List and String Processing

Thomas Schwarz, SJ

Activities 1

- Write a program that checks (returns True/False) whether a string ends with .edu
    - one solution with endswith
    - one solution using a slice and comparing strings
    - one solution using indices and comparing characters

Activities 1 Solutions

```
def check1(a_string):
    return a_string.endswith('.edu')

def check2(a_string):
    return a_string[-4:] == '.edu'

def check3(a_string):
    return (a_string[-4] == '.' and a_string[-3] == 'e' and
            a_string[-2] == 'd' and a_string[-1] == 'u')
```

Activities 2

- A function counter that counts the number of consonants in a string

Activities 2 Solutions

```
def cons(a_string):
    count = 0
    for letter in a_string:
        if letter.lower() in 'bcdfghjklmnpqrstvwxyz':
            count += 1
    return count
```

Activities

- A function that removes all vowels in a string

Activities 3 Solutions

```
def rem_vol(a_string):
    result = [ ]
    for letter in a_string:
        if letter not in 'aeiouAEIOU':
            result.append(letter)
    return ''.join(result)
```

Formatting method

- Python has two type of special strings:
  - r-strings for raw strings: no escapes
  - f-strings for formatting
  - Using f-strings results in more compact and readable code

f-strings

- f-strings are defined with a pair of quotation marks preceded immediately by an “f” or “F”

```
fstring = f'hello world'
```

- An f-string can contain a variable name surrounded by brackets in its definition
- The bracket is then replaced by the value of the variable

f-strings

- Example:

```
number = 6.35
astring = “hello"
fstring = f"{astring}, the number is {number}"
```

- Variable fstring is then

```
      'hello, the number is 6.35'
```

f-strings

- The expression in brackets inside an f-string gets evaluated at run time.
- For example, we can say

```
f"{2+3*4}"
```

- or

```
astring = “hello"
string = f"{astring.upper()} World"
```

which evaluates to

```
'HELLO World'
```

r-strings

- Because of their similarity with f-strings, we mention r-strings
- An r-string uses the escape character only as an escape character, so there is no escaping at all
  - This is useful for strings containing the backslash such as Windows file names

```
address = r"c:\Windows\System32\system.ini"
```

Hangman - Ahorcado

- A slightly morbid childrens' game
  - Guess a word letter by letter
  - For each wrong letter, a part of a hanged man is drawn

