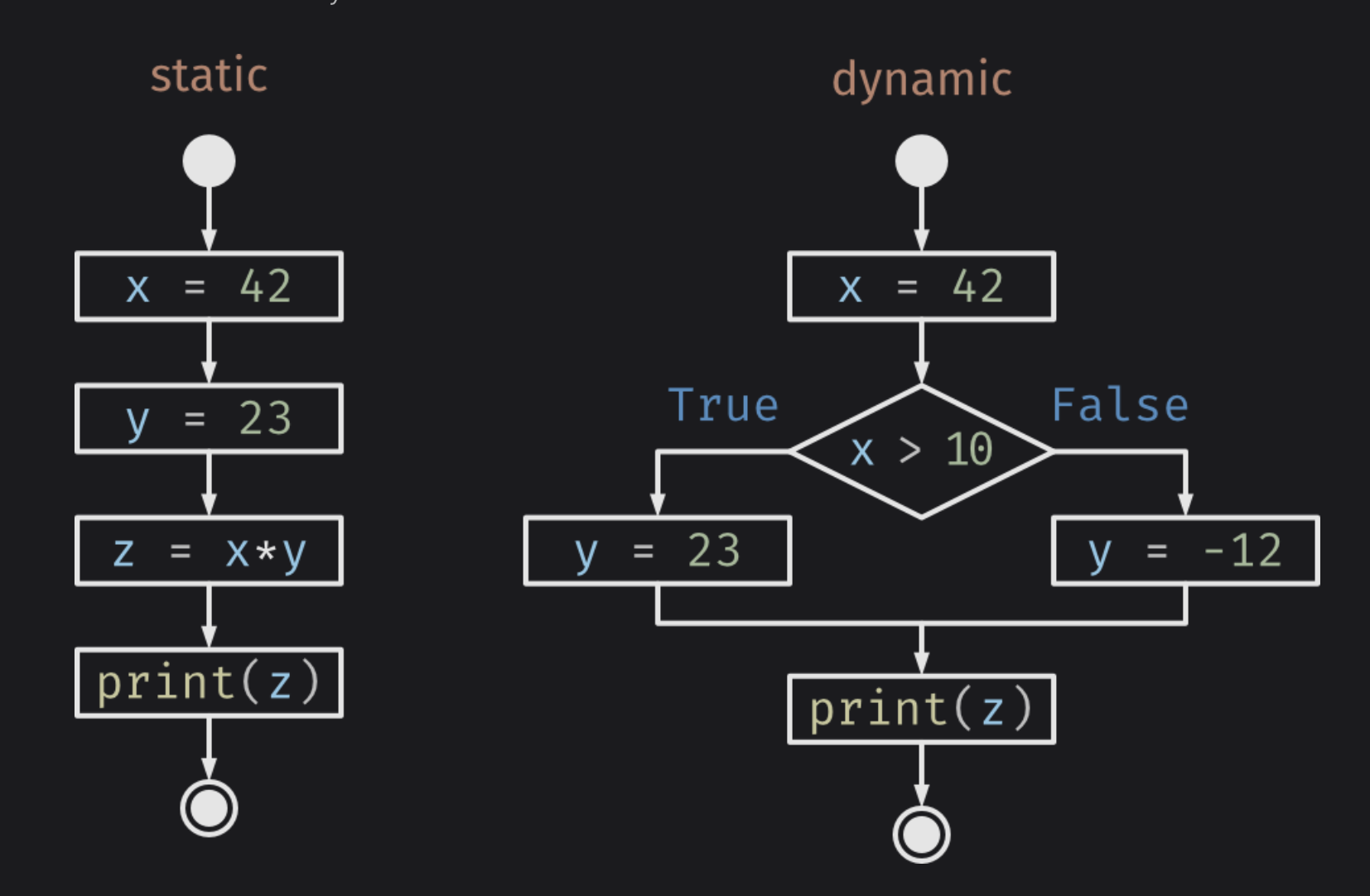
# Python Control Flow

## Static vs. Dynamic Program Flow

Dalam program statis semua pernyataan dijalankan tepat satu kali, satu demi satu. Sementara untuk pernyataan conditional (dynamic), beberapa bagian dari program tidak dijalankan sementara yang lainnya dijalankan.



## If.. Elif.. Else

Pengambilan keputusan diperlukan ketika kita ingin mengeksekusi kode hanya jika kondisi tertentu terpenuhi.

