COL1000: Introduction to Programming

Nuts & Bolts of Python — Conditionals (Cont.) & Loops

Subodh Sharma | Lec 5 | Aug 12

Reminders!

- Do **NOT** share passwords!
- Check https://moodlenew.iitd.ac.in/ General-> Course Homepage -> Schedule
  - Under Lecture Code —> SVS find the `lec4.py` and play with it!
- First **lab test** in Week 5 (29 Aug - 5 Sept), then every other week in your regular lab slot
- Ungraded extra exercise available under the section “Self Study Exercises”
- **Help sessions in CSC lab from 5-6 pm on all working days! (Use only if you need it)**

Example: Find the Max, Mid, Min of 3 Ints

- Live Programming exercise!
- How do we know it is correct for **all** possible inputs
  - Use the idea of **invariants** in the form of **preconditions** and **postconditions**
  - **Formal Specification:**
    - **Precondition:** An invariant that holds true at the start of the program
    - **Postcondition:** An invariant that holds true after the execution of the program

Short Circuiting in Conditionals

- The logical operators `and` and `or` can be **short-circuited**
  - That is, stop the moment final result is determined
    - This implies only partial evaluation - one could avoid **unnecessary computation!**
  - `A and B # if A is falsy, then B is not evaluated`
  - `A or B # if A is truthy, then B is not evaluated`
  - **Also for safety** — `user is not None and user.is_authenticated`
    - If `user is None` then the 2nd (unsafe) expr will not be evaluated
  - Short-circuiting also exists for chained conditionals: Eg: `a<b<=c` (≡ to `a<b` and `b<=c`)

Advanced Conditional Concepts:

- Assignment in condition
  - Use := instead of =
- Floats equivalence:
  - Use math.isclose instead of ==

```
if(n := len(data))>b:
```

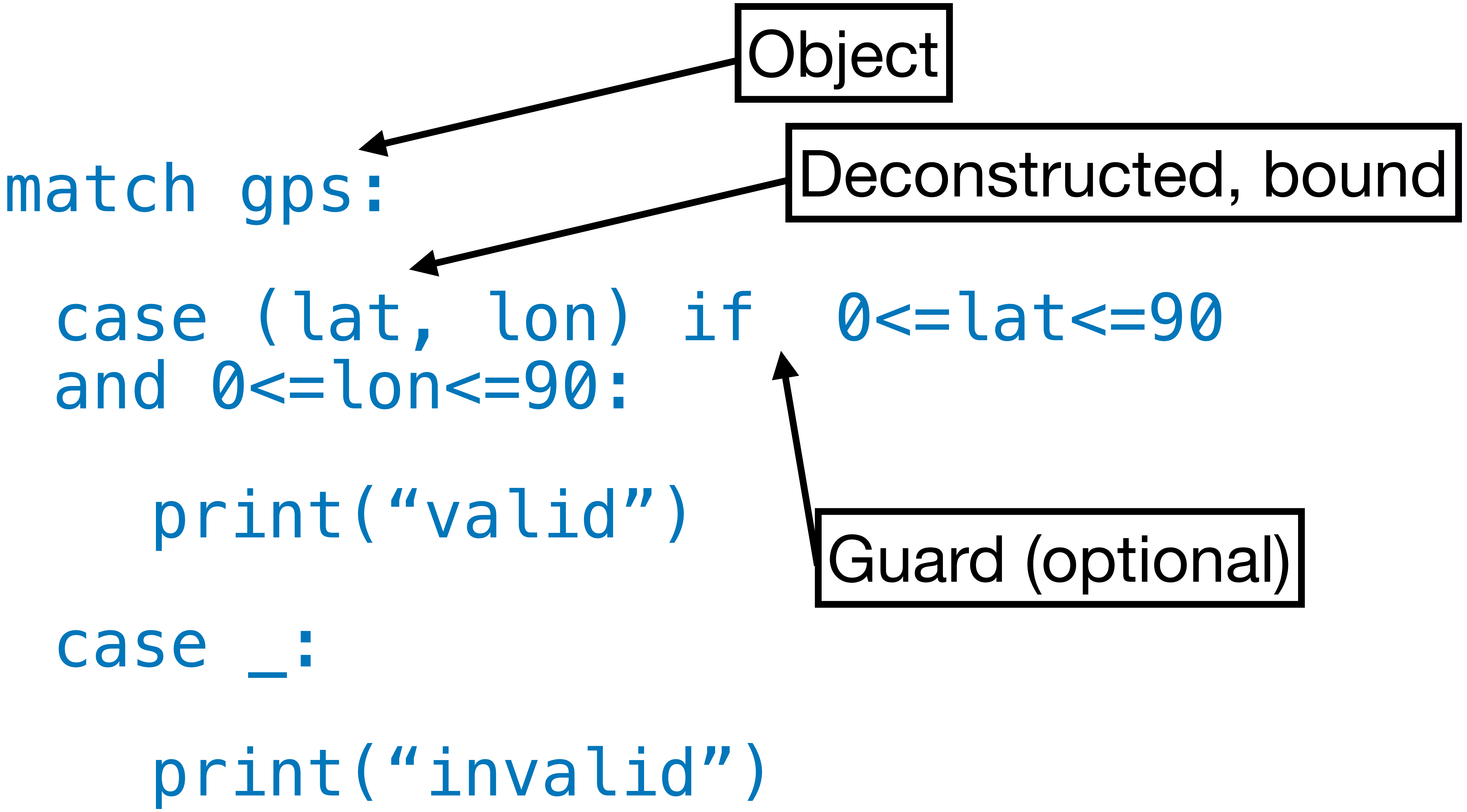
Try:

```
a,b,c = 0.1, 0.2, 0.3
print( (a+b) == c)
import math
print(math.isclose(a+b,c,rel_tol=1e-6)
```

