Chapter_5_Conditionals_and_recursion

March 6, 2024

The floor division operator, //, divides two numbers and rounds down to an integer.

[1]:
```
minutes = 105
minutes / 60
```

[1]: 1.75

[2]:
```
minutes = 105
hours = minutes // 60
hours
```

[2]: 1

[3]:
```
#To get the remainder, you could subtract off one hour in minutes:
remainder = minutes - hours * 60
remainder
```

[3]: 45

[4]:
```
#An alternative is to use the modulus operator, %, which divides two numbers␣
 ↪and returns
#the remainder.
remainder = minutes % 60
remainder
```

[4]: 45

[5]:
```
#If you are using Python 2, division works differently. The division operator, /
 ↪, performs
#floor division if both operands are integers, and floating-point division if␣
 ↪either operand is
#a float.
```

[6]:
```
5 == 6
```

[6]: False

[7]:
```
type(True)
```

[7]:
```
bool
```

[8]:
```
not (2 > 0)
```

[8]:
```
False
```

[9]:
```
42 and True
```

[9]:
```
True
```

There is no limit on the number of statements that can appear in the body, but there has to be at least one. Occasionally, it is useful to have a body with no statements (usually as a place keeper for code you haven't written yet). In that case, you can use the pass statement, which does nothing

[10]:
```
x=-1
if x < 0:
    pass # TODO: need to handle negative values!
```

[11]:
```
# if choice == 'a':
#     draw_a()
# elif choice == 'b':
#     draw_b()
# elif choice == 'c':
#     draw_c()
```

Each condition is checked in order. If the first is false, the next is checked, and so on. If one of them is true, the corresponding branch runs and the statement ends. Even if more than one condition is true, only the first true branch runs.

function that calls itself is recursive; the process of executing it is called recursion

If a recursion never reaches a base case, it goes on making recursive calls forever, and the program never terminates. This is known as infinite recursion, and it is generally not a good idea.

In most programming environments, a program with infinite recursion does not really run forever. Python reports an error message when the maximum recursion depth is reached

If you encounter an infinite recursion by accident, review your function to confirm that there is a base case that does not make a recursive call. And if there is a base case, check whether you are guaranteed to reach it.

"input" function returns what the user typed as a string. In Python 2, the same function is called "raw_input".

[ ]:
```

```