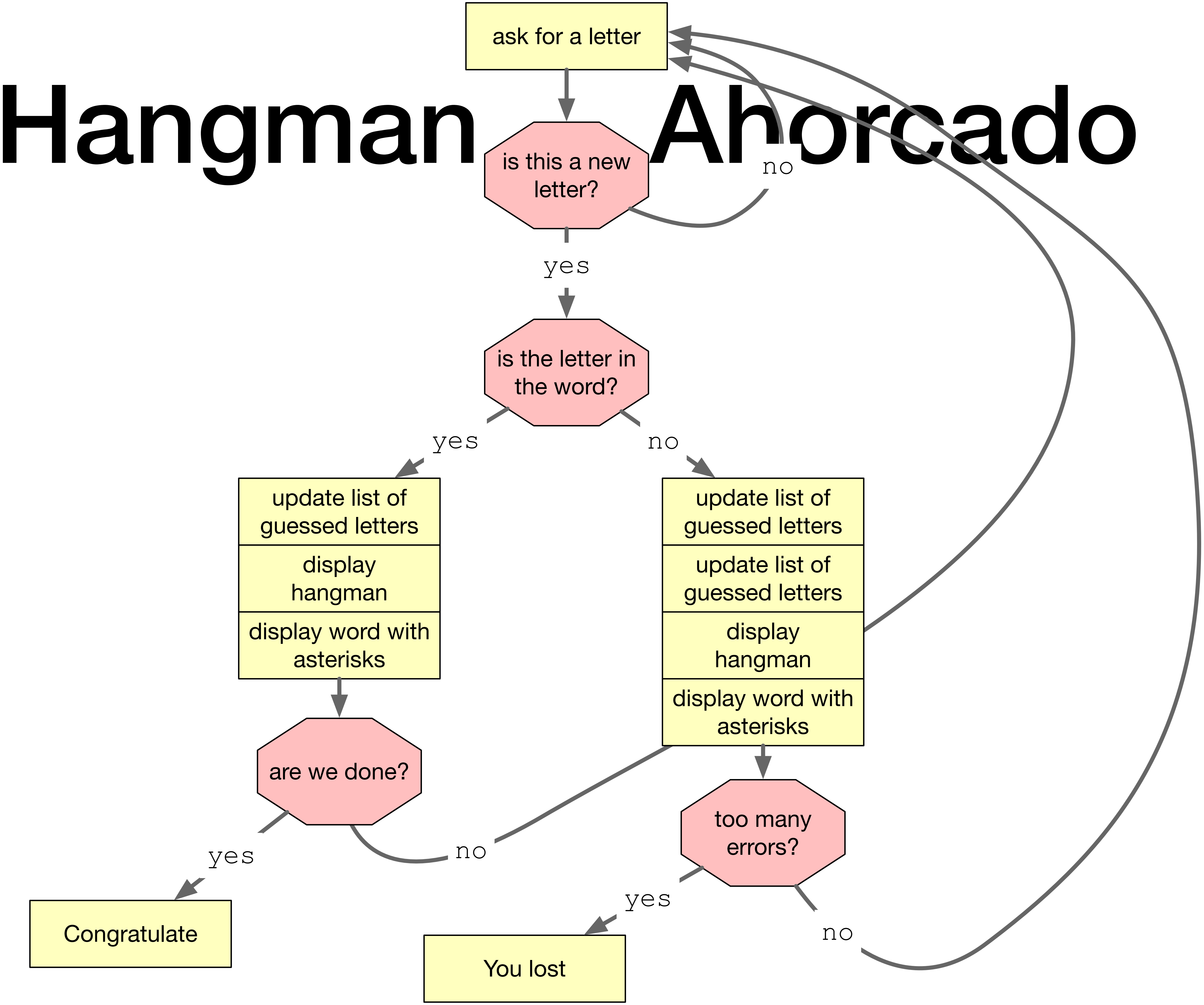
```
Enter a letter j
     +------+
     |      |
     |      o
     |     /|\
     |     / \
     |
     |
     |
     |
     |
     |

you looser you
```

Hangman — Ahorcado

- How to plan a software project?
  - Principal idea: divide tasks into simpler components
  - Make a diagram of program logic:
    - This is apt to change

Hangman Ahorcado



ask for a letter

is this a new letter?

no

yes

is the letter in the word?

yes

no

update list of guessed letters

display hangman

display word with asterisks

update list of guessed letters

update list of guessed letters

display hangman

display word with asterisks

are we done?

yes

no

too many errors?

