Un cuaderno de actividades de Dinoscript

Dinoscript con Pip

Escribe tu primer programa, crea tres juegos y resuelve desafíos matemáticos.



Francisco Claude-Faust

Soy Pip, un tricerátops con tres cuernos y un teclado. Abre Dinoscript y ven a aprender conmigo. En cada página encontrarás algo breve que escribir, ejecutar o cambiar.

Primeros pasos
Páginas 1 a 8

Juegos de texto
Páginas 9 a 11

Desafíos matemáticos
Páginas 12 a 15

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

Haz hablar a un dinosaurio

Dice Pip

“Mi primer programa dijo hola. El segundo pidió algo para comer.”

Abre Dinoscript. Escribe la línea de abajo en Código o elige Probar código si estás leyendo esta página en la aplicación.

Haz clic en Ejecutar o presiona F5. La palabra print le indica a Dinoscript que muestre algo. Las palabras entre comillas dobles aparecen en Salida.

DINOSCRIPT

```
print("¡Hola, dinosaurio!")
```

Lo que verás

```
¡Hola, dinosaurio!
```

Ponlo a prueba

Cambia las palabras entre comillas y vuelve a ejecutar el programa. ¿Puedes hacer que Pip se presente en dos líneas?

Pista de Pip

Prueba con print("¡Soy Pip!") en una línea nueva, debajo del primer print.

Tus notas

Dales un lugar a las palabras y los números

Dice Pip

“En mi mochila hay un sándwich y tres piedras. Adivina qué quería guardar.”

El texto entre comillas dobles se llama cadena de texto. Los números se escriben sin comillas. Separa con comas lo que quieras mostrar en una misma línea.

Una línea que empieza con # es un comentario. Puedes dejarlo como nota para quien lea el código después. Dinoscript se lo salta.

DINOSCRIPT

```
# Pip revisa sus bolsillos.
print("Piedras en mi bolsillo: ", 3)
print("Si recojo dos más: ", 3 + 2)
```

Lo que verás

```
Piedras en mi bolsillo: 3
Si recojo dos más: 5
```

Ponlo a prueba

Pon 3 + 2 entre comillas dobles y vuelve a ejecutar el programa. ¿Qué cambió? Luego agrega un comentario que me cuente de dónde salieron las piedras.

Pista de Pip

Con comillas, Dinoscript muestra el texto 3 + 2. Sin comillas, hace la suma y muestra 5.

Tus notas

Lleva la cuenta con let

Dice Pip

“Puedo contar unas cuantas bayas con mis garras. Para contar más, necesito una variable.”

Una variable es un nombre para un valor que quieres guardar. La primera línea guarda un 4 en una variable llamada bayas.

La siguiente línea lee el valor, le suma 3 y guarda el resultado en la misma variable. La última línea vuelve a leer el valor.

DINOSCRIPT

```
let bayas = 4
let bayas = bayas + 3
print("Tengo ", bayas, " bayas.")
```

Lo que verás

```
Tengo 7 bayas.
```

Ponlo a prueba

Pip se come dos bayas antes de contarlas. Agrega una línea antes de print para restarlas. ¿Cuál debería ser el nuevo total?

Pista de Pip

Escribe let bayas = bayas - 2. Deberías ver que quedan 5 bayas.

Tus notas

Pide una respuesta

Dice Pip

“Si escribes 4, te prometo que voy a contar los cuatro dinosaurios imaginarios.”

print muestra la pregunta. Los paréntesis vacíos de readint() indican que no hay que ponerle nada adentro. La función espera a que escribas un número entero.

Haz clic en Ejecutar. Cuando aparezca el cuadro de entrada en Salida, escribe tu respuesta y presiona Enter. Puedes usar Detener si el programa sigue esperando.

DINOSCRIPT

```
print("¿Cuántos dinosaurios vienen?")
let invitados = readint()
print("Necesitamos ", invitados * 2, " platos de hojas.")
```

Prueba ingresando 4

```
¿Cuántos dinosaurios vienen?
Necesitamos 8 platos de hojas.
```

Ponlo a prueba

Prueba con 4 invitados. Luego cambia el programa para que cada invitado necesite tres platos. Prueba también con 0 invitados.

Pista de Pip

Con el código original, 4 invitados necesitan 8 platos. Con tres platos por invitado, necesitan 12.

Tus notas

Elige qué pasa

Dice Pip

“Para cruzar el arroyo y llegar al pícnic, necesito al menos dos piedras.”

if comprueba una condición que puede ser verdadera o falsa. Aquí, >= significa «mayor o igual que». Si la condición es verdadera, se ejecutan las líneas después de then.

Si la condición es falsa, se ejecutan las líneas después de else. end if cierra el bloque.

DINOSCRIPT

```
let piedras = 2
if piedras >= 2 then
  print("¡Podemos cruzar!")
else
  print("Busca otra piedra para cruzar.")
end if
```

Lo que verás

```
¡Podemos cruzar!
```

Ponlo a prueba

Prueba con 1, 2 y 3 piedras. ¿Puedes predecir qué mensaje aparecerá con cada número?

