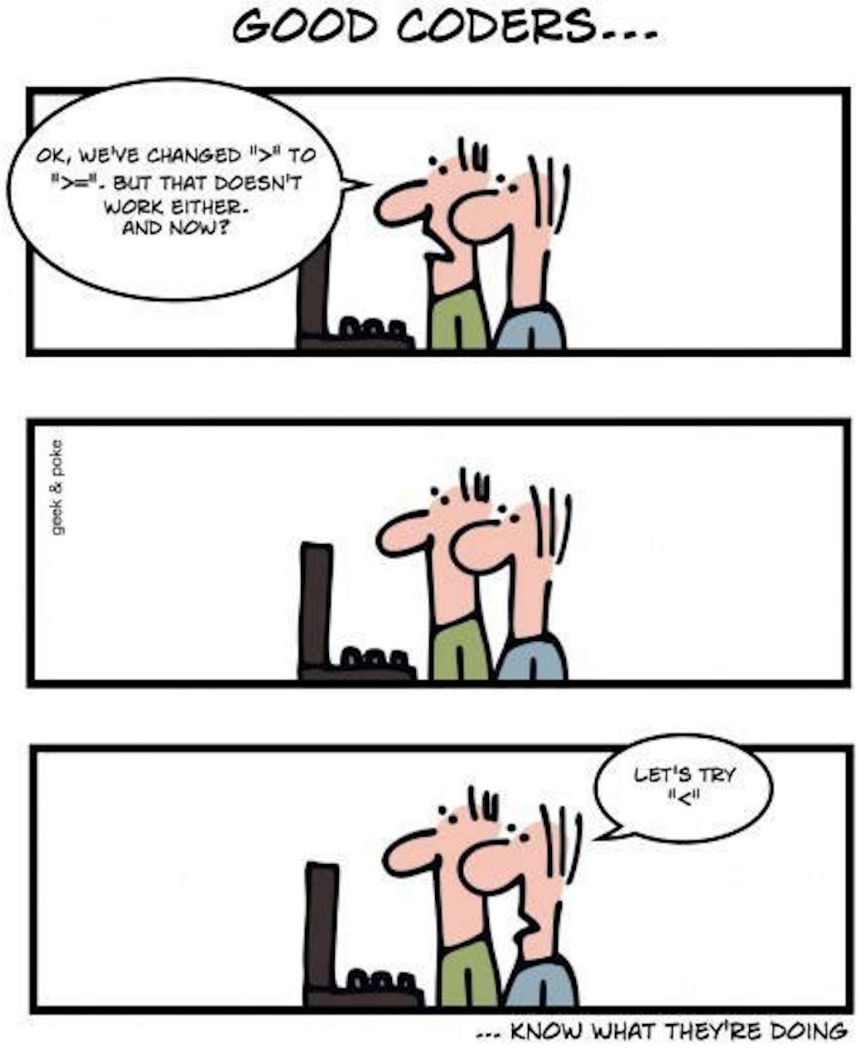
GOOD CODERS...
OK, WE'VE CHANGED ">" TO ">=". BUT THAT DOESN'T WORK EITHER. AND NOW?
geek & poke
LET'S TRY "<"
... KNOW WHAT THEY'RE DOING

CS 110 while-loops

Benjamin Dicken

Other Announcements

- PA 4 Deadline
- Exam 1
  - Individual Exam 1, September 21
  - Group Exam 1, September 23
    - If More than 5 mins late to group, take on your own or with others who are late
  - Review Session: Tues, Sept 20, 5-7pm
  - Study Guide

Activity

Implement Steps A

```
Enter number of steps: 2

##
####
```

```
Enter number of steps: 4

##
####
######
########
```

```
Enter number of steps: 7

##
####
######
########
##########
############
##############
```

Solution for Steps A

```
size = int(input('Enter number of steps: '))
print()
index = 1
while index <= size:
    print('##' * index)
    index += 1
```

Activity

Implement Steps B

```
Enter number of steps: 2

  ##
####
```

```
Enter number of steps: 4

      ##
    ####
  ######
########
```

```
Enter number of steps: 7

            ##
          ####
        ######
      ########
    ##########
  ############
##############
```

Solution for Steps B

```
size = int(input('Enter number of steps: '))
print()
index = 1
while index <= size:
    space = '  ' * (size - index)
    step_row = '##' * index
    print(space + step_row)
    index += 1
```