yes

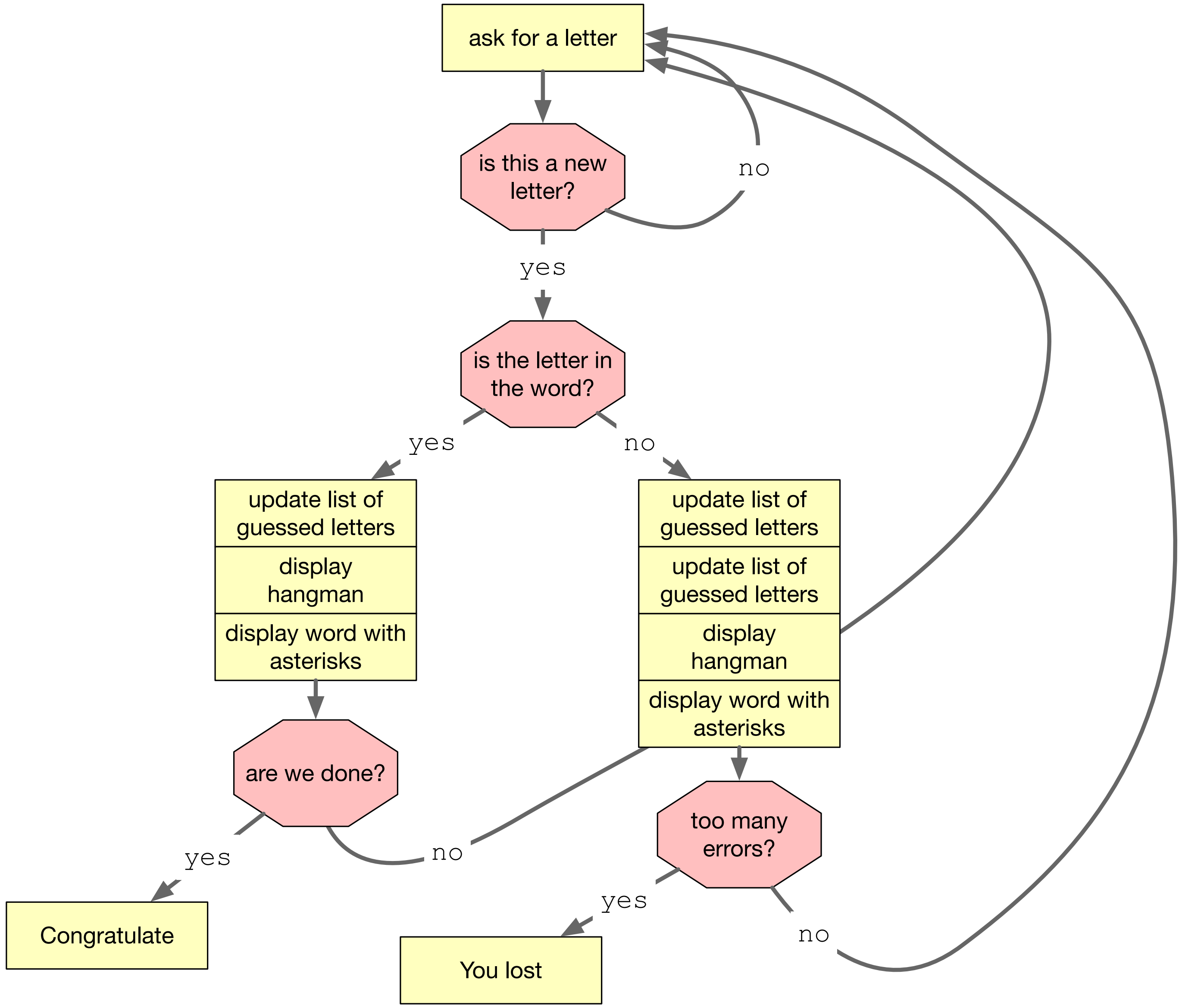
no

Congratulate

You lost

Hangman — Ahorcado

- Observation:
  - We need a list of guessed letters to decide whether this is a letter
  - We need to do more input control
    - User enters digit
    - user enters capital letters
    - …



ask for a letter

is this a new letter?

no

yes

is the letter in the word?

yes

no

update list of guessed letters

display hangman

display word with asterisks

update list of guessed letters

update list of guessed letters

display hangman

display word with asterisks

are we done?

yes

no

Congratulate

too many errors?

