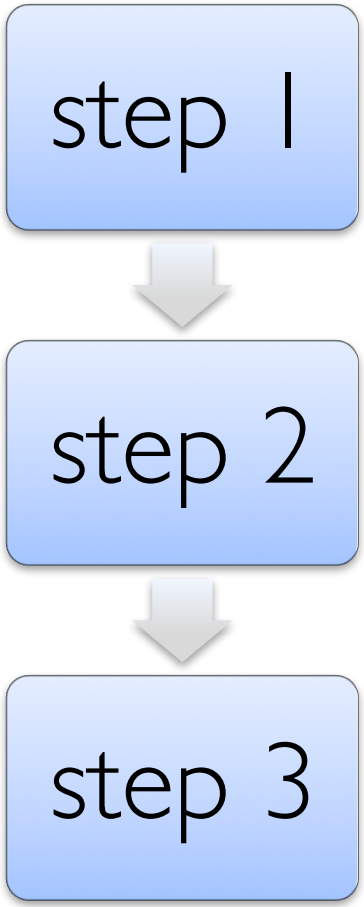
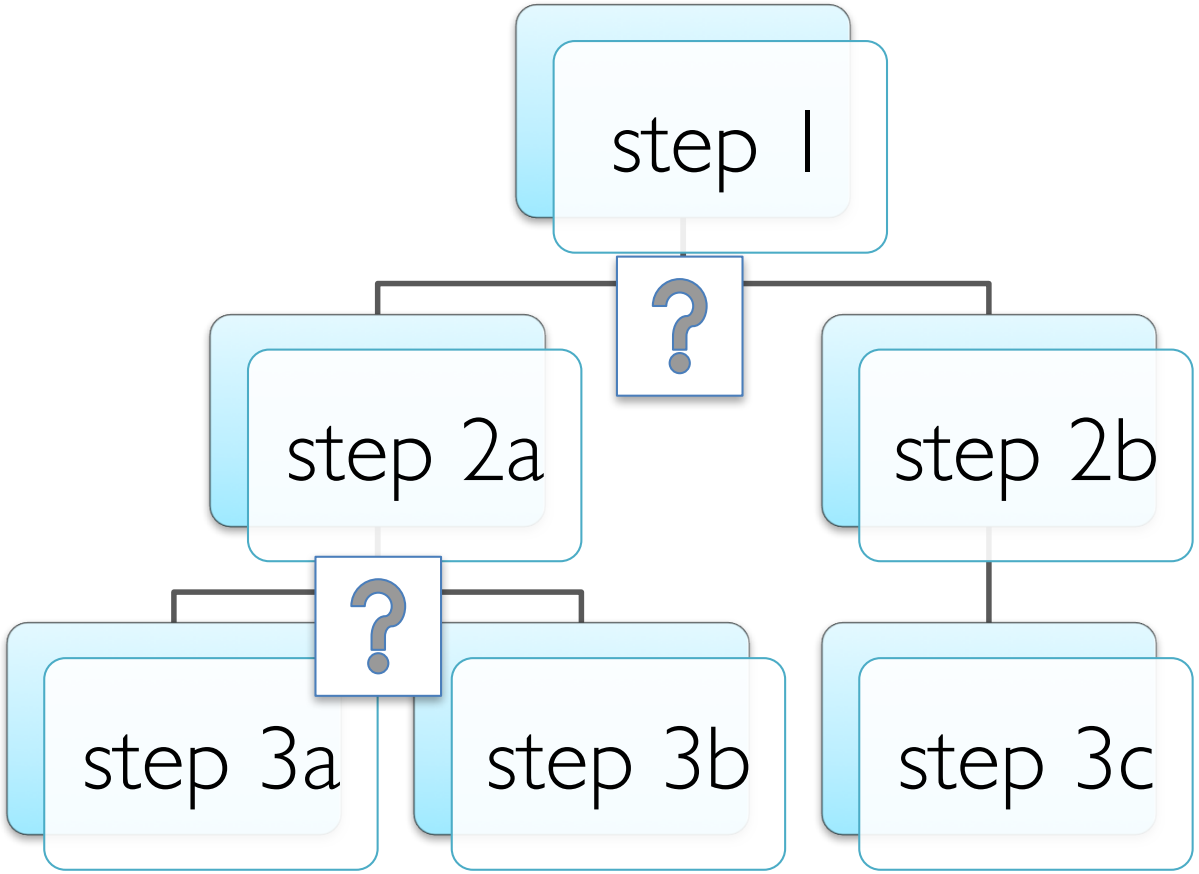
