Conditional Execution

Boolean Expressions

- A Boolean expression, sometimes called a predicate, may have only one of two possible values: false or true.
- We have seen that the simplest Boolean expressions are False and True.
- Boolean variable is also a Boolean expression.
- Other kinds of Boolean expressions use relational operators to compare two expressions.

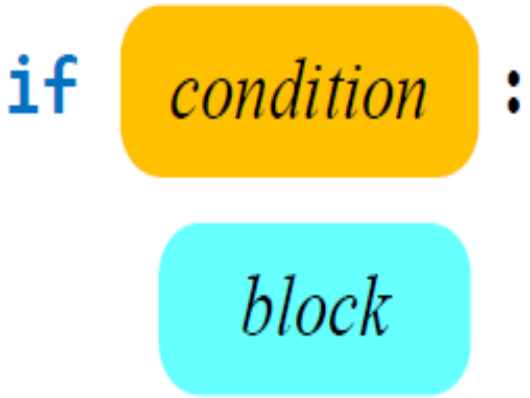
The Python relational operators

Expression	Meaning
x == y	True if $x = y$ (mathematical equality, not assignment); otherwise, false
x < y	True if $x < y$; otherwise, false
x <= y	True if $x \leq y$; otherwise, false
x > y	True if $x > y$; otherwise, false
x >= y	True if $x \geq y$; otherwise, false
x != y	True if $x \neq y$; otherwise, false

Expression	Value
`10 < 20`	`True`
`10 >= 20`	`False`
`x < 100`	`True` if `x` is less than 100; otherwise, `False`
`x != y`	`True` unless `x` and `y` are equal

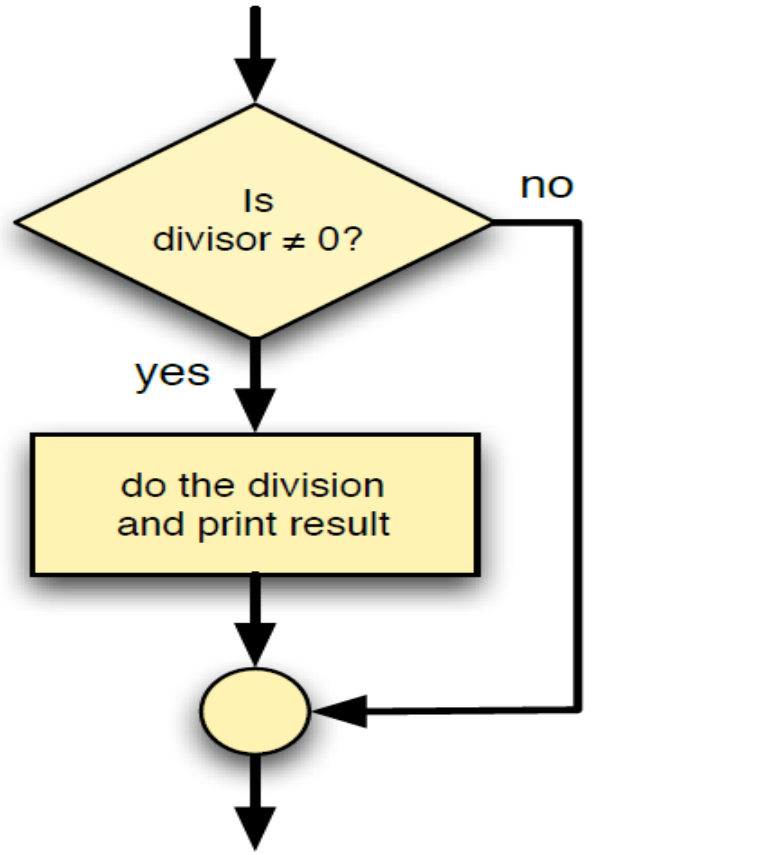
The Simple if Statement

The general form of the `if` statement is:



