



Control flow : if statements

Ruth Anderson

UW CSE 160

Spring 2018

Making decisions



- How do we compute absolute value?

Absolute value of 5 is

Absolute value of 0 is

Absolute value of -22 is

If *the value is negative*, negate it.

Otherwise, use the original value.

Absolute value solution

If *the value is negative*, negate it.

Otherwise, use the original value.

```
val = -10  
  
# calculate absolute value of val  
if val < 0:  
    result = -val  
else:  
    result = val  
  
print result
```

Condition must be a Boolean expression

Indentation is significant

else is not required

In this example, **result** will always be assigned a value.

Absolute value solution

If *the value is negative*, negate it.

Otherwise, use the original value.

```
val = -10

# calculate absolute value of val
if val < 0:
    result = -val
else:
    result = val

print result
```

Another approach
that does the same thing
without using **result**:

```
val = -10

if val < 0:
    print -val
else:
    print val
```

In this example, **result** will always be assigned a value.

Absolute value solution

As with loops, a sequence of statements could be used in place of a single statement:

```
val = -10

# calculate absolute value of val
if val < 0:
    result = -val
    print "val is negative!"
    print "I had to do extra work!"
else:
    result = val
    print "val is positive"
print result
```

Absolute value solution with zero

```
val = 0

# calculate absolute value of val
if val < 0:
    print "val is negative"
    print val
    result = -val
elif val == 0:
    print "val is zero"
    print val
    result = val
else:
    print "val is positive"
    print val
    result = val

print result
```

Another absolute value solution

What happens here?

```
val = 5

# calculate absolute value of val
if val < 0:
    result = -val
    print "val is negative!"
else:
    for i in range(val):
        print "val is positive!"
    result = val
print result
```

Another if

It is not required that anything happens...

```
val = -10

if val < 0:
    print "negative value!"
```

What happens when val = 5?

The if body can be any statements

```
# height is in km
if height > 100:
    print "space"
else:
```

then
clause

Execution gets here only
if "height > 100" is false

```
# height is in km
if height > 500:
    print "space"
elif height > 100:
```

```
    if height > 50:
        print "mesosphere"
    else:
```

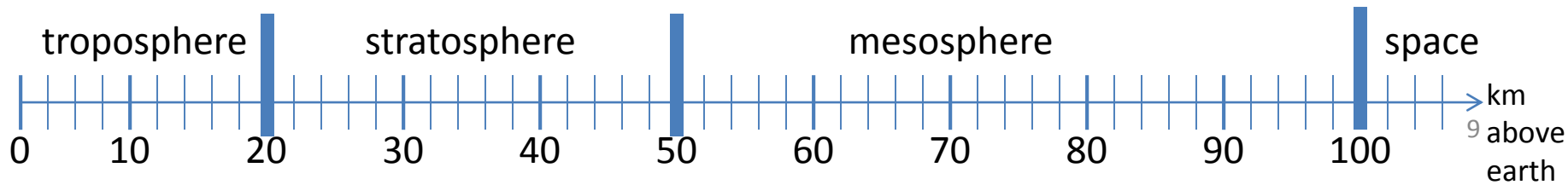
else
clause

Execution gets here only
if "height > 100" is false
AND "height > 50" is true

```
        print "mesosphere"
    elif height > 20:
        print "stratosphere"
    else:
        print "troposphere"
```

```
        if height > 20:
            print "stratosphere"
        else:
            print "troposphere"
```

```
        print "troposphere"
```



Version 1

```
# height is in km
```

```
if height > 100:
```

```
    print "space"
```

Execution gets here only if "height <= 100" is true

```
else:
```

```
    if height > 50:
```

```
        print "mesosphere"
```

Execution gets here only if "height <= 100" is true AND "height > 50" is true

```
    else:
```

```
        if height > 20:
```

