Think Python 2e, Chapter 5 Notes

Geoffrey Matthews

September 19, 2022

Three kinds of division

```
>>> 13 / 5
2.6
>>> 13 // 5
2
>>> 13 % 5
3
```

- Integer division is good for getting hours and minutes from minutes.
- Pints and gallons, feet and inches, ...

Boolean expressions

- A new type of object: boolean

```
>>> 5 == 5
True
>>> 5 == 6
False
```

```
>>> type(True)
<class 'bool'>
>>> type(False)
<class 'bool'>
```

Relational operators

Relate two numbers

```
x != y                    # x is not equal to y
x > y                     # x is greater than y
x < y                     # x is less than y
x >= y                    # x is greater than or equal to y
x <= y                    # x is less than or equal to y
```

Logical operators

Operate on one or two booleans

```
x and y          # true when both x and y are true
x or y           # true when x or y or both are true
not x            # true when x is not true
```

Many other things act truthy:

```
>>> 33 and True
True
>>> 0 and True
0
>>> 'hello' and True
True
>>> '' and True
''
```

Conditional execution

```
1 if x > 0:
2     print('x is positive')
```

- The boolean expression is called the **condition**
- The indented part is called the **body**
- Any number of statements can be in the body
- If you want zero statements in the body:

```
1 if x > 0:
2     pass
```

Alternative execution

```
1 if x % 2 == 0:
2     print('x is even')
3 else:
4     print('x is odd')
```

- The two alternatives are called **branches**
- Only one branch will be executed

Chained conditionals

```
1 if x < y:
2     print('x is less than y')
3 elif x > y:
4     print('x is greater than y')
5 else:
6     print('x and y are equal')
```

- The alternatives are called **branches**
- If more than one condition is true only the first such branch is executed.

Nested conditionals

These are equivalent:

```
if x == y:
    print('x and y are equal')
else:
    if x < y:
        print('x is less than y')
    else:
        print('x is greater than y')
```

```
if x < y:
    print('x is less than y')
elif x > y:
    print('x is greater than y')
else:
    print('x and y are equal')
```

- Generally the `elif` form is clearer.

Circular definition

vorpal: an adjective referring to any thing that is vorpal

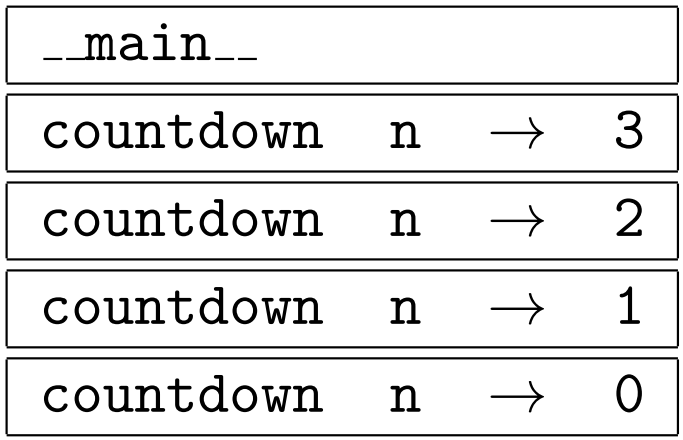
Recursive definition

vorpal:

- a cat
- a dog
- a box containing something that is vorpal

Recursion

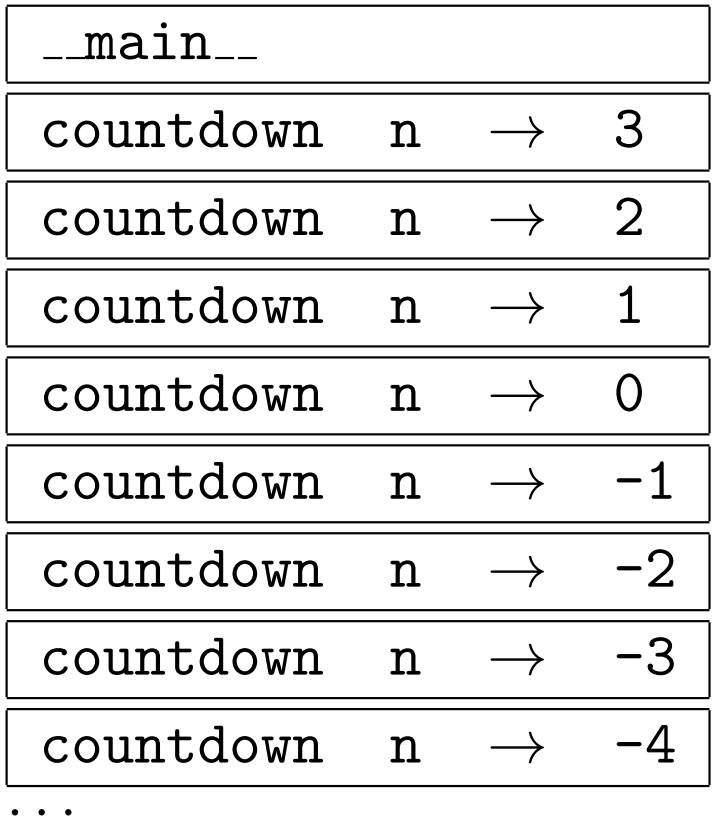
```
1 def countdown(n):
2     if n <= 0:
3         print('Blastoff!')
4     else:
5         print(n)
6         countdown(n-1)
7
8 countdown(3)
```



