

Ifs, If-elses, If-elifs

If Statements

If statements allow you to run code only if a condition is met.

```
num = 97
if num >= 90:
    print "Hello"
print "World"
```

OUTPUT

Hello

World

If Statements

If statements will skip the code if the condition is not met.

```
num = 50
if num >= 90:
    print "Hello"
print "World"
```

OUTPUT
World

If Statements

Code inside an if statements must be indented

```
num = 97
if num >= 90:
    → print "Hello"
print "World"
```

OUTPUT

Hello

World

Relational Operators

Relational operators compare two values and return True or False

```
num = 150  
print (num <= 90)
```

OUTPUT
False

Relational frequently used operators

Name	Use
==	Determines if two values are equal
!=	Determines if two values are not equal
>	Determines if a value is greater than another
<	Determines if a value is less than another
>=	Determines if a value is greater than or equal to another
<=	Determines if a value is less than or equal to another

**open
ifs.py**

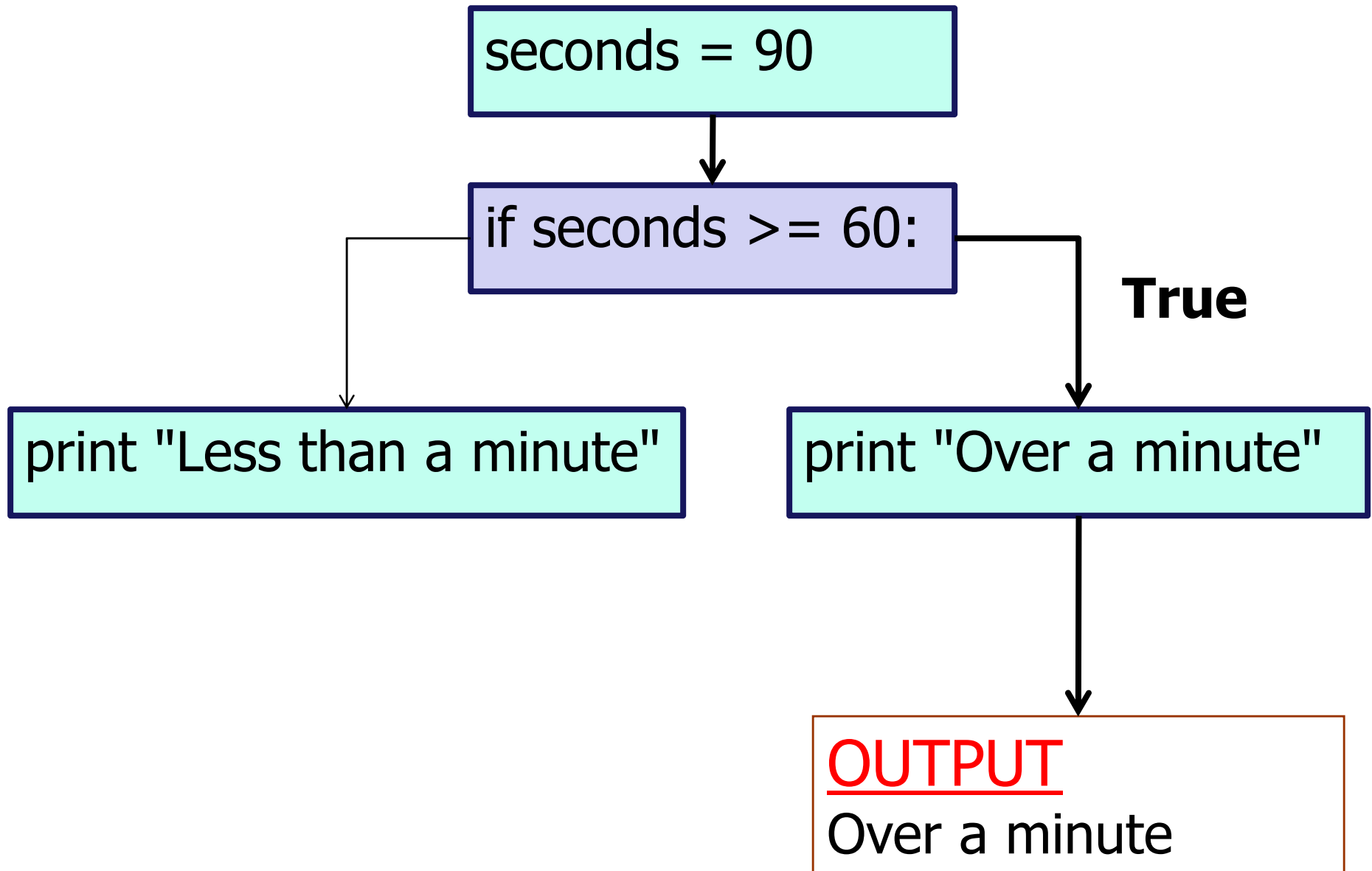
If-Else Statements

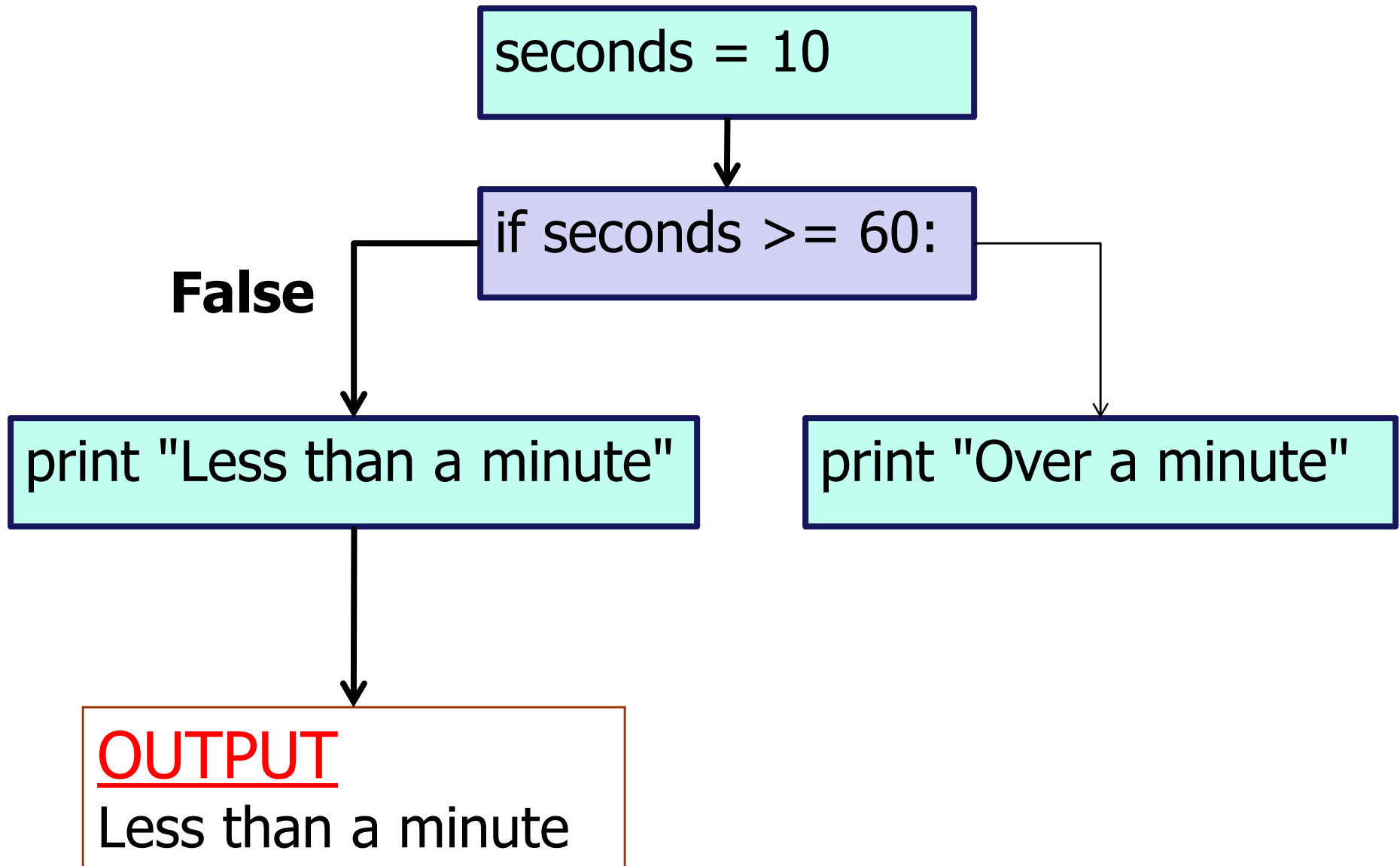
If-Else statements allow you to execute different code if the statement is false

```
seconds = 10
if seconds >= 60:
    print "Over a minute"
else:
    print "Less than a minute"
```