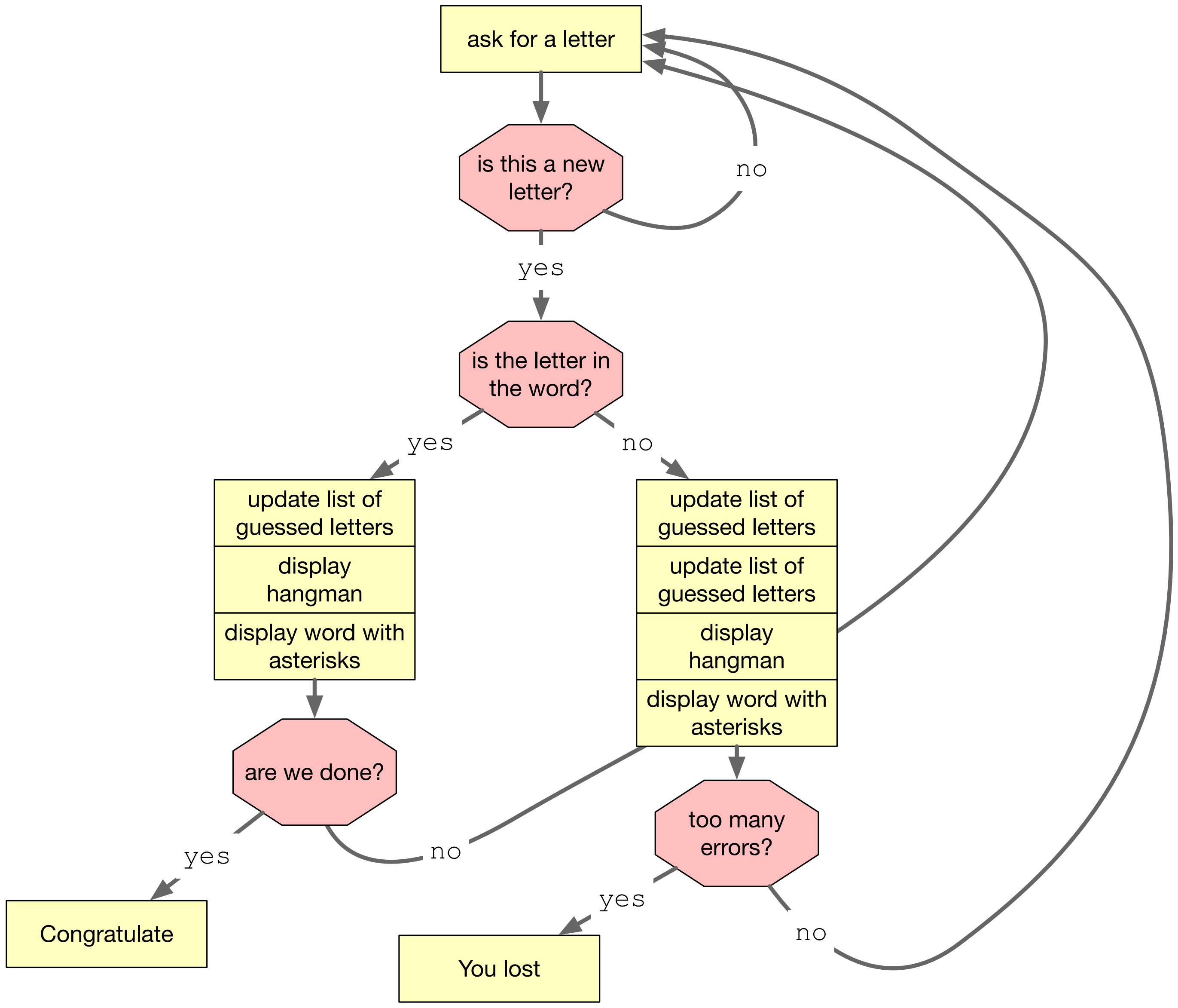
yes

no

You lost

Hangman — Ahorcado

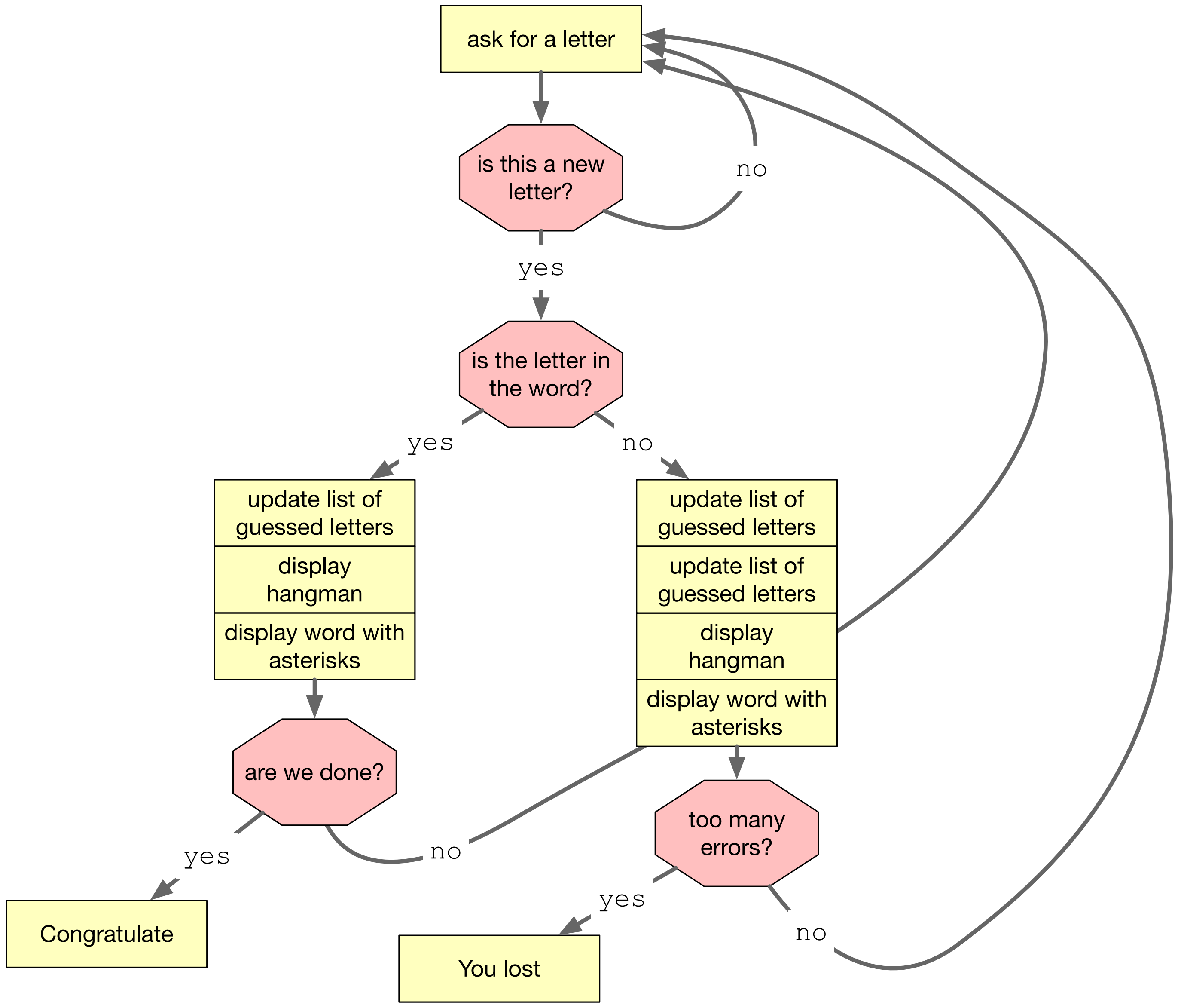
- All of the yellow boxes are candidates for functions
- We can see some common data:
    - The secret word
    - The list of guessed letters
    - The number of bad guesses



ask for a letter
is this a new letter?
no
yes
is the letter in the word?
yes
no
update list of guessed letters
display hangman
display word with asterisks
update list of guessed letters
update list of guessed letters
display hangman
display word with asterisks
are we done?
yes
no
too many errors?
yes
no
Congratulate
You lost

Hangman — Ahorcado

- We can also see that at the heart is a giant loop
- Python-style:
    - Make the loop an infinite loop
    - Break out