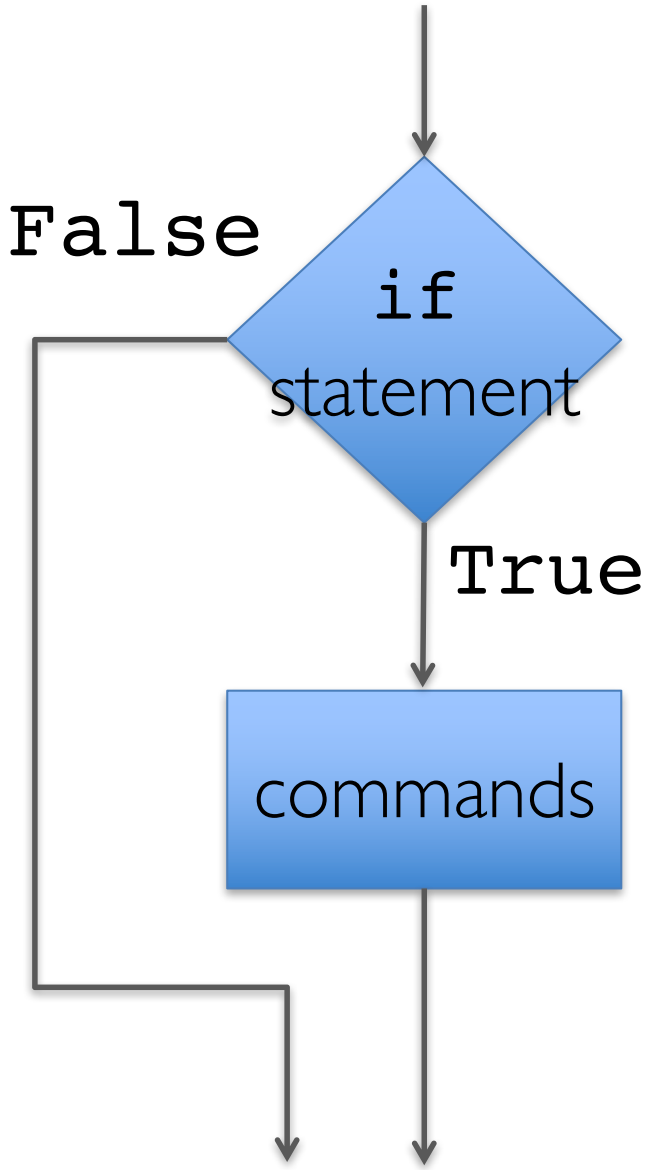
Conditional statements in programming