If... elif...else digunakan untuk proses pengambilan keputusan dari code yang dijalankan di Python.

Kondisi pernyataan if ... : harus mengevaluasi apakah hal tersebut bool True atau False.

### Comparison Operators

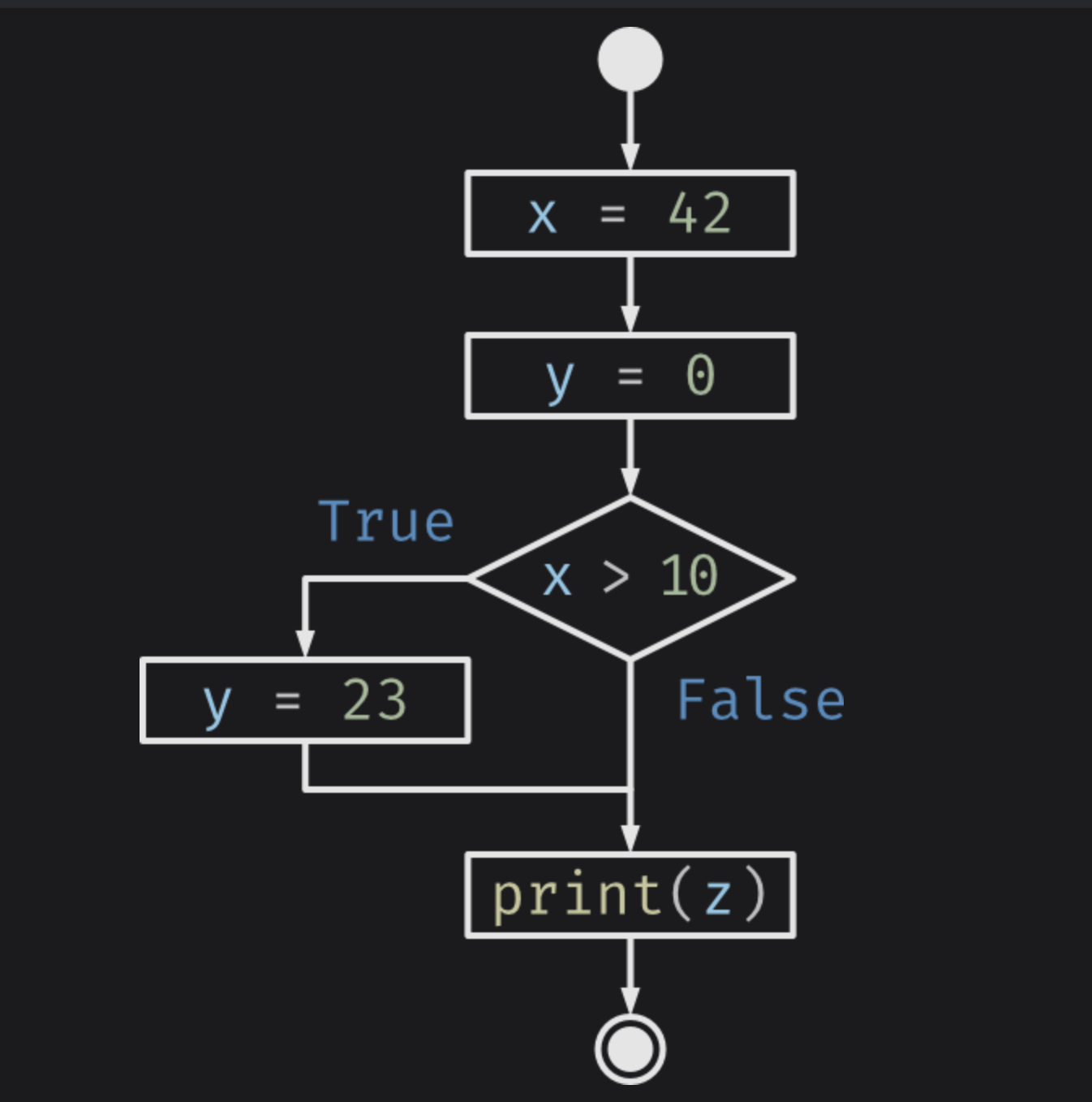
| Operators | Descriptions | Example |
| --- | --- | --- |
| > | Less than | x < 9 is True if x is less than 9 |
| < | Greater than | x > y is True if x is greater than y |
| == | Equal | x == 13 is True if and only if x is 13 |
| != | Not equal | x != y is True if and only if x is not y |
| <= | Less than or equal | x <= 23 is True if x is 23 or less |
| >= | Greater than or equal | x >= y is True if x is y or greater |

