COL1000: Introduction to Programming

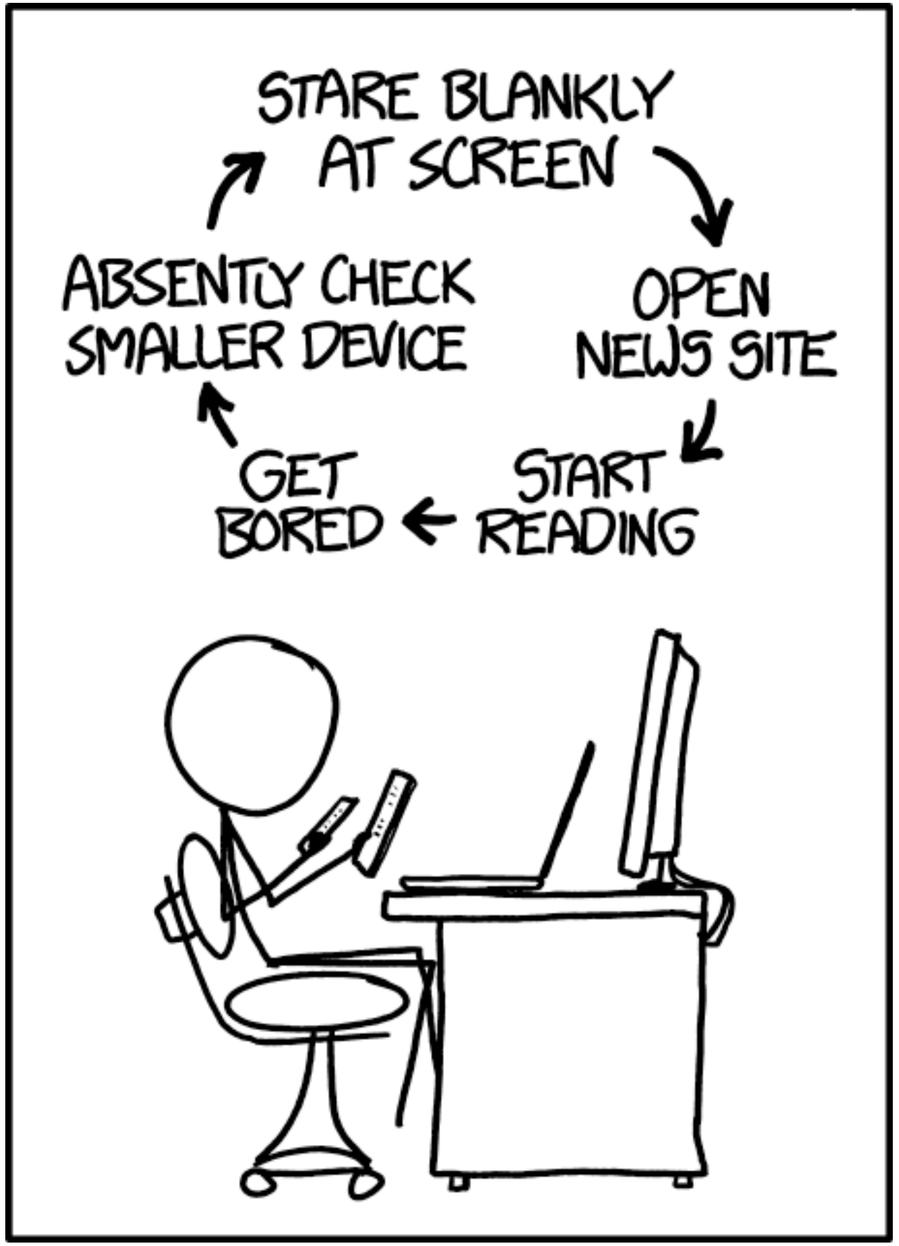
Nuts & Bolts of Python — And more loops …

Subodh Sharma | Lec 7 | Aug 14

Reminders!

- Use lab time to get lecture-doubts cleared
- Check https://moodlenew.iitd.ac.in/ General-> Course Homepage -> Schedule
  - Under Lecture Code —> SVS find the `lec6.py` and play with it!
- **Help sessions in CSC lab from 5-6 pm on all working days! (Use only if you need it)**
- **Talk to me or send me a personal email re:course feedback!**

LOOPS:


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

Loops: While (RECAP)

- **Semantics**:
  - Iterate so long as the condition is true
    - Execute the body **sequentially**
    - Control returns back to condition evaluation
  - Exit the loop when condition becomes false
- Perfect setting: when you don't know how many times you need to repeat something

Syntax

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

Don't do this!

```
while(True):
  print("…")
```

Loops: While

- Postcondition:
  - len(to_print) == num_x
  - Where to assert this postcondition?

```
num_x = int( input('How many times should I print the letter X?'))
#precondition: True

to_print = ''

# Specification: concatenate X to to_print num_x times
while(len(to_print) == num_x):
    to_print += 'X'

# Postcondition: (len(to_print) == num_x) and to_print == 'XX..X'
print(to_print)
```

Loops with Conditions: Program Development

- **Live coding —** begin with postcondition and walk back
-

```
"""
For an input list of integers, print
all the even entries
"""

strLst = input("Enter numbers separated by space:").split()
```

Loops with Conditions: Program Development

```
strLst = input("Enter numbers separated by space:").split()

# map -- higher order function -- returns a map object
numLst = map(int, strLst)

print(type(numLst))

numLst = list(numLst)
print(type(numLst))

idx = 0

resLst = []
#Precondition: len(numLst) >= 0, forall i: numLst[i] is int
while idx < len(numLst):
    if numLst[idx] % 2 == 0:
        resLst.append(numLst[idx])
        print(f"Found even:{numLst[idx]}")
    idx += 1
# Postcondition: idx == len(numLst) and
# Postcondition: forall i: resLst[i]%2 == 0
print(f"All even nums: {resLst}")
```

Applies function `int` to every element of `strLst`

Converts a map object to a list object

Loop Control Variable

Why < and why not <= in the Loop Condition?

Loops: Control Statement

- **break:** terminate the **enclosing** loop;
  - Works for `while` as well as `for` loops
  - E.g.:

```
cnt = 1
while cnt <= 10:
    if cnt == 3:
        break  # terminates the loop even if the loop condition is true!
    print(cnt)
    cnt += 1

print(f"Loop ended early:{cnt}")
```

Loops: Control Statement

- **continue:** skips the current iteration and goes to the next iteration of the **enclosing** loop;
  - Works for `while` as well as `for` loops
- **Live Programming**

— Loop5, Loop6

```
cnt = 1
while cnt <= 10:
    if cnt == 3:
        continue  # skips printing 3
    print(cnt)
    cnt += 1
```