

MadLibs.

Design Problem 3

if nothing else, write `#cleancode`

Agenda

- Problem Background
- Learning Objectives
- Coding

Background

- MadLibs is a game where you write a story with missing words, then ask someone else to fill in the blanks!
- Since they don't know what your story is about, the result is often wonderful and silly.
- Traditionally MadLibs is done with paper and ink, but we are in the digital age!
- **Goal:** Write a Python program that uses string methods and user input to play MadLibs with the user, and output the created story at the end.

Our school cafeteria has really squirm^y food. Just thinking about it makes my stomach hop. The spaghetti is supersonic and tastes like blob. One day, I swear one of my meatballs started to draw! The turkey tacos are totally flaming and look kind of like old dry ice. My friend Dana actually likes the meatloaf, even though it's rare and lumpy. I call it "mystery meatloaf" and think it's really made out of goomba. My dad said he'd make my lunches, but the first day, he made me a sandwich out of screen and peanut butter! I think I'd rather take my chances with the cafeteria!

Background

- Groom's Name: _____
- Bride's Name: _____
- Noun: _____
- Number: _____
- School: _____
- Place on Campus: _____
- Verb (-ing): _____
- Verb (-ed): _____
- Greeting: _____
- Adjective: _____

Background

- Groom's Name: _____
- Bride's Name: _____
- Noun: _____
- Number: _____
- School: _____
- Place on Campus: _____
- Verb (-ing): _____
- Verb (-ed): _____
- Greeting: _____
- Adjective: _____

HOW THEY MET

WHEN groom's name MET bride's name, IT WAS now AT
number SIGHT. HE SAW HER AT school name HIGH SCHOOL AS SHE WAS
STANDING NEXT TO THE place on campus, verb (ing) TO ANOTHER FRIEND
OF HIS. HE verb (ad) OVER TO SAY greeting, INTRODUCED HIMSELF,
AND ASKED HER NAME. SHE WAS SO adjective! "HI THERE!", SHE SAID.
adverb, MY NAME IS bride's name. SHE HAD SEEN HIM AROUND
AND THOUGHT HE WAS SUPER adjective.

What whimsical stories will you write...

Learning Objectives

- Practice working with strings and using string methods.
- Practice with user input.
- Practice creating custom functions.

THE MAGIC COMPUTERS

Today, every student has a computer small enough to fit into his _____.
_____ NOUN He can solve any math problem by simply pushing the computer's little _____.

Computers can add, multiply, divide, and _____.

VERB (PRESENT TENSE) They can also _____.

VERB (PRESENT TENSE) better than a human. Some computers are _____.

PART OF BODY (PLURAL) Others have a/an _____.

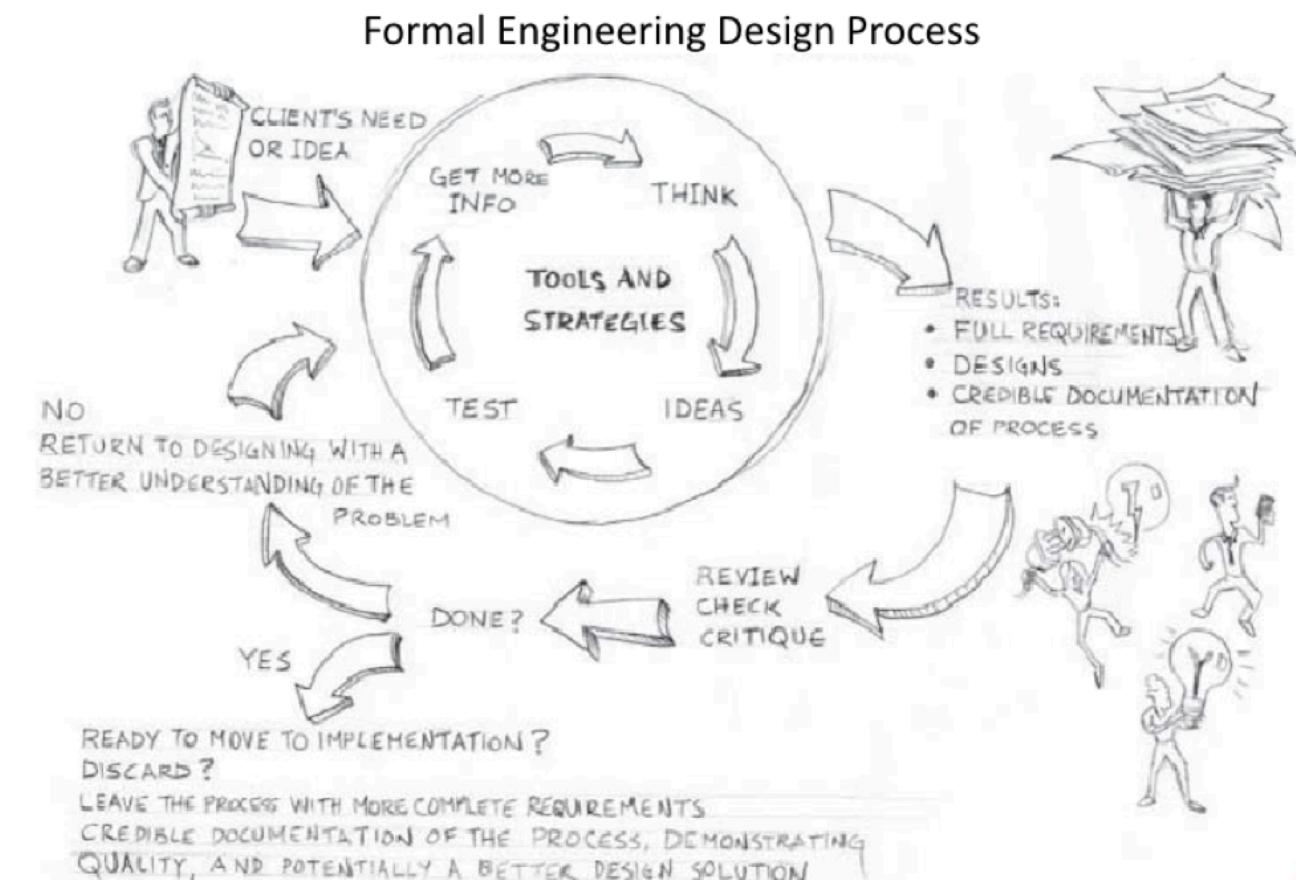
ADJECTIVE screen that shows all kinds of _____.