= *if* [this is true] *then* [do this] *or else* [do this]

Python: `if` statements



syntax:

```
…[previous commands]
if [condition] :
    [command]
    [command]
    [command]
…[subsequent commands]
```

- indentations within "`if`" statements required

Python: example of `if`

replace a number by its absolute value,
if necessary:

```
num= -1
if num < 0 :
    print('You gave me a negative number!')
    num = abs(num)
print(sqrt(num))
```

this part executed only in some circumstances

this part always executed

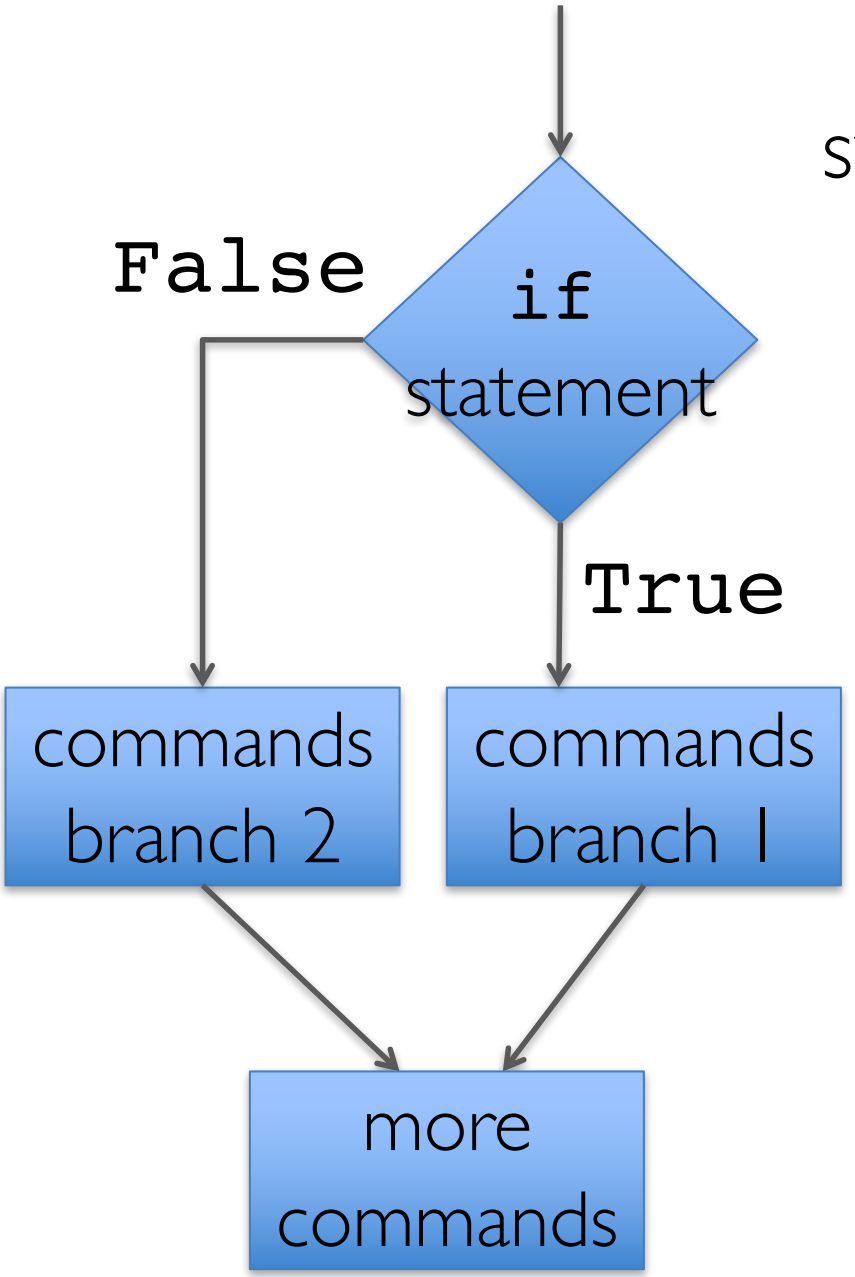
Python: practice with `if`

- write a script that prompts the user for an hour of the day, and then responds with the value of the next hour (`Lec7.ipynb`)
- break it down into algorithmic steps:
  1) prompt user and input value for hour
  2) add 1 to hour to get next hour
  3) check if hour is 12, and if so, change next hour to 1
  4) respond with value
- now re-write, using a function for steps 2 & 3

Another conditional situation

- aircraft onboard-computer asks pilot what the destination is, and gives response about whether or not it's ok to go
- can we do this with an "`if`" statement?