Example: **betterdivision.py**

flowchart



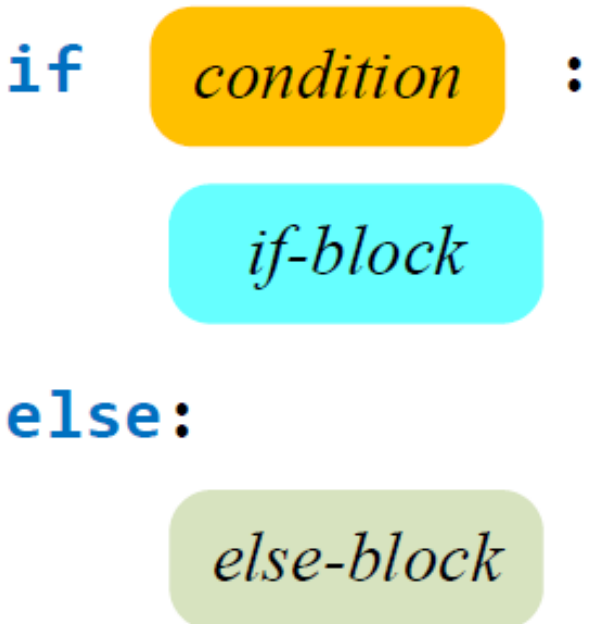
Example: betterdivision.py

```
print('Please enter two numbers to divide.')
dividend = int(input('Please enter the first number to divide: '))
divisor = int(input('Please enter the second number to divide: '))
#  If possible, divide them and report the result
if divisor != 0:
    print(dividend, '/', divisor, "=", dividend/divisor)
```

```
Please enter two numbers to divide.
Please enter the first number to divide: 32
Please enter the second number to divide: 8
32 / 8 = 4.0
```