PLURAL NOUN and _____.

ADJECTIVE figures.

Engineering Design Process

- Learn to define the problem.
- Practice defining test cases.
- Develop an algorithm plan (i.e. a workflow!).
- Program your solution and debugging.



Some reminders.

String Indexing

- An **index** is a position within the **string**
- A particular element of the string is accessed by the **index** of the element surrounded by square brackets
- Positive indices count from the left-hand side, beginning with the first character at index 0, the second index 1, and so on...
- Negative indices count from the right-hand side, beginning with the last character at index -1, the second last at index -2, and so on...

```
>>> x = "Yolo"
```

```
>>> x[0]
```

```
'Y'
```

```
>>> x = "Yolo"
```

```
>>> x[2]
```

```
'l'
```

```
>>> x = "Yolo"
```

```
>>> x[-1]
```

```
'o'
```

String Indexing

- An **index** is a position within the **string**
- A particular element of the string is accessed by the index of the element surrounded by square brackets
- Positive indices count from the left-hand side, beginning with the first character at index 0, the second index 1...
- Negative indices count from the right-hand side, beginning with the last character at index -1, the second last at index -2, and so on...

```
>>> x = "I Love Cats"
```

| | | | | | | | | | | |
|-----|-----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| I | | L | o | v | e | | C | a | t | s |
| -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |

```
>>> x[0]
```

```
'I'
```

```
>>> x[6]
```

```
' '
```

```
>>> x[-11]
```

```
'I'
```

String Slicing

- We can extract more than one character (or substring) using **slicing**
- Uses the syntax [**start** : **finish**], where:
 - **start** is the index where we start the slice
 - **finish** is the index of **one after** where we end the slice
- When either **start** or **finish** are not provided:
 - If **start** index is missing, it defaults to the beginning
 - If **finish** index is missing, it defaults to the end

```
>>> x = "Yolo"
```

```
>>> x[0:3]
```

```
'Yol'
```

```
>>> x = "Yolo"
```

```
>>> x[:3]
```

```
'Yol'
```

```
>>> x = "Yolo"
```

```
>>> x[-2:]
```

```
'lo'
```

String Methods

- Strings are objects and just like other objects, the `str` type has associated methods that are only valid for strings
- To find out which methods are associated with objects, use the built-in function `dir`

```
>>> dir(str)
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__init__',
 '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__',
 '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__rmod__', '__rmul__',
 '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'capitalize',
 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find',
 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal',
 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace',
 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans',
 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit',
 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title',
 'translate', 'upper', 'zfill']
```

String Method: replace

- The method `replace(old, new)` returns a copy of the string in which the occurrences of old have been replaced with new.