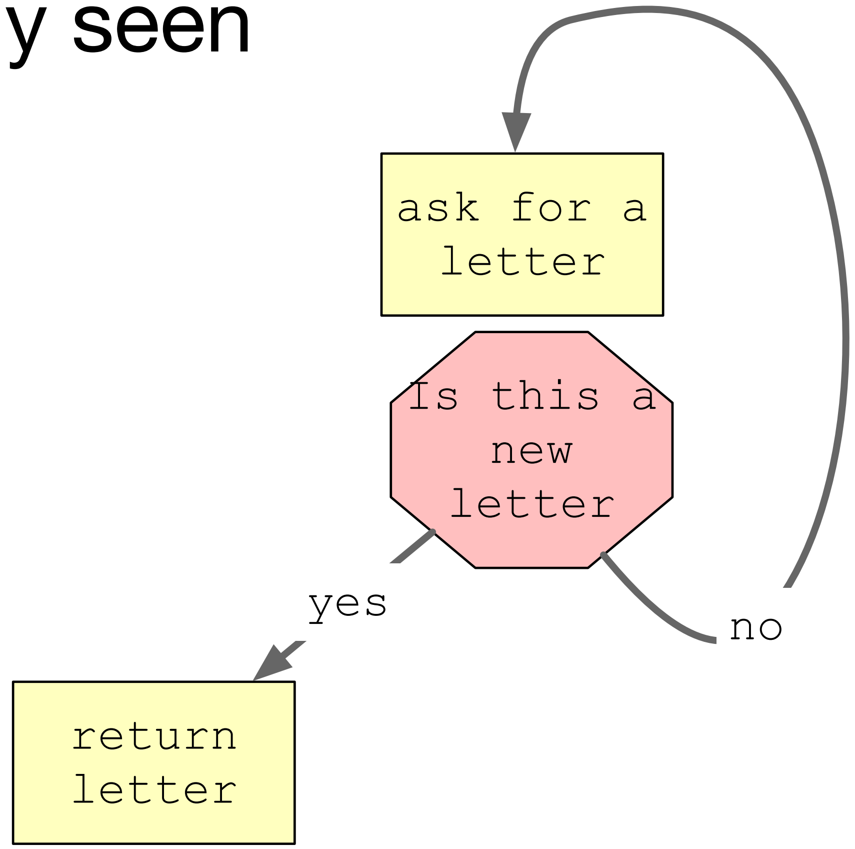

ask for a letter
is this a new letter?
no
yes
is the letter in the word?
yes
no
update list of guessed letters
display hangman
display word with asterisks
update list of guessed letters
update list of guessed letters
display hangman
display word with asterisks
are we done?
yes
no
too many errors?
yes
no
Congratulate
You lost

Hangman — Ahorcado

- A word about diagrams:
  - Programming has become a lot easier over the years
    - So we program more difficult things
    - And focus has shifted
  - Some methods are very data-centric
    - Useful for big data implementation or graphics, e.g.
  - Some methods focus on processing
    - As we just did