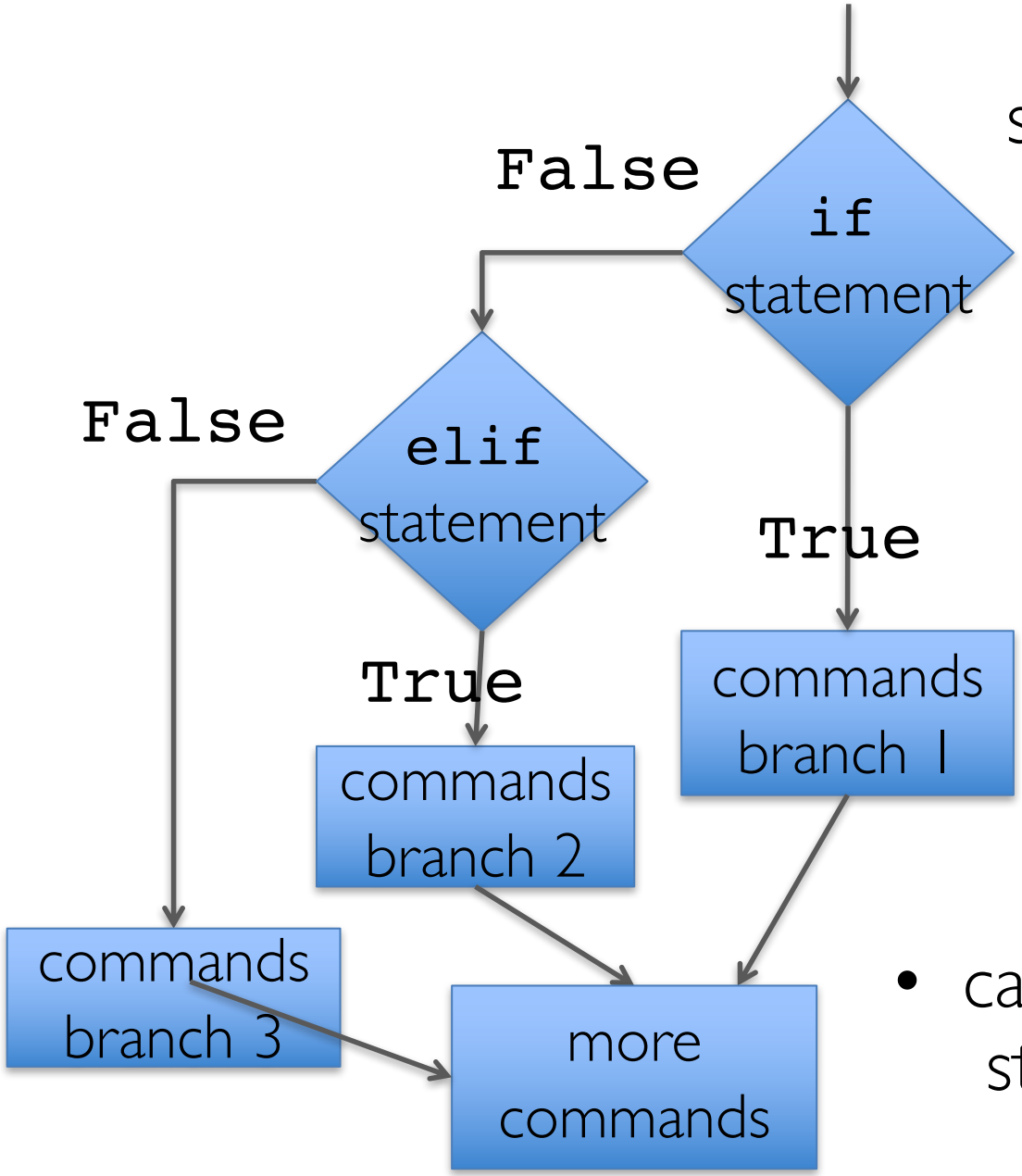
Python: `if-else` statements



syntax:

```
…[previous commands]
if [condition] :
    [command]
    [command]
else :
    [command]
    [command]
…[subsequent commands]
```

Python: `if-elif` statements



syntax:

```
…[prev. commands]
if [condition] :
    [command]
elif :
    [command]
else :
    [command]
…[next commands]
```

- can use multiple "`elif`" statements

Python: other conditionals

- conditional assignment (ternary operator) :
  - `a = val1 if [condition] else val2`
  - `x = -x if x < 0 else x # abs value!`
- what if **`if`**/**`else`**/**`elif`** needs to be applied to an array of values? (e.g. for plotting!)
- won't work…
- use **`vfunc = vectorize(myfunc)`** or "decorator" **`@vectorize`** before **`def myfunc`**
- or logical indexing for variable assignments:
  - `val[condition] = new_val`
  - `val = where(condition,newval1,newval2)`