Much like a header in C

Advanced Conditional Concepts:

- **Structural Pattern Matching**
  - Deconstructs a structure in to parts, binds to vars and optionally allows addition of guards/checks
  - **More later in the course**



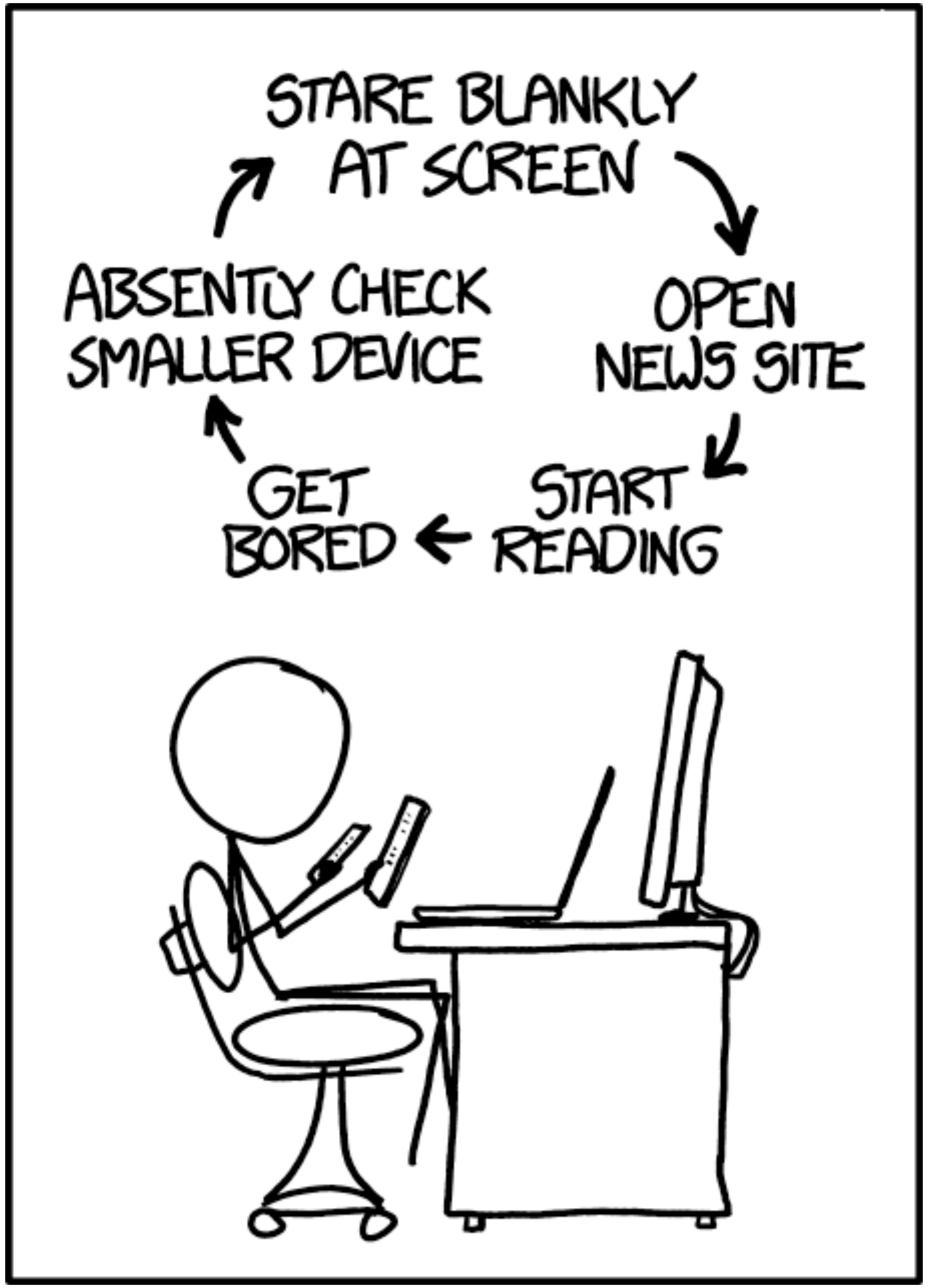
```
match gps:

  case (lat, lon) if  0<=lat<=90
  and 0<=lon<=90:

    print(“valid”)

  case _:

    print(“invalid”)
```