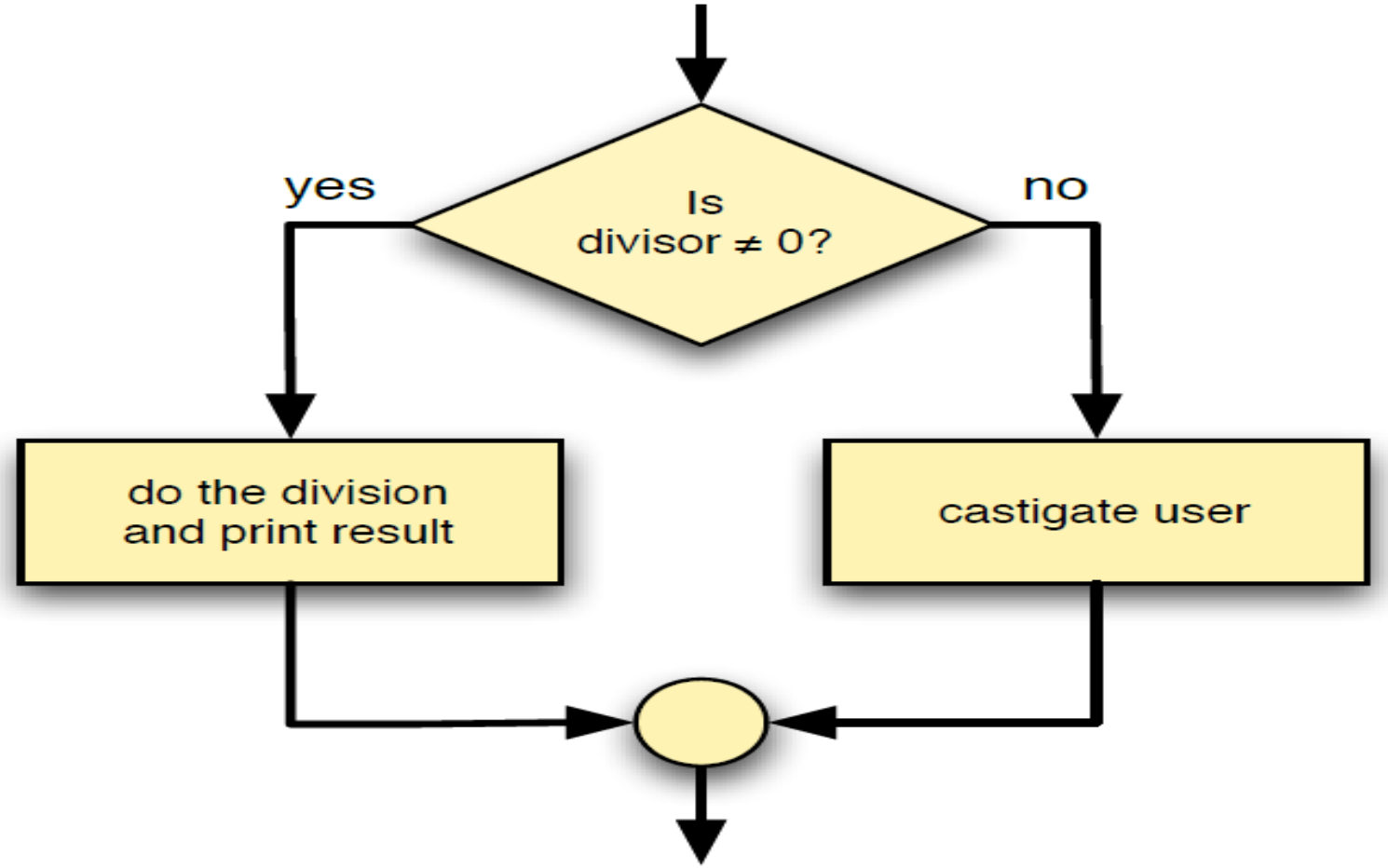
The if/else Statement

The general form of an `if/else` statement is



Example: betterfeedback.py

flowchart



Example: betterfeedback.py

```
#  Get two integers from the user
dividend = int(input('Please enter the number to divide: '))
divisor = int(input('Please enter dividend: '))
#  If possible, divide them and report the result
if divisor != 0:
    print(dividend, '/', divisor, "=", dividend/divisor)
else:
    print('Division by zero is not allowed')
```

```
Please enter the number to divide: 32
Please enter dividend: 0
Division by zero is not allowed
```

Nested Conditionals

- We can use these nested if statements to develop arbitrarily complex program logic.

```
value = int(input("Please enter an integer value in the range 0...10: ")
if value >= 0:          # First check
    if value <= 10:     # Second check
        print("In range")
print("Done")
```

Compound Boolean Expressions

- We can combine simple Boolean expressions, each involving one relational operator, into more complex Boolean expressions using the logical operators and, or, not.

Logical operators—e_1 and e_2 are Boolean expressions

e_1	e_2	e_1 and e_2	e_1 or e_2	not e_1
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Compound Boolean Expressions

```
value = int(input("Please enter an integer value in the range 0...10: ")
if value >= 0 and value <= 10: # Only one, slightly more complicated check
    print("In range")
print("Done")
```

Some points

Python allows an expression like

```
x <= y and y <= z
```

which means $x \leq y \leq z$ to be expressed more naturally:

```
x <= y <= z
```

Similarly, Python allows a programmer to test the equivalence of three variables as

```
if x == y == z:
    print('They are all the same')
```

Example: binaryconversion.py

The base 10 place value system

$\cdots$	$\boxed{4}$	$\boxed{7}$	$\boxed{3}$	$\boxed{4}$	$\boxed{0}$	$\boxed{6}$
$\cdots$	10^5	10^4	10^3	10^2	10^1	10^0
$\cdots$	100,000	10,000	1,000	100	10	1

$$\begin{aligned} 473,406 &= 4\times 10^5 + 7\times 10^4 + 3\times 10^3 + 4\times 10^2 + 0\times 10^1 + 6\times 10^0 \\ &= 400,000 + 70,000 + 3,000 + 400 + 0 + 6 \\ &= 473,406 \end{aligned}$$