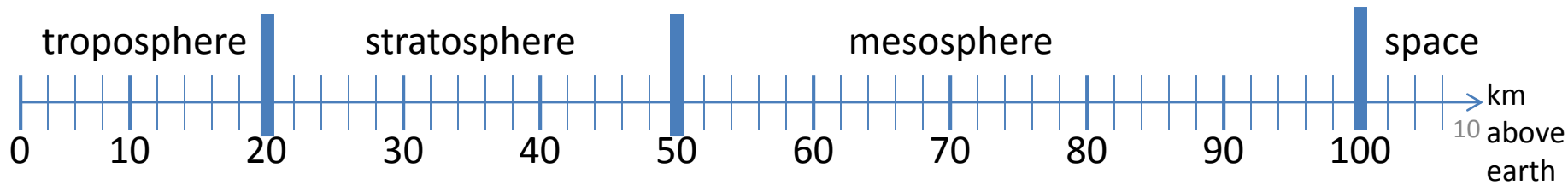
```
            print "stratosphere"
```

```
        else:
```

```
            print "troposphere"
```

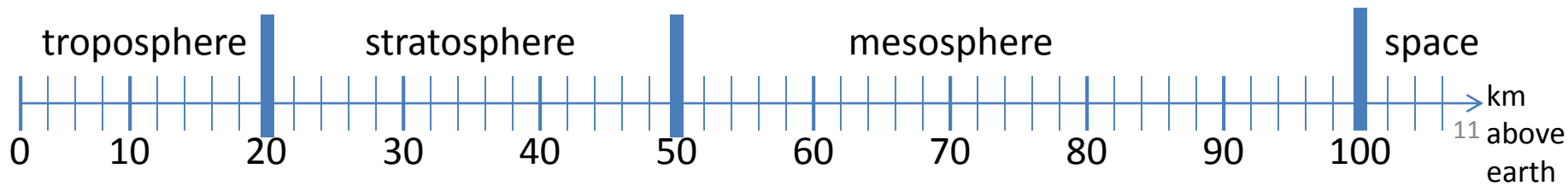
else clause

then clause



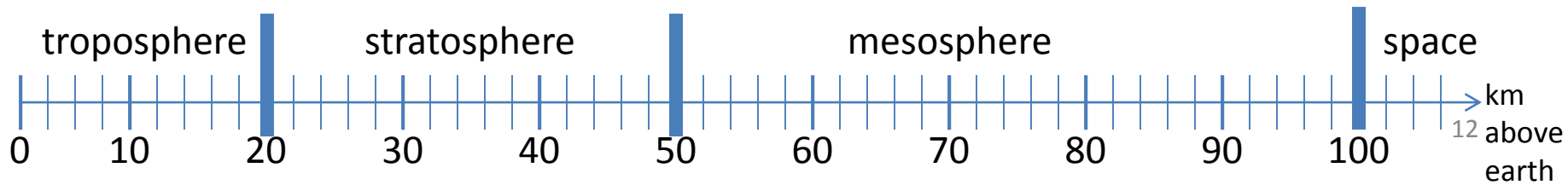
Version 1

```
# height is in km
if height > 100:
    print "space"
else:
    if height > 50:
        print "mesosphere"
    else:
        if height > 20:
            print "stratosphere"
        else:
            print "troposphere"
```



Version 2

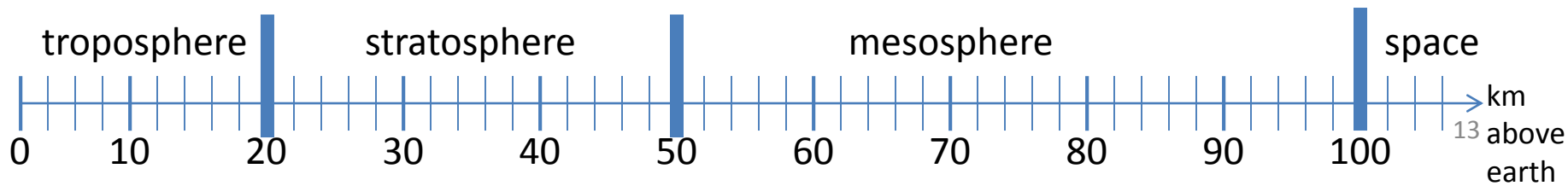
```
if height > 50:  
    if height > 100:  
        print "space"  
    else:  
        print "mesosphere"  
else:  
    if height > 20:  
        print "stratosphere"  
    else:  
        print "troposphere"
```



Version 3 (Best)

```
if height > 100:  
    print "space"  
elif height > 50:  
    print "mesosphere"  
elif height > 20:  
    print "stratosphere"  
else:  
    print "troposphere"
```

