

Announcements

- ▶ Homework 1 is due tonight!
 - ▶ Make sure you upload a pdf of your written work called HW1.pdf
 - ▶ Make sure to change the assignment status to DONE in the assignment README.
 - ▶ All changes should be committed to the same master branch
 - ▶ If you have issues, let me know earlier rather than later!
 - ▶ I'm aiming to grade these on Sunday, and I'll let you know when I'm done so you can check for any comments
- ▶ Homework 2 will be posted later today
- ▶ Add/Drop deadline next Monday
- ▶ Polling: `rembold-class.ddns.net`

Some Musings...

My experience has been that people good at following directions tend to be better at giving directions. Programming is all about giving directions. If you make a concerted effort to focus and improve your ability to follow directions, I think you'll be surprised at how it improves your coding!

Review Question

Which of the below expressions could be assigned to `x` and would result in

```
('hi all' * 3) != x
```

evaluating to True?

A) `x = 'hi all' + 'hi all' + 'hi all'`

B) `x = 'hi allhi allhi all'`

C) `x = str(3) + 'hi all'`

D) `x = "hi" + 2 * ' allhi' + " all"`

Solution: `x = str(3) + 'hi all'`

Getting Your Input

- ▶ We already have `print()` for writing things to the terminal
- ▶ How can we give the program some sort of input short of editing the code?
 - ▶ One method is the aptly named `input()` function
 - ▶ Gives you a `prompt` at the terminal to enter something

```
guess = input('Guess any number between 1 and 10: ')
```

- ▶ *Always* returns a `string`, so further conversion may be necessary

Troubleshooting:

- ▶ If your program reports an error when you run it:
 - ▶ Look for the last line in the error message
 - ▶ Frequently will give you some info on what bad thing happened and where in the code it happened
 - ▶ Sometimes the actual mistake is not on that line, but immediately above it
- ▶ If your program reports no error but a test is failing:
 - ▶ Something is flawed in your set of instructions
 - ▶ Walk yourself line by line through your code
 - ▶ Frequently can help to **hand-simulate** the code, writing down values and changing them as you move through your code.
 - ▶ A great source to help you visualize this (and not by hand!) is **PythonTutor!**

If `if` fails

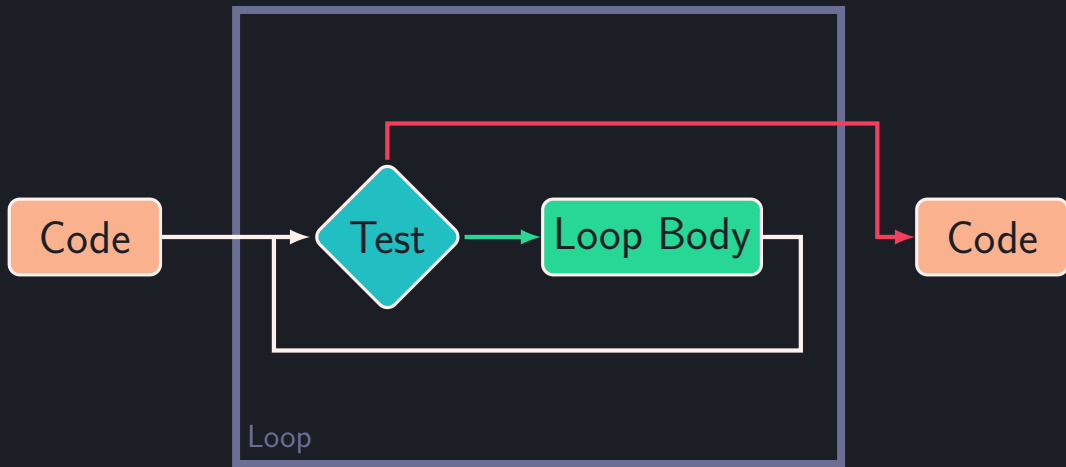


- ▶ Moving to right (for instance) always resets to the same screen
- ▶ Moving any other direction reaches the exit
- ▶ How to code?

Coding the Lost Woods

```
if <move right>:  
    <background = lost woods>  
    if <move right>:  
        <background = lost woods>  
        if <move right>:  
            <background = lost woods>  
            ...  
        else:  
            <background = exit>  
    else:  
        <background = exit>  
else:  
    <background = exit>
```

Groundhog Day



Python Notation

Multiple types of loop in Python, but looking at a **while** loop first:

```
<code>
```

```
while <test>:  
    <loop body>
```

```
<code>
```

Loop Simulations

Let's determine what the below code does by working through it by hand:

```
x = 1
total = 0
while x < 10:
    if x % 2 == 1:
        total = total + x ** 2
    x = x + 1
```

Loop Simulations

Let's determine what the below code does by working through it by hand:

```
x = 1
total = 0
while x < 10:
    if x % 2 == 1:
        total = total + x ** 2
    x = x + 1
```

- ▶ Should have found it adds the squares of all the odd numbers less than 10
- ▶ Result should have been 165

While Loop Tips

- ▶ Make sure the body of your `while` loop is indented and has a colon after the condition
- ▶ `while` loops test some condition, so you may well need to define variable for that condition before the start of the loop.
- ▶ *Make sure your loop will actually terminate at some point!* It can be easy to make infinite loops!

Infinite Loop Advice!

If you find yourself in an infinite loop, pressing `Ctrl+C` in the terminal window will generally interrupt and stop the program!