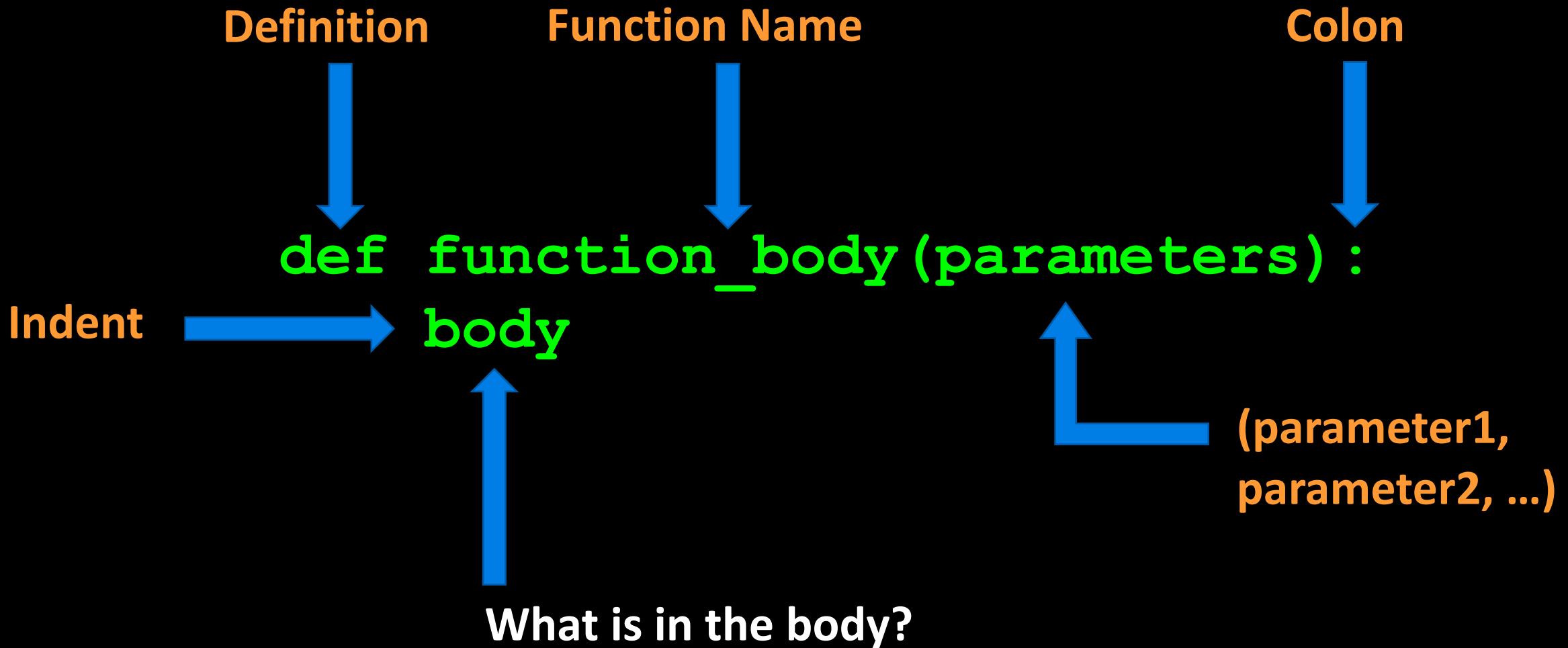
```
>>> white_rabbit = "I'm late! I'm late! For a very important date!"
```

```
>>> white_rabbit.replace('late','early')
```

```
"I'm early! I'm early! For a very important date!"
```



Function Definitions



Function Definitions

```
def function_body(parameters):
```

1. **"""DOCSTRING""" (optional)**
2. Code the does the thing
3. **return [expression]**

The **return** statement is optional and if it is not included, it's the same as writing **return None**

Calling Functions

- The general form of a function call:

`function_name(arguments)`

- Terminology

- *argument*: a value given to a function.
- *pass*: to provide an argument to a function.
- *call*: ask Python to execute a function (by name).
- *return*: give a value back to where the function was called from.

In **Python** names of variables and functions use low case and underscores.