Stack diagram

Infinite recursion

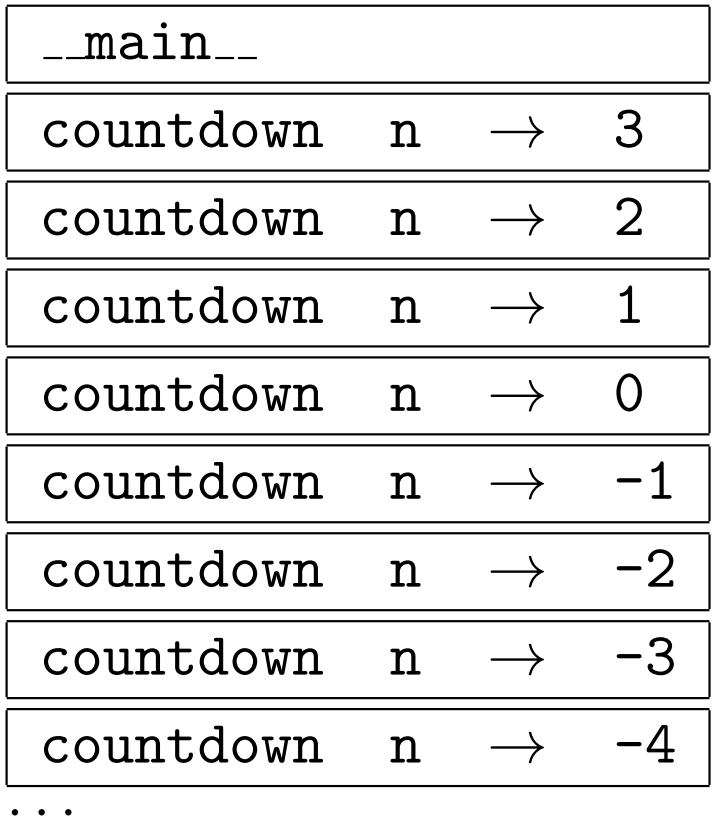
```
1 def countdown(n):
2     print(n)
3     countdown(n-1)
4
5 countdown(3)
```



Stack diagram

Infinite recursion

```
1 def countdown(n):
2     print(n)
3     countdown(n-1)
4
5 countdown(3)
```



Stack diagram

- Must have a base case
- All other branches must get closer to base case

Input

```
>>> text = input()
What are you waiting for?
>>> text
'What are you waiting for?'
```

Input, provide a prompt

```
>>> name = input('What...is your name?\n')
What...is your name?
Arthur, King of the Britons!
>>> name
'Arthur, King of the Britons!'
```

Vocabulary

floor division: An operator, denoted //, that divides two numbers and rounds down (toward negative infinity) to an integer.

modulus operator: An operator, denoted with a percent sign (%), that works on integers and returns the remainder when one number is divided by another.

boolean expression: An expression whose value is either True or False.

relational operator: One of the operators that compares its operands: ==, !=, >, <, >=, and <=.

logical operator: One of the operators that combines boolean expressions: `and`, `or`, and `not`.

Vocabulary

conditional statement: A statement that controls the flow of execution depending on some condition.

condition: The boolean expression in a conditional statement that determines which branch runs.

compound statement: A statement that consists of a header and a body. The header ends with a colon (:). The body is indented relative to the header.

branch: One of the alternative sequences of statements in a conditional statement.

chained conditional: A conditional statement with a series of alternative branches.

nested conditional: A conditional statement that appears in one of the branches of another conditional statement.

Vocabulary

return statement: A statement that causes a function to end immediately and return to the caller.

recursion: The process of calling the function that is currently executing.

base case: A conditional branch in a recursive function that does not make a recursive call.

infinite recursion: A recursion that doesn't have a base case, or never reaches it. Eventually, an infinite recursion causes a runtime error.