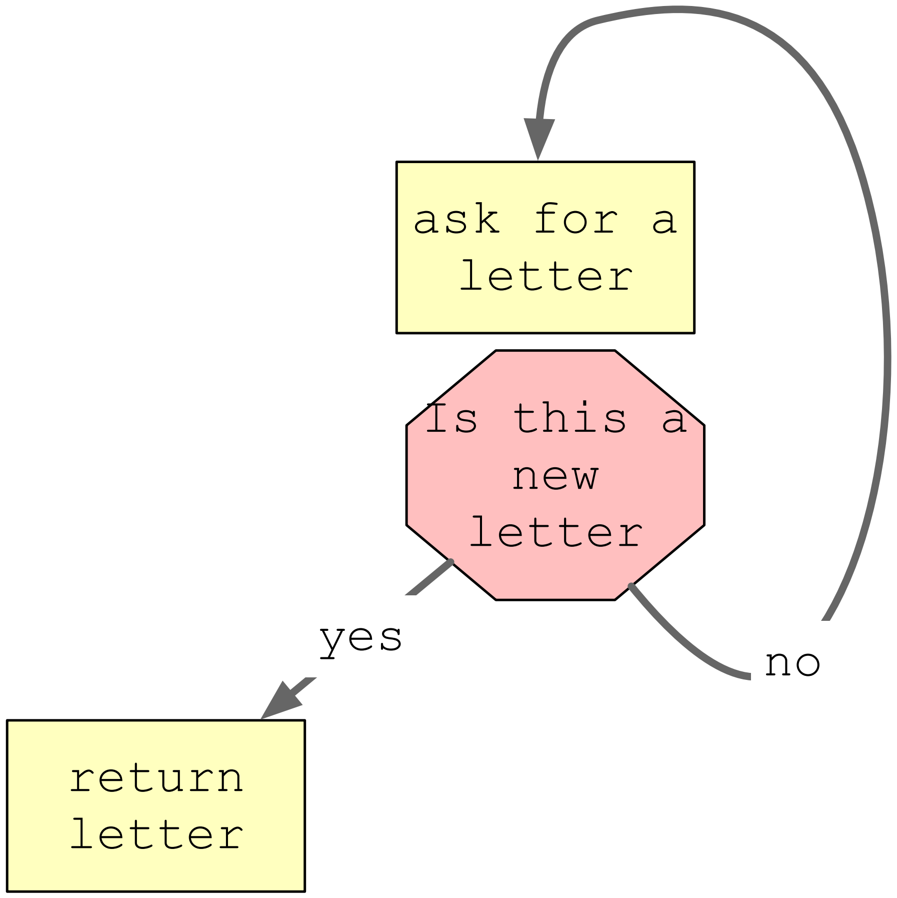
Hangman — Ahorcado

- "Enter a letter" function:
  - Needs one parameter: list of guessed letters
  - Should do error checking (homework / project)
  - Returns a letter not previously seen



Hangman – Ahorcado

```
def get_letter(lol):
    while True:
        x = input('Enter a letter ')
        x = x[0]
        if x in lol:
            print('This letter is already guessed. Try again.')
        else:
            return x
```



Hangman — Ahorcado

- Check whether we are done
    - All the letters in the secret are in the list of letters already guessed (lol)

```
def done(lol, secret):
    for letter in secret:
        if letter not in lol:
            return False
    return True
```

Hangman — Ahorcado

- Print out the hangman: An exercise in ASCII art

```
Enter a letter a
     +------+
     |      |
     |
     |
     |
     |
     |
     |
     |
     |

Good job. The word is *******a
```

Hangman — Ahorcado

```
Enter a letter b
     +-------+
     |       |
     |
     |
     |
     |
     |
     |
     |
     |

Good job. The word is *****b*a
```

Hangman — Ahorcado

```
Enter a letter d
     +------+
     |      |
     |      o
     |
     |
     |
     |
     |
     |
     |

Not quite. The word is c****b*a
```

Hangman — Ahorcado

```
Enter a letter e
     +------+
     |      |
     |      o
     |      |
     |
     |
     |
     |
     |
     |
     |

Not quite. The word is c****b*a
```

Hangman — Ahorcado

```
Enter a letter f
     +------+
     |      |
     |      o
     |     /|
     |
     |
     |
     |
     |
     |
     |

Not quite. The word is c****b*a
```

Hangman — Ahorcado

```
Enter a letter g
     +------+
     |      |
     |      o
     |     /|\
     |
     |
     |
     |
     |
     |
     |

Not quite. The word is c****b*a
```

Hangman — Ahorcado

```
Enter a letter h
     +------+
     |      |
     |      o
     |     /|\
     |     /
     |
     |
     |
     |
     |
     |

Not quite. The word is c****b*a
```