Example:

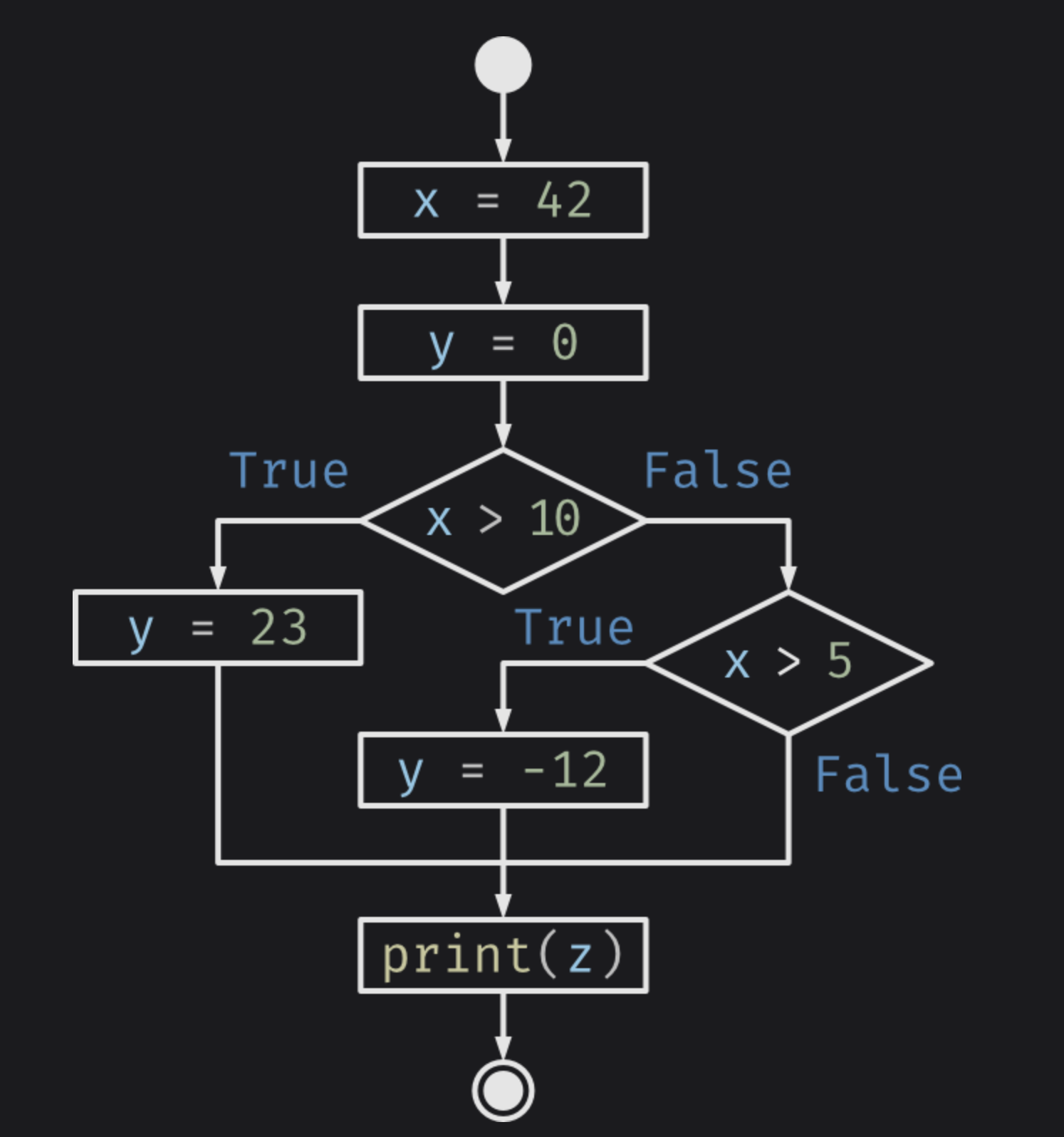
| if \_\_name\_\_ == "\_\_main\_\_":  x = 42   if x > 10:  y = 23  else:  y = -12   z = x\*y  print(z) |
| --- |

### Else-if: elif

If tidak harus diikuti oleh else. Yang akan diwakili dalam diagram dibawah ini:



| if \_\_name\_\_ == "\_\_main\_\_":  x = 42  y = 0   if x > 10:  y = 23   z = x\*y  print(z) |
| --- |