Pista de Pip

Con 1 piedra, necesitas otra. Con 2 o 3, puedes cruzar.

Tus notas

Cuenta luciérnagas con un ciclo

Dice Pip

“Una, dos, tres, cuatro, cinco. Quédense quietitas, por favor.”

while comprueba la condición antes de cada vuelta. El ciclo sigue mientras la condición sea verdadera.

La última línea del ciclo cambia el valor de luciernaga, así que en algún momento la condición se vuelve falsa. Si olvidas cambiar el número, puedes hacer clic en Detener.

DINOSCRIPT

```
let luciernaga = 1
while luciernaga <= 5 do
  print("Luciérnaga ", luciernaga)
  let luciernaga = luciernaga + 1
end while
```

Ponlo a prueba

Cuenta hacia atrás del 5 al 1. Cambia el valor inicial de luciernaga, la condición y la línea que actualiza su valor.

Pista de Pip

Empieza en 5, usa luciernaga >= 1 y resta 1 en cada vuelta.

Tus notas

Crea tu propia instrucción

Dice Pip

“Una rampa para tres dinosaurios necesita seis ladrillos. Uso mucho ese cálculo.”

Una función es un conjunto de instrucciones con un nombre. ladrillosPara tiene un parámetro llamado invitados. Al llamar a ladrillosPara(3), invitados toma el valor 3.

Dentro de una función, output devuelve un valor a la línea que hizo la llamada. Usa print para mostrar ese valor en Salida.

DINOSCRIPT

```
function ladrillosPara(invitados)
  output invitados * 2
end function

print(ladrillosPara(3))
print(ladrillosPara(5))
```

Lo que verás

```
6
10
```

Ponlo a prueba

Cada dinosaurio necesita dos ladrillos y la rampa necesita uno extra. Haz un solo cambio en la función para corregir los dos resultados.

Pista de Pip

Usa output invitados * 2 + 1. Las dos líneas deberían mostrar 7 y 11.

Tus notas

Sigue la pista de un error

Dice Pip

“Un error te da una pista. Hay errores tan chicos que se esconden detrás de una comilla.”

Ejecuta el programa una vez. Luego borra end if y vuelve a ejecutarlo. Lee el mensaje en Salida y haz clic en Ir al error.

Dinoscript salta a la línea donde empieza el bloque que quedó abierto. Vuelve a escribir end if debajo de la línea con print y ejecuta el programa otra vez.

DINOSCRIPT

```
let tengoHambre = true
if tengoHambre then
  print("¡Hora de la colación!")
end if
```

Lo que verás

```
¡Hora de la colación!
```

Ponlo a prueba

Ahora borra la comilla doble de cierre en la línea con print. ¿Qué te pide corregir el nuevo mensaje?

Pista de Pip

Vuelve a poner la comilla después del signo de exclamación y antes del paréntesis de cierre.

Tus notas

Dos caminos entre los helechos

Dice Pip

“Traje un mapa. Dice que el helecho de la izquierda se parece mucho al de la derecha.”

Usa los números como botones. El juego espera tu elección y usa un bloque if para contar un final distinto según el camino que tomes.

DINOSCRIPT

```
print("1. Sendero de cristales")
print("2. Cueva con musgo")
let camino = readint()
if camino = 1 then
  print("¡Ves cristales de ámbar!")
else
  if camino = 2 then
    print("Un murciélago tímido te muestra una puerta secreta.")
  else
    print("El mapa solo muestra el 1 y el 2.")
  end if
end if
```

Ponlo a prueba

Prueba con 1, 2 y 9. Luego haz que el murciélago haga otra pregunta. ¿Qué pasa si respondes que sí con un 1?



Pista de Pip

Agrega otro print y un readint() en el bloque de la cueva, antes de su end if. Usa otro if para revisar la respuesta.

El fósil secreto

Dice Pip

“Escondí un fósil debajo de una piedra numerada. Tienes cuatro intentos.”

Cada intento gasta un turno. El ciclo sigue mientras queden turnos y encontrado sea false. Prueba ganar y también quedarte sin turnos.

DINOSCRIPT

```
let secreto = 7
let turnos = 4
let encontrado = false
print("Adivina el número de la piedra, del 1 al 10.")
while turnos > 0 and not encontrado do
  let intento = readint()
  if intento = secreto then
    print("¡Encontraste el fósil!")
    let encontrado = true
  else
    if intento < secreto then
      print("El número es mayor.")
    else
      print("El número es menor.")
    end if
  end if
  let turnos = turnos - 1
end while
if not encontrado then
  print("El fósil estaba bajo la piedra ", secreto, ".")
end if
```

Ponlo a prueba

Pídele a alguien que cambie secreto sin que la otra persona mire. ¿Puedes agregar un mensaje que muestre cuántos turnos quedan?

Pista de Pip

Justo después de restar un turno, agrega print("Turnos que quedan: ", turnos). Ponlo en un if si quieres ocultarlo cuando alguien gane.