Hangman — Ahorcado

```
Enter a letter i
     +------+
     |      |
     |      o
     |     /|\
     |     /
     |
     |
     |
     |
     |
     |

Good job. The word is c****bia
```

Hangman — Ahorcado

```
Enter a letter j
     +------+
     |      |
     |      o
     |     /|\
     |     / \
     |
     |
     |
     |
     |
     |

you looser you
```

Hangman — Ahorcado

- "printing the hangman"
    - Two possibilities:
        - Draw the same string with slight changes for different number of false guesses
        - Draw different strings (using copy and paste)
    - Can use multi-dimensional strings
        - or use string arithmetic (which becomes unreadable)

```
def print_it(n):
    if n <= 0:
        print(5*' ' +'+------+\n' + 5*' ' +'|      |\n' + 8*(5*' '+'|\n'))
    elif n == 1:
        print(5*' ' +'+------+\n' + 5*' ' +'|      |\n'
              +5*' '+'|      o\n' +7*(5*' '+'|\n'))
    elif n == 2:
        print(5*' ' +'+------+\n' +  5*' ' +'|      |\n'
              +5*' '+'|      o\n' ++5*' '+'|      |\n'
              +7*(5*' '+'|\n'))
    elif n == 3:
        print(5*' ' +'+------+\n' +  5*' ' +'|      |\n'
              +5*' '+'|      o\n' + 5*' '+'|     /|\n'
              +7*(5*' '+'|\n'))
    elif n == 4:
        print(5*' ' +'+------+\n' +  5*' ' +'|      |\n'
              +5*' '+'|      o\n' + 5*' '+'|     /|\\ \n'
              +7*(5*' '+'|\n'))
    elif n == 5:
        print(5*' ' +'+------+\n' +  5*' ' +'|      |\n'
              +5*' '+'|      o\n' + 5*' '+'|     /|\\ \n'
              +5*' '+'|     /  \n'
              +6*(5*' '+'|\n'))
    elif n == 6:
        print(5*' ' +'+------+\n' +  5*' ' +'|      |\n'
              +5*' '+'|      o\n' + 5*' '+'|     /|\\ \n'
              +5*' '+'|     / \ \n'
              +6*(5*' '+'|\n'))
```

Hangman — Ahorcado

- Now we are ready for the game:
  - First, define the data structures

```
def game():
    secret = 'colombia'
    lol = []
    false_guesses = 0
        …
```

Hangman — Ahorcado

- Then start the while loop:

```
def game():
    secret = 'colombia'
    lol = []
    false_guesses = 0
    while True:
       …
```

Hangman — Ahorcado

- First, get the letter and do not forget to update your list of guessed letters (lol)
- We have hidden some logic in get_letter

```
while True:
        x = get_letter(lol)
        lol.append(x)
```

Hangman — Ahorcado

- If the letter is a good guess:
    - Print hangman and word, then check whether we are done

```
if x in secret:
    print_it(false_guesses)
    if done(lol, secret):
        print('You won')
        break
    else:
        print('Good job. The word is', display(secret, lol))
```

Hangman — Ahorcado

- If the letter is bad:
  - update false guesses
  - print hangman
  - decide on whether we lost

```
if x not in secret:
    false_guesses += 1
    print_it(false_guesses)
    if false_guesses >= 6:
        print("you looser you")
        break
    else:
        print('Not quite. The word is', display(secret, lol))
```

Hangman — Ahorcado

- Notice: We could have used return in order to get out of the loop

```
def game():
   secret = 'colombia'
   lol = []
   false_guesses = 0
   while True:
      x = get_letter(lol)
      lol.append(x)
      if x in secret:
         print_it(false_guesses)
         if done(lol, secret):
            print('You won')
            break
        else:
            print('Good job. The word is', display(secret, lol))
        if x not in secret:
            false_guesses += 1
            print_it(false_guesses)
            if false_guesses >= 6:
                print("you looser you")
                break
            else:
                print('Not quite. The word is', display(secret, lol))
```