The base 2 place value system

$\cdots$	$\boxed{1}$	$\boxed{0}$	$\boxed{0}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$
$\cdots$	2^5	2^4	2^3	2^2	2^1	2^0
$\cdots$	32	16	8	4	2	1

$$\begin{aligned} 100111_2 &= 1\times 2^5 + 0\times 2^4 + 0\times 2^3 + 1\times 2^2 + 1\times 2^1 + 1\times 2^0 \\ &= 32 + 0 + 0 + 4 + 2 + 1 \\ &= 39 \end{aligned}$$

Example: binaryconversion.py

```
#  Get number from the user
value = int(input("Please enter an integer value in the range 0...1023: "))
#  Create an empty binary string to build upon
binary_string = ''
#  Integer must be less than 1024
if 0 <= value < 1024:
    if value >= 512:
        binary_string += '1'
        value %= 512
    else:
        binary_string += '0'
    if value >= 256:
        binary_string += '1'
        value %= 256
    else:
        binary_string += '0'
    if value >= 128:
```

Example: binaryconversion.py

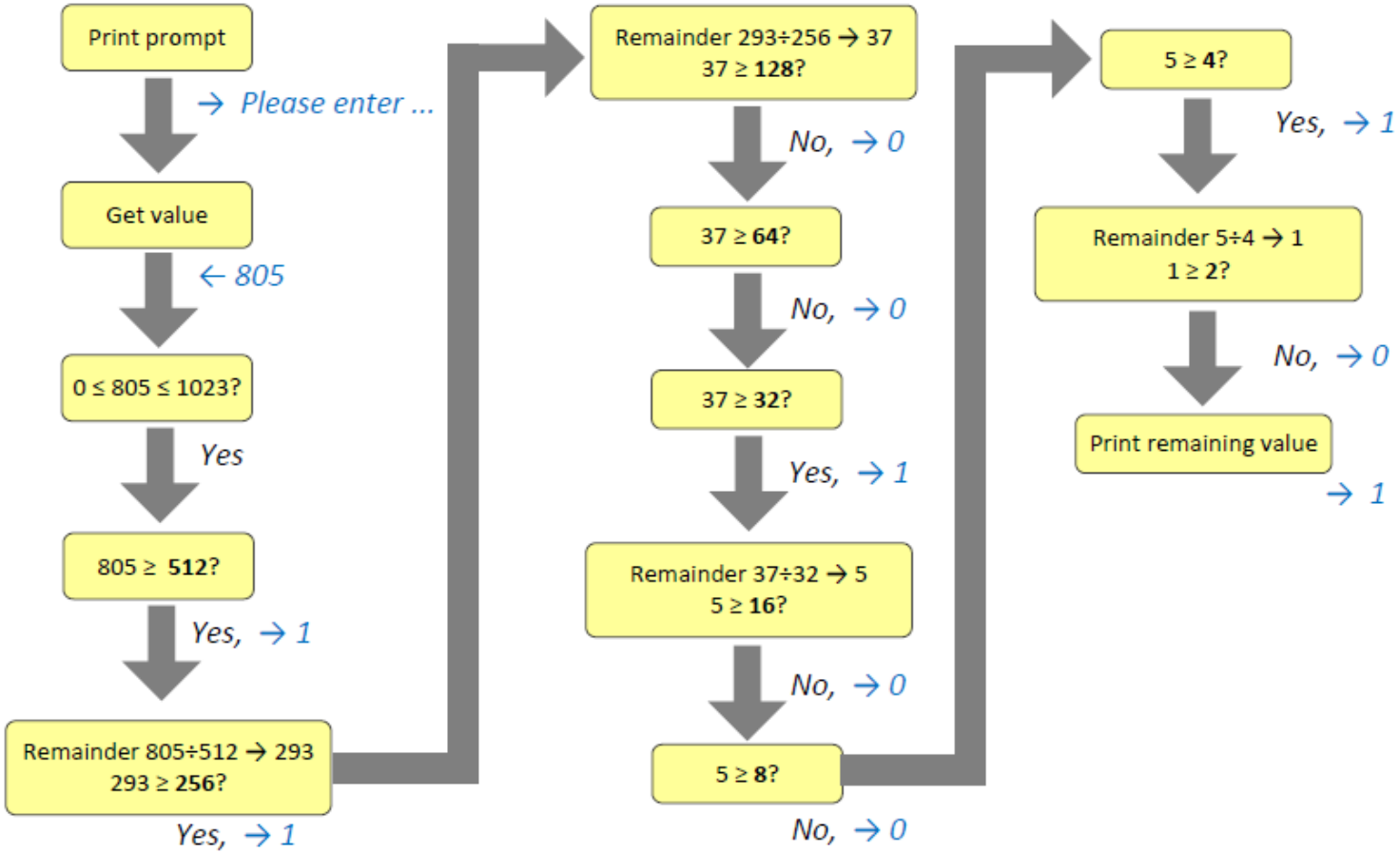
```
        binary_string += '1'
        value %= 128
    else:
        binary_string += '0'
    if value >= 64:
        binary_string += '1'
        value %= 64
    else:
        binary_string += '0'
    if value >= 32:
        binary_string += '1'
        value %= 32
    else:
        binary_string += '0'
    if value >= 16:
        binary_string += '1'
        value %= 16
    else:
        binary_string += '0'
    if value >= 8:
        binary_string += '1'
        value %= 8
    else:
        binary_string += '0'
```

