

## date\_fashion

```
def date_fashion(you, date):
    if you <= 2 or date <= 2:
        return 0

    if you >= 8 or date >= 8:
        return 2

    return 1
```

## squirrel\_play

```
def squirrel_play(temp, is_summer):
    if is_summer:
        if 60 <= temp <= 100:
            return True
        else:
            return False
    else:
        if 60 <= temp <= 90:
            return True
        else:
            return False
```

Here's another way, returning the Boolean result directly rather than including an if / else

```
def squirrel_play(temp, is_summer):
    if is_summer:
        return 60 <= temp <= 100
    else:
        return 60 <= temp <= 90
```

## caught\_speeding

```
def caught_speeding(speed, is_birthday):
    # The variable "birthday_add" is used to reduce the level of
    # "nesting" of the if statements
    if is_birthday:
        birthday_add = 5
    else:
        birthday_add = 0

    if speed <= 60 + birthday_add:
        return 0
    elif speed <= 80 + birthday_add:
        return 1
    else:
        return 2
```



## sorta\_sum

```
def sorta_sum(a, b):
    sum = a + b
    if 10 <= sum <= 19:
        return 20
    else:
        return sum
```

## alarm\_clock

Note that the `== True` part of the first `if` statement could be omitted, and written as just `if vacation`:

```
def alarm_clock(day, vacation):
    if vacation == True:
        if 1 <= day <= 5:
            return "10:00"
        else:
            return "off"
    else:
        if 1 <= day <= 5:
            return "7:00"
        else:
            return "10:00"
```

## in1to10

```
def in1to10(n, outside_mode):
    if outside_mode:
        return n <=1 or n >= 10
    else:
        return 1 <= n <= 10
```

## love6

```
def love6(a, b):
    if a == 6 or b==6 or abs(a-b) == 6 or a+b==6:
        return True
    else:
        return False
```

## near\_ten

```
def near_ten(num):
    return num % 10 <= 2 or num %10 >= 8
```