Activity

Implement Pyramid

```
Enter number of steps: 2

 ##
####
```

```
Enter number of steps: 4

   ##
  ####
 ######
########
```

```
Enter number of steps: 7

      ##
     ####
    ######
   ########
  ##########
 ############
##############
```

Solution for Pyramid

```
size = int(input('Enter number of steps: '))
print()
index = 1
while index <= size:
    space = ' ' * (size - index)
    step_row = '##' * index
    print(space + step_row)
    index += 1
```

Activity

Implement Christmas Tree

```
Enter number of steps: 2

 %%
****
```

```
Enter number of steps: 4

   %%
  ****
 ######
%%%%%%%%
```

```
Enter number of steps: 10

         %%
        ****
       ######
      %%%%%%%%
     **********
    ############
   %%%%%%%%%%%%%%
  ****************
 ##################
%%%%%%%%%%%%%%%%%%%%
```

```
size = int(input('Enter number of steps: '))
print()
index = 1
while index <= size:
    step_row_chars = ''
    if index % 3 == 0:
        step_row_chars = '##'
    elif index % 3 == 1:
        step_row_chars = '%%'
    else:
        step_row_chars = '**'
    space = ' ' * int((size * 2 - index * 2)/2)
    step_row = step_row_chars * index
    print(space + step_row)
    index += 1
```

String indexes

- Each character in a string is located at a particular **index**
- The index is zero-based

String indexes

- Each character in a string is located at a particular **index**
- The index is zero-based

```
name = 'jeremiah'
```

String indexes

- Each character in a string is located at a particular **index**
- The index is zero-based

```
name = 'jeremiah'
```

index	0	1	2	3	4	5	6	7
character	j	e	r	e	m	i	a	h

Checking the value of a character

```python
name = 'jeremiah'
print( name )
print( name[0] )

character_3 = name[3]
print( character_3 )
```

Activity

What will this print?

```python
sentence = 'mailed list'
char_1 = sentence[7]
char_2 = sentence[1]
char_3 = sentence[10]
char_4 = sentence[4]
print(char_1 + char_2 + char_3 + char_3 + char_4)
```

Activity

What will this print?

```
sentence = 'mailed list'
length = len(sentence)
char_1 = sentence[length]
char_2 = sentence[1]
char_3 = sentence[3]
print(char_1 + char_2 + char_3 + char_3)
```

Activity

What will this print?

```
sentence = 'mailed list'
length = len(sentence) - 1
char_1 = sentence[length]
char_2 = sentence[1]
char_3 = sentence[3]
print(char_1 + char_2 + char_3 + char_3)
```

Activity

Add the code to print ‘cores’

```
sentence = 'computer science'
```

Add the code to print ‘cores’

```
sentence = 'computer science'
char_1 = sentence[0]
char_2 = sentence[1]
char_3 = sentence[7]
char_4 = sentence[12]
char_5 = sentence[9]
print(char_1 + char_2 + char_3 + char_4 + char_5)
```

Activity

What will this print? What does it do?

```
digits = input('Type some digits: ') # '2511'
count = 0
i = 0
while i < len(digits):
    value = int(digits[i])
    count += value
    i += 1
print('count:', count)
```

Activity

Password Validation

- Write some code that takes a string password as input, and determines if it is a “valid” password or not
- A valid password is one that:
    - Has at least one upper-case letter (use **`isupper()`**)
    - Is at least 8 characters long
    - Has at least one of these characters:

      `!   ?   ;`
- Print “valid” if valid and “not valid” if not

```python
password = input('Enter a password:\n')

if len(password) < 8:
    print("Invalid password.")
    exit()

has_upper = False
has_special = False
i = 0
while i < len(password):
    if password[i].isupper():
        has_upper = True
    if password[i] == '!' or password[i] == '?' or password[i] == ';':
        has_special = True
    i += 1

if has_upper and has_special:
    print("Valid Password")
else:
    print("Invalid password.")
```

Activity

What will this print?

```
one = 'the lost world'
two = 'the last stride'
i = min(len(one), len(two)) - 1
count = 0
while i >= 0:
    if one[i] == two[i]:
        count += 1
    i -= 1
print('tally:', count)
```