`function_name`
`Function_Name`
`FunctionName`

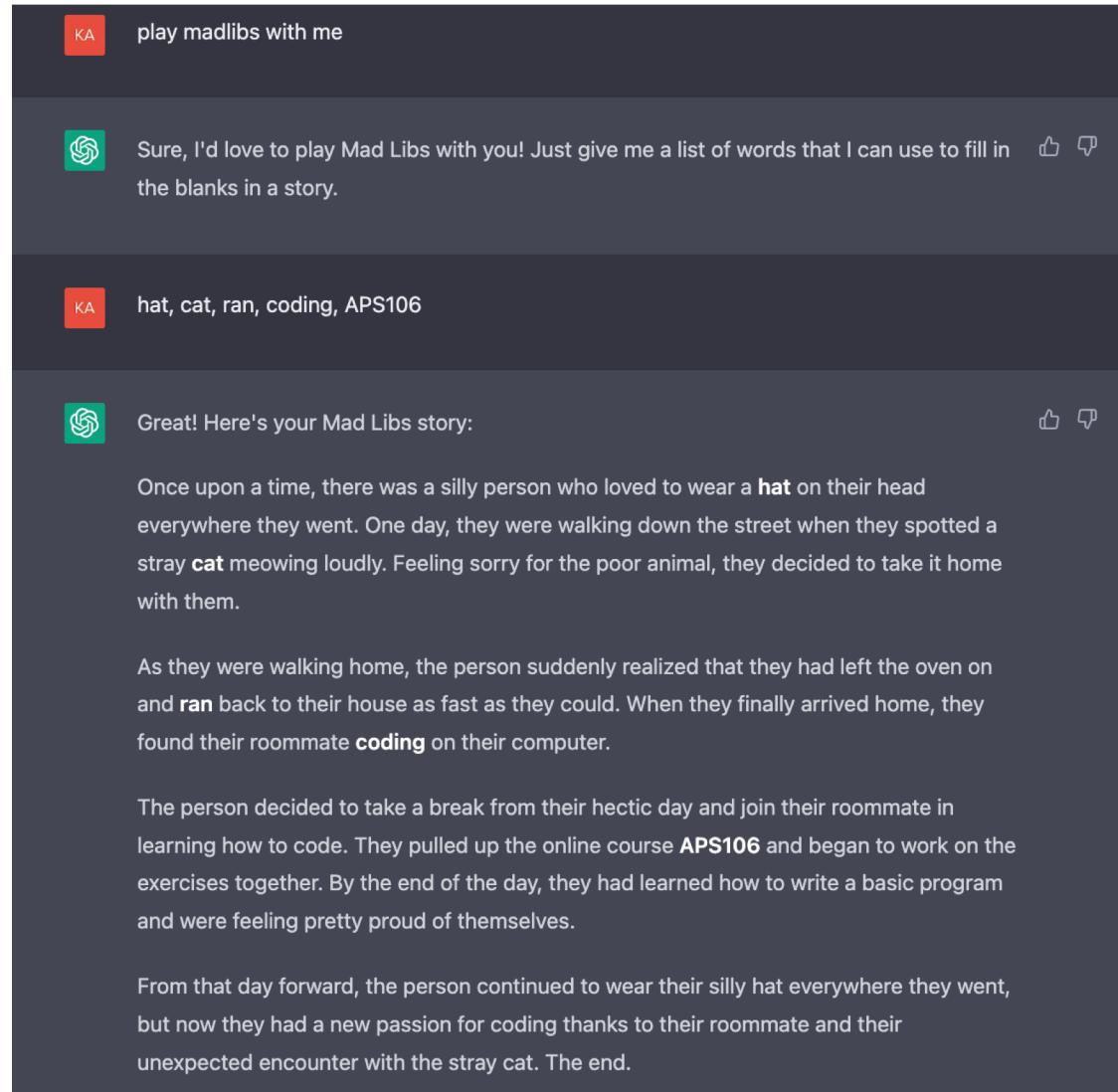
Input

- Python has a built-in function named **input** for reading text from the user.
- The general form of a **input** function call:

input(argument)

- The **argument** is the text you want displayed to the user.
 - “*What is your name?*”
- The value returned by the **input** function is always a string.

ChatGPT Strikes Again



play madlibs with me

Sure, I'd love to play Mad Libs with you! Just give me a list of words that I can use to fill in the blanks in a story.

hat, cat, ran, coding, APS106

Great! Here's your Mad Libs story:

Once upon a time, there was a silly person who loved to wear a **hat** on their head everywhere they went. One day, they were walking down the street when they spotted a stray **cat** meowing loudly. Feeling sorry for the poor animal, they decided to take it home with them.

As they were walking home, the person suddenly realized that they had left the oven on and **ran** back to their house as fast as they could. When they finally arrived home, they found their roommate **coding** on their computer.

The person decided to take a break from their hectic day and join their roommate in learning how to code. They pulled up the online course **APS106** and began to work on the exercises together. By the end of the day, they had learned how to write a basic program and were feeling pretty proud of themselves.

From that day forward, the person continued to wear their silly hat everywhere they went, but now they had a new passion for coding thanks to their roommate and their unexpected encounter with the stray cat. The end.

MadLibs.

Design Problem 3

if nothing else, write `#cleancode`