OUTPUT

Less than a minute





**open
ifelse.py**

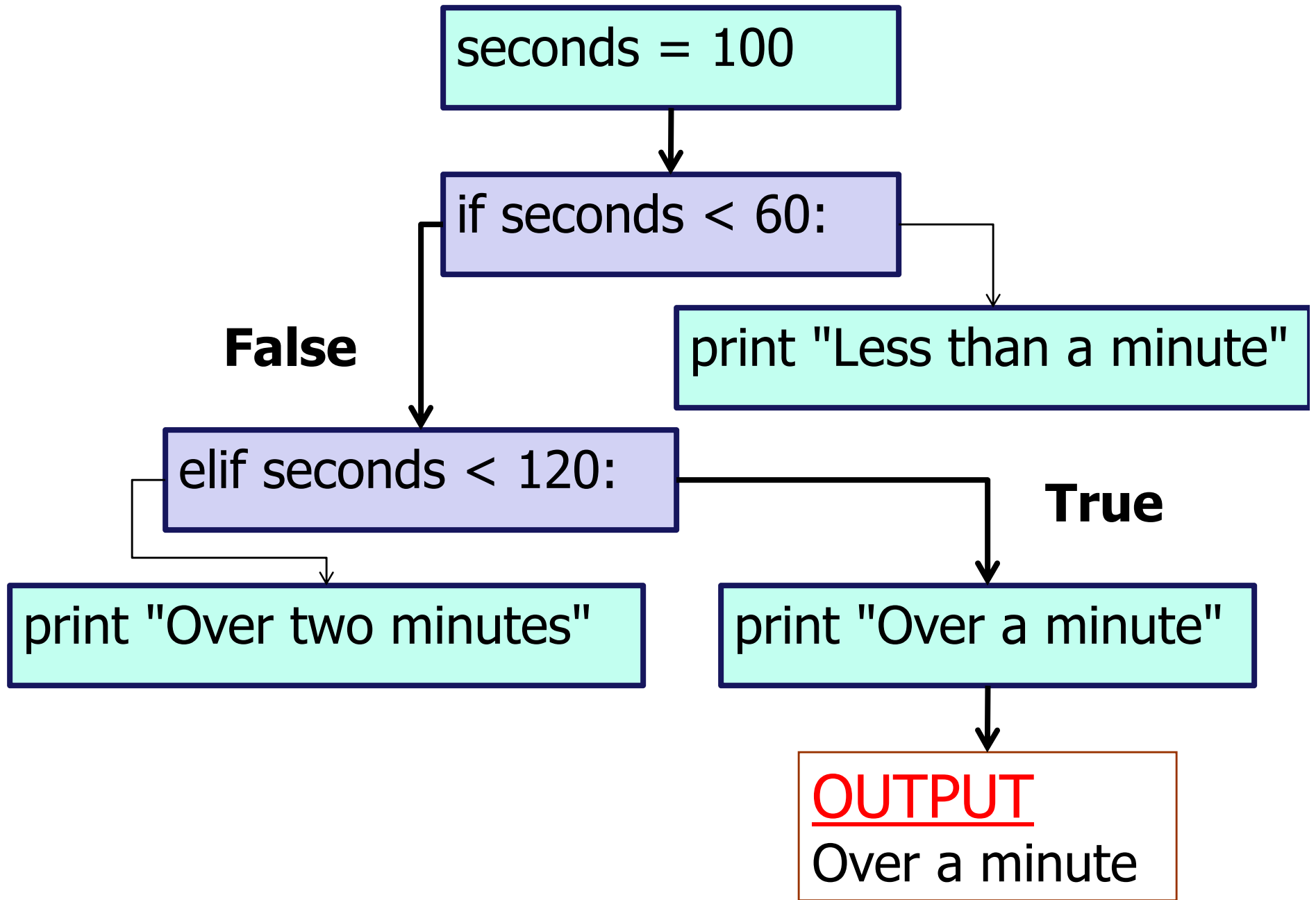
If-elif Statements

If-elif statements allows you to check multiple conditions to determine which code to execute

```
seconds = 100
if seconds < 60:
    print "Less than a minute"
elif seconds < 120:
    print "Over a minute"
else:
    print "Over two minutes"
```

OUTPUT

Over a minute



**open
ifelif.py**

Logical Operators

Logical operators allows you to check multiple conditions in the same if statement

```
x = 50
y = 100
if x < 400 and y < 300:
    print "Upper right"
elif x < 400 and y > 300:
    print "Upper left"
else:
    print "Bottom"
```

OUTPUT
Upper right

Relational frequently used operators

Name	Use
x or y	Either x or y must be true
x and y	Both x and y must be true
not x	If x is true, it becomes false If x is false, it becomes true

**open
logical.py**

Nested ifs

If statements can be put inside other if statements

```
grade = 97
if grade >= 70:
    print "Passing"
    if grade >= 90:
        print "A"
else:
    print "Failing"
```

OUTPUT

Passing

A

Nested ifs

The else part of and if-else statement MUST be in line with it's corresponding if

```
grade = 75
if grade >= 70:
    print "Passing"
    if grade >= 90:
        print "A"
    else:
        print "Failing"
```

OUTPUT

Passing

Failing

**open
nestedifs.py**

Nested ifs

The code in the lab runner files use nested ifs and elifs to check which key is pressed

```
if event.type==QUIT or (event.type==KEYUP and
    event.key==K_ESCAPE):
    sys.exit()
```

```
elif event.type==KEYUP:
    if event.key==K_UP:
        keys = "Up Key"
    elif event.key==K_DOWN:
        keys = "Down Key"
    elif event.key==K_LEFT:
        keys = "Left Key"
    elif event.key==K_RIGHT:
        keys = "Right Key"
```

open
nestedpygame.py

Start work

on IIS Labs