ONE of the print statements is guaranteed to execute:
whichever condition it encounters first that is true

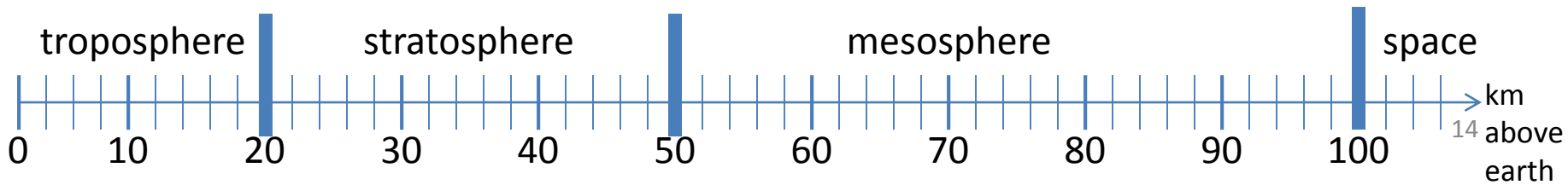


Order Matters

```
# version 3
if height > 100:
    print "space"
elif height > 50:
    print "mesosphere"
elif height > 20:
    print "stratosphere"
else:
    print "troposphere"
```

```
# broken version 3
if height > 20:
    print "stratosphere"
elif height > 50:
    print "mesosphere"
elif height > 100:
    print "space"
else:
    print "troposphere"
```

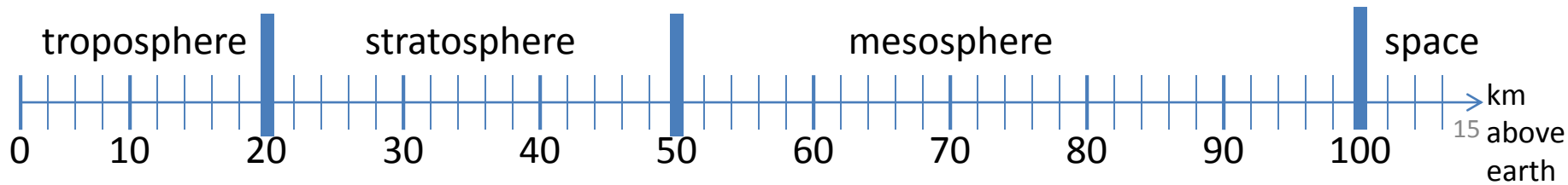
Try height = 72 on both versions, what happens?



Incomplete Version 3

```
# incomplete version 3
if height > 100:
    print "space"
elif height > 50:
    print "mesosphere"
elif height > 20:
    print "stratosphere"
```

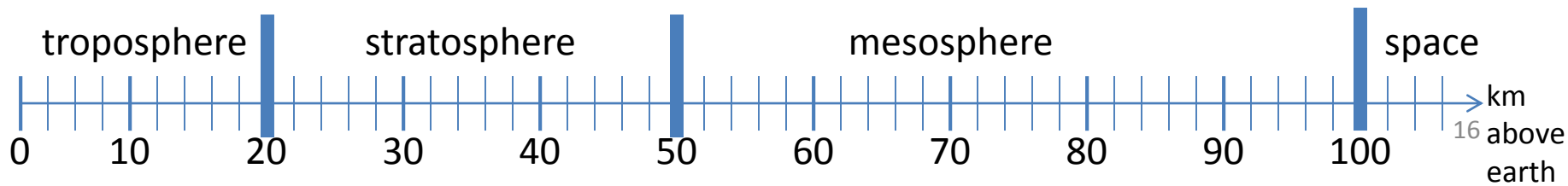
In this case it is possible that nothing is printed at all, when?



What Happens Here?

```
# height is in km
if height > 100:
    print "space"
if height > 50:
    print "mesosphere"
if height > 20:
    print "stratosphere"
else:
    print "troposphere"
```

Try height = 72



The then clause *or* the else clause is executed

[See in python tutor](#)

```
speed = 54
limit = 55
if speed <= limit:
    print "Good job!"
else:
    print "You owe $", speed/fine
```

What if we change speed to 64?