CMPSC 100 SPRING 2021

If we're going to start another week,
we might as well do it while we're here

WELCOME & COURSE INFORMATION

- For **Group B** viewers: reminder that there's no class on Wednesday
  - Welcome **Group B** viewers
- The assignment for the week is posted to #assignments
  - Recall that you need to click the link to accept and clone it
- Quiz is posted to course schedule
- First round of grades is on the course Canvas

Essentially a boolean

BOOLEAN DATA TYPE

- Booleans track values of True and False (capitalization matters)
- These are not strings; they are actual values
- Booleans are used to help make decisions
  - This is especially true of their use in something called a condition

IF YOU'RE AFRAID OF THE DARK...

```
# if it's on
light_switch = True

# if it's off
light_switch = False
```

IF STATMENTS

```
if light_switch == True:
    print("Light's on!")
```

(Equivalent to asking "is this True?")

```
if light_switch:
    print("Light's on!")
```

(Notice the indent - it's important)

IF STATEMENTS

General form:

```
if CONDITION:
    # do something
```

IF STATEMENTS

```
if light_switch:
        # killing the planet, one set of dinosaur remains at a time
        print(“Light’s on!”)
else:
        # oh, you’re a regular Captain Planet, now
        print(“Light’s off!”)
```

IF STATEMENTS

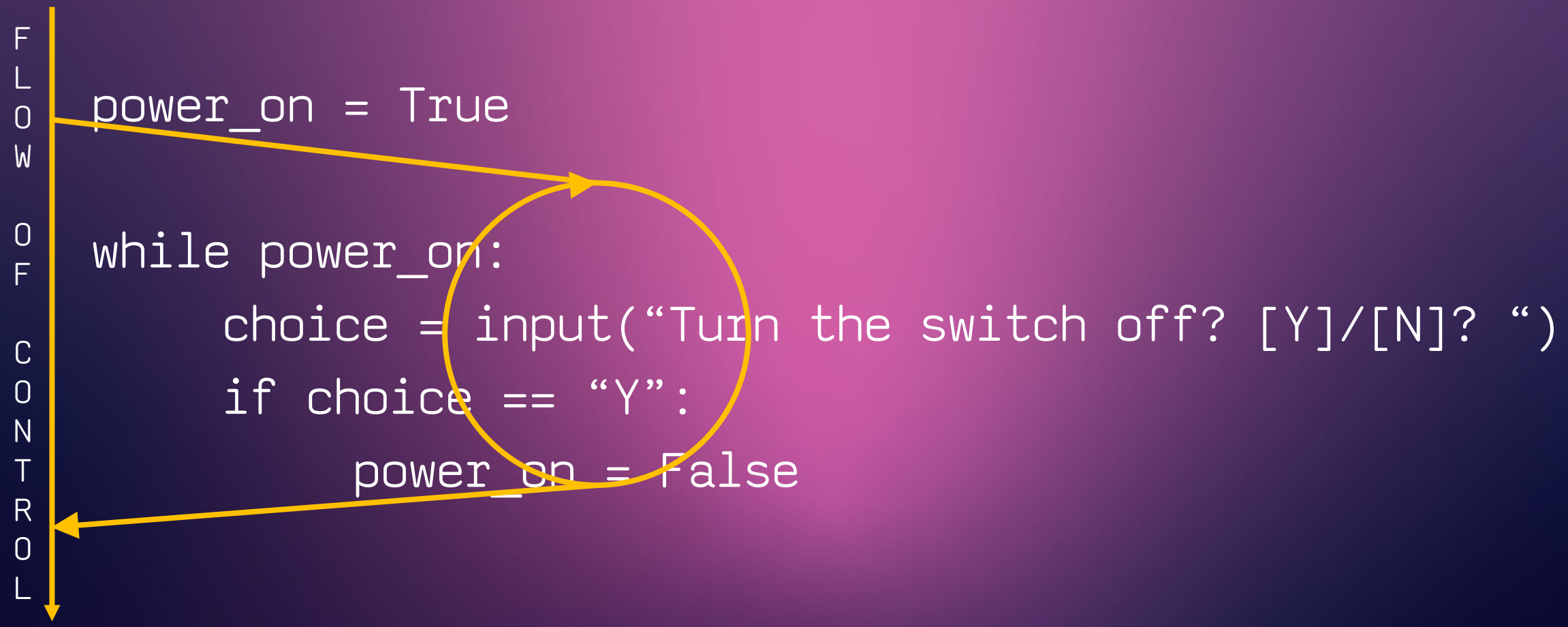
```
if light_switch:
        # if light_switch True
elif not light_switch:
        # if light_switch False
        print(“Light’s off!”)
else:
        # any other conceivable value
        print(“Must not have power.”)
```

WHILE WE'RE HERE...

```
power_on = True

while power_on:
    choice = input("Turn the switch off? [Y]/[N]? ")
    if choice == "Y":
        power_on = False
```

Again, notice the indent:
it demonstrates what
"belongs" a while or if
"block"


FLOW OF CONTROL
power_on = True
while power_on:
choice = input(“Turn the switch off? [Y]/[N]? “)
if choice == “Y”:
power_on = False

while

Model behaviors -- operations to conduct while a condition is true

Executes all “member” statements until a condition is no longer true

Conditions can be simple (while light_switch == True) or complex (while light_switch == True and power == on)

Can be used to “count” by setting up a “sentinel variable”:

```
t = 10

While t > 0:
    print(t)
    t -= 1
```