Object

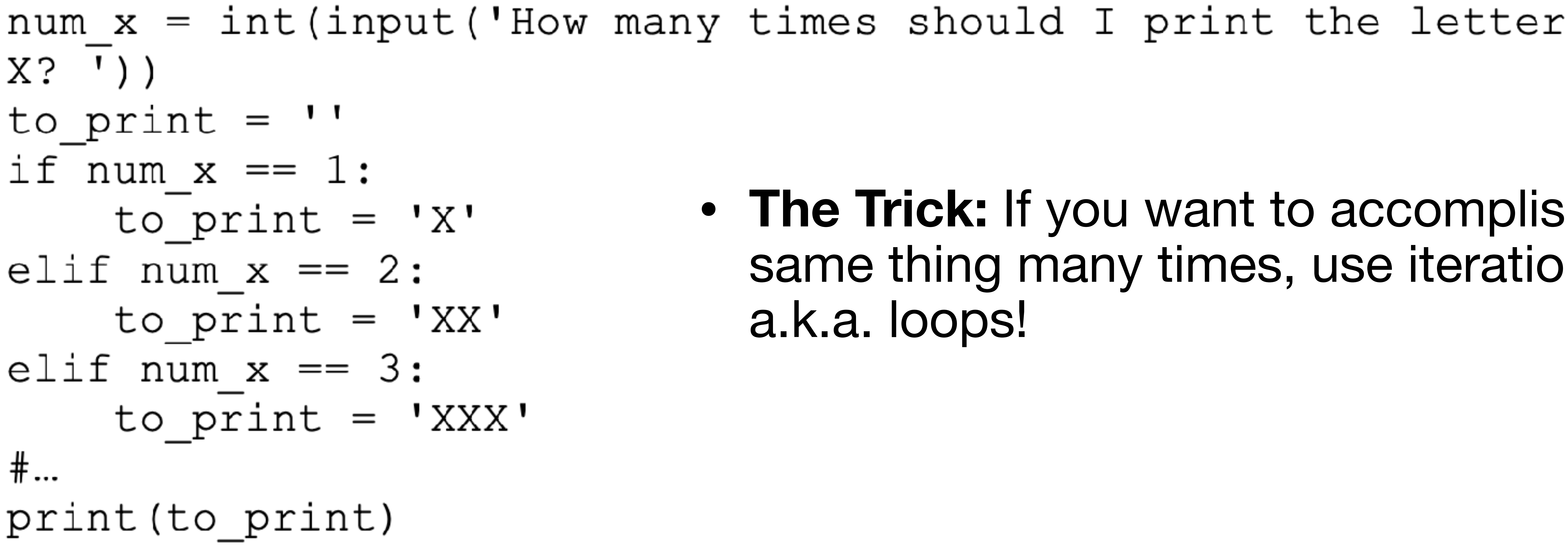
Deconstructed, bound

Guard (optional)

LOOPS:


STARE BLANKLY AT SCREEN
OPEN NEWS SITE
START READING
GET BORED
ABSENTLY CHECK SMALLER DEVICE

Loops: The necessity!



```
num_x = int(input('How many times should I print the letter
X? '))
to_print = ''
if num_x == 1:
    to_print = 'X'
elif num_x == 2:
    to_print = 'XX'
elif num_x == 3:
    to_print = 'XXX'
#...
print(to_print)
```

- **The Trick:** If you want to accomplish the same thing many times, use iteration, a.k.a. loops!

Loops: While

- Semantics:
    - Iterate so long as the condition is true
    - Exit the loop when condition becomes false

```
while (<C>):
  <loop-body>
else: # optional
   …
```

```
while(True):
  print("…")
```

Don't do this!