Example: binaryconversion.py

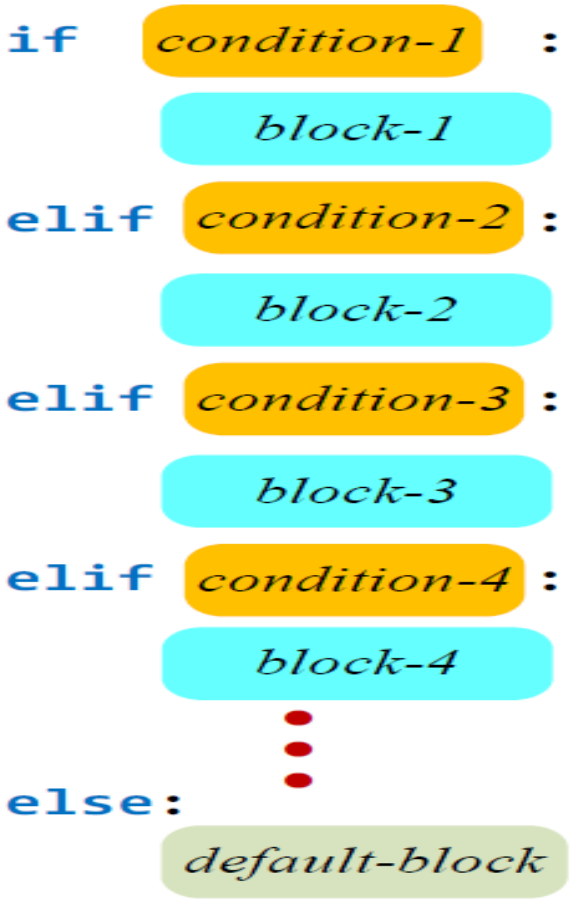
```
    if value >= 4:
        binary_string += '1'
        value %= 4
    else:
        binary_string += '0'
    if value >= 2:
        binary_string += '1'
        value %= 2
    else:
        binary_string += '0'
    binary_string += str(value)

#  Display the results
if binary_string != '':
    print(binary_string)
else:
    print('Cannot convert')
```

Example: binaryconversion.py



MULTI-WAY DECISION STATEMENTS



CONDITIONAL EXPRESSIONS

The general form of the conditional expression is

expression-1 **if** *condition* **else** *expression-2*

```
#  Get the dividend and divisor from the user
dividend = int(input('Enter dividend: '))
divisor = int(input('Enter divisor: '))
#  We want to divide only if divisor is not zero; otherwise,
#  we will print an error message
msg = dividend/divisor if divisor != 0 else 'Error, cannot divide by zero'
print(msg)
```