El puesto de colaciones de Pip

Dice Pip

“Cada pancito de bayas cuesta tres conchitas. Tienes diez conchitas y mucho apetito.”

La función calcula cuánto cuesta el pedido. El puesto revisa si la cantidad es negativa. Si el pedido es válido, te dice si te alcanzan las conchitas.

DINOSCRIPT

```
function cobrar(panes)
  output panes * 3
end function

let conchitas = 10
print("¿Cuántos pancitos de bayas quieres?")
let pedido = readint()
let costo = cobrar(pedido)
if pedido < 0 then
  print("Pide cero pancitos o más.")
else
  if costo <= conchitas then
    print("Pancitos: ", pedido)
    print("Conchitas que quedan: ", conchitas - costo)
  else
    print("Te faltan ", costo - conchitas, " conchitas.")
  end if
end if
```

Ponlo a prueba

Compra todos los pancitos que puedas. Luego inventa una colación con otro precio. ¿Qué puedes comprar ahora?

Pista de Pip

Tres pancitos cuestan 9 conchitas y te queda una. Cuatro cuestan 12 y te faltan dos.

Las canastas de huevos

Dice Pip

“Tengo 23 huevos. En cada canasta caben cuatro. ¿Me ayudas a guardarlos?”

Usa // para contar los grupos completos y % para saber cuánto sobra. Intenta calcular ambos resultados para 23 huevos antes de ejecutar el código.

DINOSCRIPT

```
let huevos = 23
let capacidad = 4
print("Canastas llenas: ", huevos // capacidad)
print("Huevos que sobran: ", huevos % capacidad)
```

Ponlo a prueba

Para 23 huevos, busca una capacidad entre 3 y 7 que deje exactamente 3 huevos de sobra. ¿Puedes encontrar dos capacidades que sirvan?



Pista de Pip

Con capacidad para 4 o 5 huevos sobran 3. En canastas de 4, llenas 5. En canastas de 5, llenas 4.

Tus notas

La escalera de piedras

Dice Pip

“Construí una escalera con piedritas. El primer peldaño tiene una, el segundo tiene dos y así sigue.”

Este programa va sumando las piedritas de cada peldaño. Con cuatro peldaños, suma 1 + 2 + 3 + 4. El total es 10.

DINOSCRIPT

```
let escalones = 4
let fila = 1
let total = 0
while fila <= escalones do
  let total = total + fila
  let fila = fila + 1
end while
print("Piedritas: ", total)
```

Lo que verás

```
Piedritas: 10
```

Ponlo a prueba

Encuentras exactamente 28 piedritas. ¿Cuántos peldaños completos puedes construir? Usa el programa para averiguarlo y cuéntale a alguien cómo llegaste a la respuesta.

Pista de Pip

Prueba con 6 peldaños y luego con 7. Para seis necesitas 21 piedritas. Para el siguiente necesitas 7 más.

Tus notas

La señal fósil

Dice Pip

“Ruge con los múltiplos de tres y da un pisotón con los de cinco. ¿Qué haces con el quince?”

Un número es múltiplo de 3 si al dividirlo por 3 el resto es cero. El código inicial ya muestra Rugido cuando corresponde. Agrega las reglas para Pisotón y Rugido y pisotón.

DINOSCRIPT

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

Ponlo a prueba

Haz que el 3 muestre «Rugido», el 5 «Pisotón» y el 15 «Rugido y pisotón». Si no se cumple ninguna regla, muestra el número. Cuando funcione, prueba con 30.

Pista de Pip

Primero comprueba si se cumplen las dos condiciones juntas con n % 3 = 0 and n % 5 = 0. Si no se cumplen ambas, revisa cada una por separado. Con 30 debería aparecer «Rugido y pisotón».

Tus notas

Pip te lee la mente

Dice Pip

“Haz la cuenta en tu cabeza. Apuesto a que igual adivino tu número.”

Juega una vez antes de mirar el código. Presiona Enter en cada pausa. Escribe tu respuesta solo cuando Pip te pida el total final.

Cada pause() espera a que presiones Enter. Al final, readint() recoge el total. Pip resta 6 para quitar lo que quedó de sumar 10 y restar 4, y usa // 2 para deshacer la multiplicación por 2.

DINOSCRIPT

```
print("Piensa en un número entero del 1 al 100. ¡No me lo digas!")
print("Presiona Enter cuando estés listo.")
pause()
print("Multiplícalo por 2 y presiona Enter.")
pause()
print("Súmale 10 y presiona Enter.")
pause()
print("Réstale 4 y presiona Enter.")
pause()
print("¿A qué número llegaste?")
let total = readint()
if total < 8 or total > 206 or total % 2 != 0 then
  print("Revisa tus cálculos y vuelve a probar el truco.")
else
  print("¡Tu número secreto era ", (total - 6) // 2, "!")
end if
```

Ponlo a prueba

Invita a alguien a jugar. Después cambia la instrucción para que sume 12. ¿Qué más debes cambiar en el código para que el truco funcione?

Pista de Pip

Ahora sobran 8. Resta 8 en la fórmula y cambia los límites del total a 10 y 208.