have 1 shell left. Four cost 12, so you'll be 2 shells short.

The egg baskets

Pip says

“I've got 23 eggs. My baskets hold four eggs each. Help me pack them.”

Use // to count full groups and % to count what's left. Try to work out both answers for 23 eggs before you run the code.

DINOSCRIPT

```
let eggs = 23
let basketSize = 4
print("Full baskets: ", eggs // basketSize)
print("Eggs left: ", eggs % basketSize)
```

Try it

For 23 eggs, find a basket size from 3 to 7 that leaves exactly 3 eggs over. Can you find two different sizes that work?



Pip's hint

With sizes 4 and 5, the remainders are both 3. With size 4, there are 5 full baskets. With size 5, there are 4.

Your notes

The stair of stones

Pip says

“I built stairs from pebbles. Step one has one pebble, step two has two, and so on.”

This program adds each row to a running total. For four rows, it adds 1 + 2 + 3 + 4. The total is 10.

DINOSCRIPT

```
let rows = 4
let row = 1
let total = 0
while row <= rows do
  let total = total + row
  let row = row + 1
end while
print("Pebbles: ", total)
```

What you'll see

```
Pebbles: 10
```

Try it

You find a pile of exactly 28 pebbles. How many complete steps can you build? Find the number of rows with the program, then explain your answer to a friend.

Pip's hint

Try 6 rows and then 7. Six rows take 21 pebbles and the next row takes 7 more.

Your notes

The fossil signal

Pip says

“Roar on a multiple of three. Stomp on a multiple of five. What about fifteen?”

A number is a multiple of 3 when the remainder after dividing by 3 is zero. This starter catches Roar numbers. Add the Stomp and Roar Stomp rules.

DINOSCRIPT

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

Try it

Make 3 say Roar, 5 say Stomp, and 15 say Roar Stomp. Any number that matches neither rule should print as itself. Once it works, check 30.

Pip's hint

First test n % 3 = 0 and n % 5 = 0 together with and. Only if that fails should you check the individual rules. At 30, you should get Roar Stomp.

Your notes

Pip Reads Your Mind

Pip says

“Do the math in your head. I bet I can still guess your number.”

Play once before you peek at the code. Press Enter at each pause. Type your answer only when Pip asks for your final total.

Each pause() waits for Enter. The last step uses readint() to get the total. Pip subtracts 6 to undo the extra 10 minus 4, then uses // 2 to undo the doubling.

DINOSCRIPT

```
print("Think of a whole number from 1 to 100. Keep it secret!")
print("Press Enter when you're ready.")
pause()
print("Multiply it by 2, then press Enter.")
pause()
print("Add 10, then press Enter.")
pause()
print("Subtract 4, then press Enter.")
pause()
print("What number did you arrive at?")
let total = readint()
if total < 8 or total > 206 or total % 2 != 0 then
  print("Check your math, then run the trick again.")
else
  print("Your secret number was ", (total - 6) // 2, "!")
end if
```

Try it

Try it on a friend. Then change the addition step to 12. What else in the code must change for the trick to work?

Pip's hint

Now there are 8 extra. Subtract 8 in the formula and change the allowed totals to 10 through 208.