Kita juga dapat menambahkan pernyataan elif didalam if, yang hanya dievaluasi ketika pernyataan if di atas dievaluasi ke False: 

| if \_\_name\_\_ == "\_\_main\_\_":  x = 42  y = 0   if x > 10:  y = 23  elif x > 5:  y = -12   z = x\*y  print(z) |
| --- |

Kita dapat menambahkan rantai **elif** di bawah satu sama lain, dan masing-masing dari mereka hanya akan dievaluasi ketika kondisi sebelumnya mengevaluasi ke **False**. Jika ada pernyataan elif yang **True**, maka semua pernyataan berikut diabaikan. Setelah elif terakhir selesai, kita dapat menambahkan **else** yang akan dieksekusi ketika semua pernyataan sebelumnya **False**.

| if \_\_name\_\_ == "\_\_main\_\_":  name = "My Lovely Pet"  species = "cat"   if species == "dog":  print(name, "barks")  elif species == "duck":  print(name, "quacks")  elif species == "cat":  print(name, "meows")  elif species == "cow":  print(name, "moos")  else: print("I do not know what is this...") |
| --- |

### Short Circuit Evaluation

Ketika kita membuat kondisi yang terdiri dari beberapa pernyataan yang ditambahkan bersama dengan **and** & **or,** interpreter Python akan mengevaluasi pernyataan satu demi satu dari kiri ke kanan.

| if a > 10 and b < 2 |
| --- |

Python pertama-tama memeriksa apakah perbandingan a > 10 mengevaluasi ke True dan kemudian akan memeriksa b < 2, namun, jika > 10 mengevaluasi ke False maka b < 2 tidak akan diperiksa.

Ini disebut short circuit evaluation dan dapat menghemat waktu ketika mengevaluasi kondisi yang kompleks.

| if \_\_name\_\_ == "\_\_main\_\_":  x = 2  y = 3  if y > 4 and x == 1:  print("success")  else:  print("not success") |
| --- |

### Ternary Operator

Untuk menetapkan nilai ke variabel berdasarkan nilai variabel lain:

| if \_\_name\_\_ == "\_\_main\_\_":  name = "Angga"  age\_years = 23   if age\_years >= 18:  legal\_status = "dewasa"  else:  legal\_status = "anak-anak"   print(name, "adalah seorang", legal\_status) |
| --- |

Menetapkan nilai ke legal\_status memerlukan empat baris kode. Tetapi kita juga dapat menulisnya dalam satu baris dengan operator ternary.

| if \_\_name\_\_ == "\_\_main\_\_":  name = "Angga"  age\_years = 13   legal\_status = "dewasa" if age\_years > 18 else "anak-anak"   print(name, "adalah seorang", legal\_status) |
| --- |

Kita dapat menghapus variabel legal\_status dengan memindahkan operator ternary ke dalam pernyataan print(...):

| if \_\_name\_\_ == "\_\_main\_\_":  name = "Angga"  age\_years = 13   print(name, "adalah seorang", "dewasa" if age\_years > 18 else "anak-anak") |
| --- |

### 

### User Input

Kita dapat juga menerima input data dari user dalam program yang kita buat, misalnya:

| if \_\_name\_\_ == "\_\_main\_\_":  name = input("Please input name:")  age\_years = input("Your age:")   legal\_status = "dewasa" if int(age\_years) > 18 else "anak-anak"   print(name, "adalah seorang", legal\_status) |
| --- |

### Nested If

Python mendukung kondisi nested if, elif, dan else.

| price = 50 quantity = 5 amount = price\*quantity  if amount > 100:  if amount > 500:  print("Amount is greater than 500")  else:  if amount < 500 and amount > 400:  print("Amount is")  elif amount < 500 and amount > 300:  print("Amount is between 300 and 500")  else:  print("Amount is between 200 and 500") elif amount == 100:  print("Amount is 100") else:  print("Amount is less than 100") |
| --- |

### 

### Pass Statement

Statement If tidak boleh kosong, tetapi jika karena alasan tertentu, kita memiliki pernyataan if yang kosong, masukkan pernyataan **pass** untuk menghindari error.

pass adalah operasi null — ketika dijalankan, tidak ada yang terjadi. pass berguna sebagai placeholder, tetapi tidak ada kode yang perlu dieksekusi

| a = 33 b = 200  if b > a:  pass |
| --- |